Genesys®

G*plus* Adapter 5.4

for Verint WFM

Installation and Configuration Guide

About Genesys

Genesys is the world's leading provider of customer service and contact 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.genesys.com for more information.

Each product has its own documentation for online viewing at the Genesys Documentation website.

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

Technical Support from VARs

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

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.
http://www.genesys.com/

Document Version: 54gp_icg_verint-wfm_05-2014_v5.4.001.00

Table of Contents

Preface

Chapter 1

Chapter 2

Chapter 3

Preface

Welcome to the Gplus *Adapter for Verint WFM - Installation and Configuration Guide*. This document describes the G*plus* Adapter for Verint WFM (the G*plus* Adapter) and lists the steps required to install and configure the components of this Adapter.

This document is valid only for the 5.4 release(s) of this product.

Note: For versions of this document created for other releases of this product, visit the Genesys Technical Support website.

For information about related resources and about the conventions that are used in this document, see the supplementary material starting on page 8.

About G*plus* Adapter for Verint WFM

The G*plus* Adapter for Verint WFM is a software solution that provides integration between a Verint Workforce Management (WFM) system and Genesys routing solutions. Genesys event data is filtered and aggregated to provide interval activity data and a real-time data feed for Verint WFM.

Intended Audience

- This document is primarily intended for system administrators or other individuals who will be installing the G*plus* Adapter and have a basic understanding of the following:
    - Computer-telephony integration (CTI) concepts, processes, terminology and applications.
    - TCP/IP networking fundamentals including routing and client/server application communications via TCP sockets.
    - Verint WFM conventions and reports.

- The network configurations used in the installation computing environment.
- The following Genesys applications and solutions:
  - Framework 7.0 - 8.x
  - Configuration Manager
  - Universal Routing 7.0 - 8.x

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@genesys.com.

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 Customer Care

If you have purchased support directly from Genesys, see the Contact Information on the Customer Care website. Before contacting Customer Care, refer to the *Genesys Care Program Guide* for complete contact information and procedures.

Related Documentation Resources

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

- The *Gplus WFM Adapter 5.4 - Hardware and Software Requirements*, which will help in ensuring that the Adapter is installed with sufficient resources.
- The *Framework 8.1 Configuration Manager Help*, which will help when using Configuration Manager.
- The *Genesys 8.1 Security Deployment Guide*, which will help when configuring secure connections to the TServer(s), SIP Server(s), Interaction Server(s) and Configuration Server(s).
- The *Genesys Technical Publications Glossary*, which ships on the Genesys Documentation Library CD and which provides a comprehensive list of the Genesys and CTI terminology and acronyms used in this document.

- The Release Notes and Product Advisories and Genesys product documentation for this product are available on the Genesys Documentation website at `http://docs.genesys.com/Documentation`.

Document Conventions

This document uses certain stylistic and typographical conventions—introduced here—that serve as shorthand 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:

54gp_icg_iex-wfm_06-2014_v5.4.001.00

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

Type Styles

The Type Styles table describes and illustrates the type conventions that are used in this document.

Type Styles

Type Style	Used For	Examples
Italic	- Document titles - Emphasis - Definitions of (or first references to) unfamiliar terms	Please consult the *Genesys Migration Guide* for more information. Do *not* use this value for this option.
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,	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

	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.	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.
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.	`smcp_server -host <confighost>`

Chapter

1 Product Description

Historical Reports

Historical Reports are contained in individual files in tab-delimited format for the four Contact Statistics reports and semicolon-delimited for the Agent Scorecard Metric report:

- Voice Contact Statistics Report - Inbound customer calls
- Outbound Statistics Report - Outbound campaign calls
- Email Contact Statistics Report - Email and iWD/Open Media
- Chat Contact Statistics Report - Customer chats
- Agent Scorecard Metric Report - Agent activity

Files for the three Contact Statistics and the Outbound Statistics reports are generated on a 15, 30 or 60 minute interval depending upon the interval option setting. The Agent Scorecard Metric report is a daily report that is generated at midnight.

The Outbound Statistics report tracks Outbound Campaign calls. The present version of the adapter does not support the “backlog” statistic.

Real Time Adherence (RTA)

RTA tracks agent state to ensure that the agents are in compliance with their set schedules. Verint does not stipulate a standard set of RTA states so the states tabulated below have been selected as the “standard” used by the G*plus* Adapter for Verint WFM. The RTA State Code is the actual `AgentMode` integer that is included in the RTA state transition message.

Verint RTA State	RTA State Code
dndOn	1100
notReady	1101
ready	1102
afterCallWork	1103
afterEmailWork	1104
loggedIn	1200
loggedOut	1201
callInbound	1300
callOutbound	1301
callInternal	1302
callOnHold	1303
callConsult	1304
callConference	1305
directCallInbound	1306
emailInbound	1400
emailReply	1401
emailOutbound	1402
chatInbound	1500
chatInternal	1501
afterChatWork	1502
outboundPreview	1600
outboundCampaignCall	1601
interactionInbound	1700
interactionInternal	1701
afterInteractionWork	1702

Note:

Chapter

2 Installation and Setup

Installation and Windows Permissions

File and folder permissions can be a factor when installing the G*plus* Adapter on Windows Server versions. It is recommended that the same User with Administrator privileges install the adapter files, the Java instance and the Adapter as a Windows Service. The installed service may fail to start if the access permissions are not aligned.

Included Files

The application installation package contains all of the required components with the exception of the Java Runtime Environment (JRE). Installation of the application is a matter of copying the distribution files into a suitable directory. The examples below demonstrate an Adapter instance installed in the `GPlusAdapter` folder.

Folders

• `GPlusAdapter/config`	– configuration files and license file
• `GPlusAdapter/lib`	– application jar files
• `GPlusAdapter/ftp`	– reports folders
• `GPlusAdapter/licenses`	– third party software licenses
• `GPlusAdapter/logs`	– log files
• `GPlusAdapter/recovery`	– recovery log files
• `GPlusAdapter/scripts`	– Windows and “unix” scripts

- `GPlusAdapter/wrappers` – Java Windows Service wrappers

Files

Note: The script files with the "`.example`" suffix listed below require configuration additions before they can be executed. The suffix must also be removed.

GPlusAdapter\

- `bootstrap.jar` – start up jar file

GPlusAdapter\scripts\windows\

- `windows_install.bat` – installation script
- `encryptPassword.bat.example` – password encryption script
- `serverHostName.bat.example` – hostname script
- `startup.bat.example` – console start up script
- `svcinstall.bat.example` – service installation script

Note: The `scripts\windows` folder contains a `subroutines` folder that holds the scripts called by `windows_install.bat`. There is nothing to configure in this folder and it should not be tampered with.

GPlusAdapter/scripts/unix_bash/

- `encryptPassword.sh.example` – password encryption script
- `serverHostName.sh.example` – hostname script
- `shutdown.sh` – shut down script
- `startup.sh.example` – start up script

GPlusAdapter\config

- `GPlusWFM.properties` – Configuration Server connection
- `product.properties` – Adapter name and version
- `jvm.properties` – JVM location, memory and type
- `log4j.xml.example` – logging configuration
- `license.txt` – license file

Logging, FTP and Report Directories

A default set of folders for log files and historical reports are included as part of the installation package. These folders (with the exception of the `recovery` folder) do not have to be used and the configuration options allow for alternatives.

Logging

- Default - `logs` folder
- The number, size and location of the adapter log files are set in the `log4j.xml` file contained in the `config` folder.
- The default service install script for Windows creates the `AdapterService`, `service-stdout` and `service-stderr` log files in the `logs` folder but their location can be changed.

Reports

- Default - `ftp/reports` folder
- Option - `historical.reports/directory`
- Reports folders for historical streams are automatically created as subfolders of the reports folder using the stream name as the subfolder name.
- Normal operation sees the report files transferred to a remote server and the local copy removed from the reports folder.
- The number of days that the report files are retained if they are not transferred is set with the `historical.reports/reportRetentionDays` Option.

FTP Source

- Default - `ftp/reports` folder
- Option - `historical.ftp/localSourceDirectory`
- The Gplus Adapter attempts to transfer all of the files contained in this folder immediately after the report generation task has ended.

FTP Backup

- Default - `ftp/reportsBackup` folder
- Option - `historical.ftp/localBackupDirectory`
- Backup folders for historical streams are automatically created as subfolders of the backups folder using the stream name as the subfolder name.
- The number of days that the FTP backup files are retained is set as an application option in the historical.ftp section.

Recovery

- Default - `recovery` folder
- This folder contains the archived logs of all of the events received by the G*plus* Adapter instance for the last seven days.
- Neither the folder nor the log files should be tampered with as the G*plus* Adapter replays the last day's events at startup to recover the state prior to shutting down.

Adding an Adapter Instance to the Configuration Server

The Adapter does not use a wizard application to simplify the installation but rather requires manual steps to add the application to the Configuration Server and complete the configuration. The present version of the Adapter does not support installation with the Genesys Administrator and its meta-data files. It is recommended that the Adapter be installed using the Genesys Configuration Manager.

Importing the Application Template

The `templates` folder contains the `GPlus_Adapter_Verint_54.apd` file that defines the properties for the adapter's Application Template. This file should be moved to a location accessible by the computer running the Genesys Configuration Manager so that it can be imported into the Configuration Server.

Once the Application Template has been imported, the name should be confirmed prior to saving it. `GPlus_Adapter_Verint_54` is one possibility but alternatives are also acceptable as long as the template name is unique.

Creating the Application in the Configuration Layer

An empty application can be created after the Application Template has been added to the Configuration Layer.

1. Select the `Applications` folder and right click in the right frame to bring up the menu.
2. Select `New` ... `Application`.
3. Select the recently imported G*plus* Adapter template.

Once the template is selected, an `Application Properties` window will appear. Once again, a unique name should be entered for the Adapter instance. At this point, one option would be to continue with the configuration of the new application object but it is suggested that new application be saved by selecting the `OK` button. Saving the new `Application` object requires that something be added to the fields in the `Start Info` tab. Periods ('.') will suffice until the start information is properly configured.

Java Virtual Machine

The G*plus* Adapter is a Java application that runs in a Java Virtual Machine (JVM) that isolates the application from the underlying operating system. The significant benefit is that the same code runs on more than one operating system. One downside is that there is some configuration required to properly start the application in the JVM. There are two steps to successfully starting the G*plus* Adapter:

- Invoking the Java application launcher to load and run the Adapter application in the JVM.
- Connecting to the Genesys Configuration Server to obtain the saved application configuration options.

The first of these two steps differs slightly between Windows and "unix" installations. Additionally, a G*plus* Adapter instance installed on a Windows server can be started as a console application or as a Windows Service.

Note: "unix" is used as a generic description for either the Solaris or Linux operating systems.

JRE or JDK Java Version

The Java Runtime Environment (JRE) is not bundled with the Adapter installation files and the actual location of the JRE is a parameter that is specified in the startup scripts. Confusingly, there are two Java versions available from Oracle: a JRE version that contains the executables and library files to run a Java application and the Java Development Kit (JDK) version that contains additional tools not available in the JRE version. It is recommended that the JDK version of Java 1.6 or higher be installed rather than the JRE version.

Windows

Note: The Windows script files are located in the `scripts\windows` folder of the Adapter installation.

windows_install.bat

New with the 5.4 version is the addition of the `windows_install.bat` script file that performs the following installation tasks as a single menu driven script. This script file is described in Appendix E below and it is the recommended way to install the G*plus* Adapter as a Windows Service.

This script incorporates many of the steps discussed below and it is also recommended that the following steps be read for background. It should be noted that the Gplus Adapter can still be installed as described below using the listed script files.

startup.bat and svcinstall.bat

Both the `startup.bat` file and the `svcinstall.bat` Windows script files have common parameters that must be set to identify the JRE that will be used as well as the allocated memory limit for the running G*plus* Adapter instance.

Setting **`WFM_JAVA_HOME`**

The `WFM_JAVA_HOME` parameter has to set to a base folder in the Java installation as shown in the example below.

```
set WFM_JAVA_HOME=../java/jdk1.6
```

JVM Memory

The memory setting configures the maximum amount of memory that the JVM will allocate on startup. The default is 1024 MB and it should be sufficient for all but the very largest call centers.

```
set JVM_MEM=1024m
```

The maximum memory available to the JVM in a Window 32 bit OS varies but it should not be set higher than 1.3 GB. In the unlikely event that more memory than those limits is required, a 64 bit version of the JVM can be used. The memory available on a 64 bit OS would be constrained by the amount of memory on the server. It should be noted that the memory requirements actually increase with a 64 bit OS as the memory pointer size doubles.

startup.bat Script

The Windows batch file `startup.bat` is immediately executable as soon as the `.example` suffix has been removed. A console window will be opened and the program will continue to run as long as that window is open. This is fine for debugging and testing the application but is not a viable mode for running a server application. Once the application has been set up and tested using the `startup.bat` file, it should be converted into a Windows service or invoked directly with the Genesys Solution Control Server.

The following is an excerpt from the Windows `startup.bat` script showing part of the command to start the adapter:

```
%WFM_JAVA_HOME%\bin\java.exe -Xms%JVM_MEM%m ...
```

The `java.exe` file is the executable that starts the JVM. The file path created by the concatenation of the `WFM_JAVA_HOME` parameter and the rest of the file path, `\bin\java.exe`, must point to the actual Java executable.

svcinstall.bat and jvm.dll

When the Adapter is installed as a Windows service, it does not use the `java.exe` file but rather links to the `jvm.dll` file. There is a server version of that file available and this is the recommended version to use with the Gplus Adapter. The excerpt shown below is taken from the `svcinstall.bat` file and the file path assumes that the server version is being used:

```
...%WFM_JAVA_HOME%\jre\bin\server\jvm.dll ...
```

Again, the combination of the `WFM_JAVA_HOME` parameter and the file path must identify an actual file in the Java installation.

Installing a Windows Service

The following description discusses installing the adapter as a Windows service using the Apache Commons™ *daemon* ™ *procrun* Java service wrapper.

The included `svcinstall.bat` file must reference the installed JRE in exactly the same way that the `startup.bat` file was configured with the `WFM_JAVA_HOME` parameter. An additional parameter, `SVCNAME`, must be set to identify the new service in the Services Console as shown below:

```
set SVCNAME=GPlusAdapter_WFM_P
```

The service name must match the Configuration Server application name of the Adapter instance that is configured in the `GPlusWFM.properties` file described below.

One point to note is the location of the `jvm.dll` file as described above. The other is that it is also possible to modify the logging of the service process. The installation script has the following service logs enabled:

- `AdapterService.YEAR-MONTH-DAY.log`
- `service-stdout.YEAR-MONTH-DAY.log`
- `service-stderr.YEAR-MONTH-DAY.log`

The logging can be disabled by removing the parameters or using the prunmgr application.

This script file must be run once to install the adapter instance as a service after the parameters have been set. As part of the service installation, the `prunsrv.exe` file is copied and given the name of the new service e.g. `GPlusAdapter_WFM_P.exe`.

32 bit or 64 bit Service Installation

If a 64 bit JRE is being used to run the Adapter, then it must be installed as a 64 bit service. The only reason to use a 64 bit instance is to increase the memory beyond the limits of the 32 bit JVM.

The default installation uses the following line to rename the `prunsrv.exe` executable but there are three versions of that file available depending on the Windows version and JRE type:

```
COPY prunsrv.exe %RENAMED_EXE%
```

Copies of the versions have been placed in the three folders contained in the `wrappers` folder. The *ia64* version is specific to a 64 bit machine using an Itanium™ processor with the *amd64* version used for all other 64 bit processors. The appropriate file should be placed in the main folder prior to executing the `svcinstall.bat` script.

- `wrappers/32_bit` – default 32 bit JRE version
- `wrappers/amd64` – default 64 bit JRE version
- `wrappers/ia64` – 64 bit JRE on Itanium**™** version

Modifying an Existing Service Installation

After a successful installation, the adapter will automatically start as a service when the server boots up and the service can be stopped or started through the Windows Services Console. For most installations, there is no need to change the JVM parameters from the defaults set in the `svcinstall.bat` script but it is possible to change the start up parameters for the JVM if required using the included `prunmgr.exe` application. This is discussed in detail in Appendix C.

Removing an Existing Service Installation

Uninstalling the service can be carried out with the following command (using the example service given above):

```
prunsrv.exe //DS//GPlusAdapter_WFM_P
```

bash Shell Scripts

Note:	The *bash* script files are located in the `scripts\unix_bash` folder of the Adapter installation.

The `startup.sh.example` and `shutdown.sh` files are standard *bash* shell scripts. The startup script has to be renamed to remove the `.example` suffix and then both files have to be made executable with the `chmod` command. An example is given below:

```
chmod 751 ./startup.sh
```

The `startup.sh` script file starts up the Adapter as a background process that will continue to run until the `shutdown.sh` script is executed. The process ID for the running adapter is contained in the `WFM_PID` file and this file should not be tampered with or that particular process will have to be stopped manually.

Setting `WFM_JAVA_HOME`

The `WFM_JAVA_HOME` parameter has to set to a base folder in the Java installation as shown in the example below.

```
WFM_JAVA_HOME=../java/jdk1.6
```

JVM Memory

The memory setting configures the maximum amount of memory that the JVM will allocate on startup. The default is 1024 MB and it should be sufficient for all but the very largest call centers.

```
JVM_MEM=1024m
```

The maximum memory available for a Linux 32 bit installation is 2.0 GB. In the unlikely event that more memory than those limits is required, a 64 bit version of the JVM can be used. The memory available on a 64 bit OS would be constrained by the amount of memory on the server. It should be noted that the memory requirements actually increase with a 64 bit OS as the memory pointer size doubles.

Configuration Server Connection

Former versions of the G*plus* Adapter had configured the connection to the Configuration Server in the startup scripts but the connection parameters should now defined in the `GPlusWFM.properties` file contained in the `config` directory of the adapter installation. The adapter will still accept the parameters defined in the script files but using the new properties file is recommended.

Identifying the Adapter's Application Object

This property references the previously created Application object in the Configuration Server so that the proper configuration information can be acquired and used by the associated adapter instance.

```
appName=<applicationName>
```

Setting the Application Type

Applications registering with the Configuration Server have to identify their application type as part of establishing the connection. The G*plus* Adapter has been configured as a `ThirdPartyServer` in the application template discussed above. The `cmeAppType` parameter has been set properly and can be ignored in most cases.

```
cmeAppType=ThirdPartyServer
```

Applications configured as a `ThirdPartyServer` application type can register with the Genesys Configuration Server once without specifying a username or password. Multiple registrations with the same Application Name (`appName`) are not allowed.

The properties file still has the username and password parameters listed although they are not usually required and they are only used if the application type is set to `ThirdPartyApplication`.

```
cmeUser=
cmePass=
```

Configuring the Adapter instance as a `ThirdPartyApplication` would allow for multiple connections to the Configuration Server using the same credentials

Encrypting the Configuration Server Password

If the application type is set to `ThirdPartyApplication`, it is possible to encrypt the password for the CME user. The `encryptPassword` script files will encrypt the provided password and store it in the `enc` file contained in the `config` folder. The Adapter will check for this file at startup and use the encrypted password as the user's credentials when connecting to the Configuration Server.

The scripts start a small Java application and the JRE to be used to run the application must be identified with the `WFM_JAVA_HOME` parameter. The script file must be run from the console as it does not prompt for the password but rather expects the password as the only parameter.

Example:

```
encryptPassword.bat  cmePassword
```

Setting Configuration Server Connection Parameters

The next set of properties specifies the connection parameters for the Genesys Configuration Server.

Socket Parameters

The parameters below are mostly self explanatory. The Adapter instance will be attempting to connect to the Configuration Server through a socket defined by a host name and port number.

```
cmeHost=<hostName>
cmePort=2020
cmeBackupHost=
cmeBackupPort=
```

Note: The host name is the actual server host name or IP address.

The backup Configuration Server host and port can also be added if there is a backup. If there is no backup server, the backup parameters should be set to the primary Configuration Server. The connection parameters defined in the following sections will be shared by both the primary and backup Configuration Server connections.

Setting Registration Timeout and Delay

The next three connection properties configure how the G*plus* Adapter registers for notifications after connecting (or reconnecting) to the Configuration Server. These parameters were added to reduce the load on the Configuration Server in a busy call center when a number of applications may be attempting to reconnect and register requests at the same time.

```
configTimeout=10m
registrationMinDelay=5m
registrationMaxDelay=15m
```

The first parameter (`configTimeout`) sets the amount of time the Adapter will wait before attempting to reregister a request for which it has not received a response. The default is 10 minutes and that should be more than sufficient for most call centers but it can be increased if required.

The other two parameters set the bounds for when the Adapter instance will randomly attempt to register for notifications after reconnecting to the Configuration Server. The delay will be random time longer than `registrationMinDelay` but less than `registrationMaxDelay`. Again, these parameters would only be relevant if a large number of Adapters were connected to the same Configuration Server.

Setting addp Protocol

The next set of parameters configures the `addp` connection protocol. This protocol acts as a *heartbeat* for the connection and each side of the connection can be notified if the connection is inadvertently dropped.

```
addpEnabled=false
addpLocalTimeout=45
addpRemoteTimeout=75
addpTraceMode=full
```

Both timeouts are in seconds and should be set to different values with the remote timeout greater than the local timeout.

It is recommended that if `addp` is enabled, the `addpTraceMode` should be set to `full` which is the equivalent of `Trace On Both Sides`. This setting will ensure that the Adapter is notified if the connection to the Configuration Server is disconnected.

Enabling Transport Layer Security (TLS)

This connection parameter enables the Adapter to use Transport Layer Security (TLS) when connecting to the Configuration Server. The actual configuration for TLS occurs on the server side.

Setting up the Configuration Server for TLS and providing a security certificate for the adapter are well beyond the scope of this document. The *Genesys 8.0 Security - Deployment Guide* does discuss the steps required to set up TLS.

```
tls=false
```

Note: The secure port on the Configuration Server may not be the default and the `cmePort` parameter has to be set to that secure port if TLS is enabled.

Solution Control Server Integration

Note: SCS integration also requires a Local Control Agent (LCA) instance installed on the server hosting the G*plus* Adapter and a properly configured `Host` object in the Configuration Server.

The G*plus* Adapter can be started and stopped using the Solution Control Interface for both the Windows and “unix” versions by invoking the Java executable with the command line arguments. Additionally, an adapter instance installed as a Windows Service can also be controlled by adding two Options to the Adapter instance's `Annex Data`.

Invoking the Application Directly

The following fields in the `Start Info` tab of the Adapter's Configuration Properties must be set as described for the Solution Control Server to start and stop the application.

Working Directory

The working directory is set to the root of the adapter installation files:

```
/Applications/GPlus_WFM_Adapter
```

Command Line

The command line starts up the JVM with the `java` command as in this example:

```
/java/jdk1.6_33/jre/bin/java
```

It is also possible to just invoke the `startup.bat` file in the case of a Windows installation if that file was configured properly:

```
startup.bat
```

Command Line Arguments

If the command line does not use the startup script, then the command line arguments are given below:

```
-Xms1024m -Xmx1024m -jar bootstrap.jar com.ariasolutions.iconnect.BootStrap
```

The only variable set in the command line arguments is the JVM memory size and both the minimum `(-Xms)` and the maximum `(-Xmx)` should be set to the same amount. The other parameters should not be changed from the example above which shows the allocated memory set to 1024 MB.

Start and Stopping a Windows Service

The following instructions are relevant if the Adapter instance has been installed as a Windows Service and the Service is to be started and stopped through the Solution Control Server.

Note: The LCA version should be 8.1.200.07 or higher and the Solution Control Server should be an 8.1.x version to be compatible with the LCA.

Working Directory

The working directory is still set to the root of the adapter installation files. The directory that contains the renamed `prunsrv.exe` as described above:

```
/Applications/GPlus_WFM_Adapter
```

Command Line

The command line just identifies the renamed `prunsrv.exe` that was created when the Service was installed:

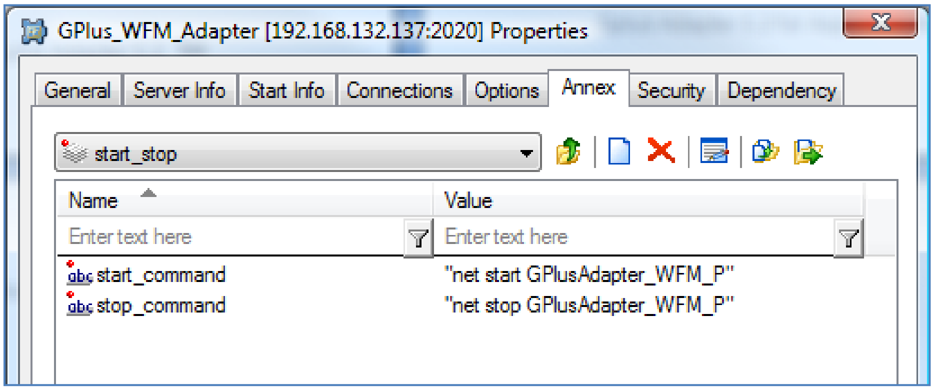
```
GPlusAdapter_WFM_P.exe
```

Command Line Arguments

The command line arguments field can be left blank.

Adding Options to the Adapter's Annex Tab

A new Section, `start_stop`, has to be added to the Adapter's `Annex` Tab. Two options, `start_command` and `stop_command`, are added with their Values containing the commands that explicitly start and stop the Windows Service installed for the Adapter instance. The example below shows the configuration for a Service instance named `GPlusAdapter_WFM_P`.



start_stop

Name	Value
start_command	"net start GPlusAdapter_WFM_P"
stop_command	"net stop GPlusAdapter_WFM_P"

Logging Configuration

The G*plus* Adapter has three types of logs and only the first two are configurable beyond being enabled or disabled.

1. TServer and Interaction Server events with the adapter state changes associated with those events.
2. A summary of errors also logged in the first log.
3. The recovery log.

Event and Error Logs

log4j.xml File Location

The first two log types are configured using the parameters found in the `log4j.xml.example` file contained in the configuration directory (`./config`). This file will not be the active logging configuration until the name is changed to `log4j.xml`.

ROLLING Appender - No Compression

The following is an excerpt from the `log4j.xml` file showing some of the parameters that can be set for the `ROLLING` appender. This appender logs the events to a set of files that are continuously updated.

```
<appender name="ROLLING" class="org.apache.log4j.RollingFileAppender">
   <param name="File" value="logs/GPlusAdapterWFM.log" />
   <param name="MaxFileSize" value="50MB" />
   <param name="MaxBackupIndex" value="2" />
   ...
</appender>
```

The log file destination folder and name are specified by the `File` parameter. In the example above, the file (`GPlusAdapterWFM.log`) will be created and updated in the `logs` directory that was created when the application was installed. The destination folder could have been set to any folder accessible from the installation.

The `ROLLING` log files are set to roll over when the file size of the current file (`GPlusAdapterWFM.log`) reaches the `MaxFileSize` (50MB). The older files are indicated by an integer appended to the file name. The `MaxBackupIndex` is set to 2 in the example so only two backup files would be kept: the most recent, `GPlusAdapterWFM.log.1` and the previous file, `GPlusAdapterWFM.log.2`. By adjusting these two parameters and depending upon the call volume, the uncompressed files could keep an hour or two of the recent activity for review.

GZIP Appender - Compressed

The events log records all of the events received by the G*plus* Adapter as well as operational states. The logs can become quite large and for that reason the `GZIP` appender was added. The `GZIP` appender continuously archives the logging information to compressed files. This file space saving feature enables the uncompressed files built by the `ROLLING` appender to serve as a recent snapshot of adapter activity and the `GZIP` files provide the longer term archive.

```
<appender name="GZIP" class="com.ariasolutions.iconnect.util.log.GZIPAppender">
    <param name="Prefix" value="logs/GPlusAdapterWFM.log" />
    <param name="RollOverField" value="hour" />
    <param name="MaxBytes" value="1500MB" />
    ...
</appender>
```

The log file destination is specified by the `Prefix` parameter for the `GZIP` appender. In the above examples, the logging directory has been set to the `logs` directory that was created when the application was installed but it could have been set to any folder accessible from the installation.

The `GZIP` appender is configured to start a new file hourly with the `RollOverField` parameter set to `hour`. The `MaxBytes` parameter specifies the maximum number of storage bytes used for all of the zipped log files. When the `MaxBytes` limit is exceeded, the adapter will start deleting the oldest zipped files until the limit is not exceeded. Each file name will have the date and time that the file was created appended to the `Prefix` specified in the configuration file.

ROLLING_ERROR Appender

The `ROLLING_ERROR` appender is essentially the `ROLLING` appender with the logging threshold level set to `error`. The errors will still be logged to the `ROLLING` and `GZIP` logs but the error log file allows for a quick check to see if the adapter is running error free rather than having to search through the event logs.

Logging Levels

Logging levels for the `log4j` library are limited to six levels with `fatal` being the least verbose and `trace` being the most. The default level set in the `log4j.xml.example` file is `debug`. In production or when detailed log entries are no longer required, the logging threshold could be increased but this can be set easily in the Configuration Layer with the `application/logThreshold` option discussed below.

Adding or removing appenders

The example file includes a definition for a STDOUT appender that has been left commented out. This appender allows logging to the console and it is not practical for a production instance. The following excerpt from the example file shows that the STDOUT appender is not included in the logging but the ROLLING, ROLLING_ERROR and GZIP appenders are included:

```
<root>
   <level value="debug"/>
   <!--  <appender-ref ref="STDOUT"/>  -->
   <appender-ref ref="ROLLING"/>
   <appender-ref ref="ROLLING_ERROR"/>
   <appender-ref ref="GZIP"/>
 </root>
```

The xml comment tags (<!-- -->) can be used to exclude any of the logging appenders.

Windows Service - out.log and err.log

The parameters for installing the adapter as a Windows Service include specifying that stdout and stderr for the adapter are redirected to two log files: service-stdout*.log and service-stderr*.log. Usually both log files will be empty but in the event of a fault with the JVM, that fault should be logged in the err.log file.

If the STDOUT appender is enabled, the adapter logs will be directed to the service-stdout*.log file. This is not recommended as the file is not "managed" and it will continue to grow until the available disk space is consumed.

Recovery Log

The recovery log will be contained separately in the recovery folder. This log has no configuration options and is managed automatically by the G*plus* Adapter. The log files contain enough information to restart the adapter after a mishap without losing the data recorded prior to the mishap. It should not be tampered with.

If data contained in the recovery log prevents the adapter from starting up, the log files can be removed from the recovery folder to allow the application to start.

Initial Start Up Test

The application has not been configured in the Configuration Manager but it should now start up and create a log file before failing. This test requires the startup script to be run and then a confirmation that a brief log file has been created listing the exceptions generated from starting without configuration. The log file(s) should be created in the directory specified in the `log4j` configuration.

In the case of the *bash* shell scripts, `shutdown.sh` has to be run to ensure that the stalled run state has cleared. Both shell script files should have been modified to be executable during the initial setup of the application.

License File

A temporary license file (`license.txt`) is included in the `config` folder of the G*plus* Adapter installation.   This license will allow the adapter to run on any server for a limited amount of days. The expiry date is listed in the license file. When a production license has been purchased, the temporary license file must be replaced with the new file.

While the temporary license allows the Adapter instance to run on any host server, the purchased permanent licenses are created with permanent fields that cannot be altered. Changing any of these fields requires creating a new license e.g. moving the Adapter to a new server would require a new license. The following fields are required and restrict the Adapter to being installed on one host server and acquiring its configuration from an existing `Application` object in the Configuration Server:

- Host name
- Adapter application name in Configuration Server
- Vendor (Verint)
- Number of seats

Determining the Host Name

Depending upon the host operating system and network configuration, the required host name may be either a simple hostname or a qualified hostname:

- Simple `adapterhost`
- Qualified `qualified.domain.adapterhost`

The JRE libraries are used to check the licensed hostname in the Adapter and those libraries are also used with the provided `serverHostName` script file. Executing the script file will display the required hostname in the console window. There are Windows and *bash* versions of these script files. The scripts start a small Java application and the JRE to be used to run the application must be identified with the `WFM_JAVA_HOME` parameter.

The `.example` extension has to be removed from the script name and the bash version has to be made executable.

Chapter

3 Application Options

Each of the G*plus* Adapter application options is discussed in this chapter. To change the application options, open the Adapter `Application` in the Configuration Manager and select the `Options` tab.

alarmCodes Section

This Option Section deals with the alarm codes sent to the Genesys Message Server. The alarm codes sent must match up with the `Log Event IDs` defined in the `Alarm Conditions` contained in the Configuration Server.

diskWriteFailure

Type:	Optional
Default Value:	*Not set*
Valid Values:	`Alarm Condition/Detect Event/Log Event IDs`
Dependencies:	`diskWriteSuccess`

This option identifies the integer `Alarm Code` sent to the Message Server when the G*plus* Adapter encounters a problem writing the event logs, the recovery logs or the historical reports to the disk drive.

diskWriteSuccess

Type:	Optional
Default Value:	*Not set*
Valid Values:	`Alarm Condition/Detect Event/Log Event IDs`
Dependencies:	`diskWriteFailure`

This option identifies the integer `Alarm Code` sent to the Message Server when the problem that generated a `diskWriteFailure Alarm Code` has been cleared.

ftpFailure

Type: Optional
Default Value: *Not set*
Valid Values: `Alarm Condition/Detect Event/Log Event IDs`
Dependencies: `ftpSuccess`

This option identifies the integer `Alarm Code` sent to the Message Server when the G*plus* Adapter encounters a problem sending the generated report files to the remote WFM server.

ftpSuccess

Type: Optional
Default Value: *Not set*
Valid Values: `Alarm Condition/Detect Event/Log Event IDs`
Dependencies: `ftpFailure`

This option identifies the integer `Alarm Code` sent to the Message Server when the problem that generated a `ftpFailure Alarm Code` has been cleared.

reportFailure

Type: Optional
Default Value: *Not set*
Valid Values: `Alarm Condition/Detect Event/Log Event IDs`
Dependencies: `reportSuccess`

This option identifies the integer `Alarm Code` sent to the Message Server when the G*plus* Adapter encounters a problem generating the report files.

reportSuccess

Type: Optional
Default Value: *Not set*
Valid Values: `Alarm Condition/Detect Event/Log Event IDs`
Dependencies: `reportFailure`

This option identifies the integer `Alarm Code` sent to the Message Server when the problem that generated a `reportFailure Alarm Code` has been cleared.

rtaConnected

Type: Optional
Default Value: *Not set*
Valid Values: `Alarm Condition/Detect Event/Log Event IDs`
Dependencies: `rtaDisconnected`

This option identifies the integer `Alarm Code` sent to the Message Server when the connection problem that generated a `rtaDisconnected Alarm Code` has been cleared.

rtaDisconnected

Type:	Optional
Default Value:	*Not set*
Valid Values:	`Alarm Condition/Detect Event/Log Event IDs`
Dependencies:	`rtaConnected`

This option identifies the integer `Alarm Code` sent to the Message Server when the G*plus* Adapter's RTA connection to the Verint server was terminated by either side.

rtaConnected:<streamName>

Type:	Optional
Default Value:	*Not set*
Valid Values:	`Alarm Condition/Detect Event/Log Event IDs`
Dependencies:	`rtaDisconnected:<streamName>`

This option identifies the integer `Alarm Code` sent to the Message Server when the connection problem that generated a `rtaDisconnected:<streamName> Alarm Code` has been cleared.

rtaDisconnected:<streamName>

Type:	Optional
Default Value:	*Not set*
Valid Values:	`Alarm Condition/Detect Event/Log Event IDs`
Dependencies:	`rtaConnected:<streamName>`

This option identifies the integer `Alarm Code` sent to the Message Server when the G*plus* Adapter's RTA connection associated with the Stream identified by `<streamName>` was terminated by either side.

serverConnected

Type:	Optional
Default Value:	*Not set*
Valid Values:	`Alarm Condition/Detect Event/Log Event IDs`
Dependencies:	`serverDisconnected`

This option identifies the integer `Alarm Code` sent to the Message Server when the connection problem that generated a `serverDisconnected Alarm Code` has been cleared.

serverDisconnected

Type:	Optional
Default Value:	*Not set*
Valid Values:	`Alarm Condition/Detect Event/Log Event IDs`
Dependencies:	`serverConnected`

This option identifies the integer `Alarm Code` sent to the Message Server when one the Genesys Server (TServer, SIPServer or Interaction Server) connections has been disconnected.

serverConnected:<serverName>

Type:	Optional
Default Value:	*Not set*
Valid Values:	`Alarm Condition/Detect Event/Log Event IDs`
Dependencies:	`serverDisconnected:<serverName>`

This option identifies the integer `Alarm Code` sent to the Message Server when the connection problem that generated a `serverDisconnected Alarm Code` has been cleared.

serverDisconnected:<serverName>

Type:	Optional
Default Value:	*Not set*
Valid Values:	`Alarm Condition/Detect Event/Log Event IDs`
Dependencies:	`serverConnected:<serverName>`

This option identifies the integer `Alarm Code` sent to the Message Server when the Genesys Server (TServer, SIPServer or Interaction Server) connection identified by `<serverName>` has been disconnected.

application Section

This Option Section deals with the general options for the Adapter instance.

agentIdMode

Type:	Mandatory
Default Value:	personUserName
Valid Values:	agentLogin - Switch Login ID field personEmployeeId - Employee ID field personUserName - User Name field
Dependencies:	None

This option identifies which of the Genesys Person fields will be reported as the agent identifier in G*plus* Adapter reports. In multi-site implementations, it is recommended this option be set to personUserName or personEmployeeId to avoid duplicated agentLogin IDs across PBXs. In email and chat implementations, only personUserName or personEmployeeId option values can be used as no PBX switch is involved.

annexSection

Type:	Optional
Default Value:	*Not set*
Valid Values:	Any text string acceptable as an Annex Section name.
Dependencies:	None

This option identifies the name of the Annex Section that will contain the G*plus* specific options added to either a Place Group, VQ or Skill object in the Configuration Server. It is most commonly used when multiple primary G*plus* Adapter applications are configured for site-based reporting. More than one Annex Section containing a site name can be configured for a particular Place Group.

Note: Contact Professional Services prior to configuring this option.

callType

Type:	Mandatory	
Default Value:	queueNumber	
Valid Values:	skill	list of Skill Names.
	queueNumber	list of Virtual Queue Numbers.
	queueAlias	list of Virtual Queue Aliases.
	custom	list contained in callTypes.custom
Dependencies:	genesys.queueGroups, callTypes.custom	

This option specifies the Configuration Object type (or in the case of Virtual Queues, the field within the type) that will be used as the contact ID for each voice call. The G*plus* Adapter uses the appropriate Configuration Objects to create a list of contact IDs. There are two consequences of this setting:

1. The created list of contact IDs must correspond to the contacts/queues expected by the Verint server in the Contact Statistics report.
2. It must be possible to parse the contact ID from the attributes of an EventQueued received from one of the monitored DNs.

The use of the event.properties/callType.determiner allows the event parsing to be customized. This should minimize the need for any changes to the Genesys environment. In addition, if one of the queue values has been selected, all Virtual Queues to be included in the list of reported contact types must be included in the genesys.queueGroup Section of the G*plus* Adapter application.

When the contact types used by the WFM application do not map to either Skills or Virtual Queues, a custom option value must be used and a callTypes.custom Section containing a list of the reported contact types must be added to the application options.

countUnattachedACW

Type:	Optional
Default Value:	false
Valid Values:	true, false
Dependencies:	None

This option is applicable to environments where the soft-phone/hardset allows an agent to choose the ACW (AfterCallWork) state at any time. When set to true, an agent choosing the ACW state in a login session with no previous routed calls to associate with that state (e.g. Login, ACW and Logout) will have the time counted as ACW Time against a default callType in the G*plus* Adapter reports. If set to false, any ACW time in a login session with no previous handled routed calls will be reported as NotReady/Aux time.

defaultAnnexSection

Type:	Deprecated
Default Value:	*Not set*
Valid Values:	Any valid Annex section name
Dependencies:	None

This option specifies an Annex section name to be used by the adapter in the event that the default Annex section name, GPlus_WFM, cannot be used. This option is applicable to all of the configuration objects used by the adapter.

inheritOptionsFromApp

Type:	Optional
Default Value:	*Not set*
Valid Values:	Any *GPlus* adapter application name.
Dependencies:	None

This option would typically be set when configuring a backup G*plus* Adapter application. The backup application will inherit relevant options from the adapter instance identified by the specified application name. Any options set for the backup adapter instance will override those set for the primary instance.

logThreshold

Type:	Mandatory	
Default Value:	debug	
Valid Values:	fatal	Least verbose level
	error	
	warn	
	info	
	debug	
	trace	Most verbose level
Dependencies:	None	

The logging level of the application at startup is set to the level defined in the config/log4j.xml file. This option defines the logging level that the application will use after the configuration information is read from the Configuration Server.

Appendix D discusses different log file examples based on the configured logThreshold.

siteBased

Type:	Optional
Default Value:	`false`
Valid Values:	`true`, `false`
Dependencies:	Site name added to `Place Group Annex` tab.

This option allows the information presented in the historical reports to be grouped by Sites in a multi-site environment. Sites used by the G*plus* Adapter are defined by adding Site names to the `Annex` section options of the `Place Group` configurations. The Site name will then prefix the `Queue` name in the `Group` column of the Contact Statistics reports.

This Option only applies to the stream defined in the `historical.reports` Section. In the case of multiple streams, each additional historical stream has its own `siteBased` Option.

Note: Contact Professional Services prior to enabling this option.

tenantDbid

Type:	Optional
Default Value:	`101 (Resources)`
Valid Values:	A `dbid` of a configured Tenant
Dependencies:	Configuration Server tenant type

This option must be set if the G*plus* Adapter is connected to a Genesys Multi-Tenant Configuration Server and the Adapter will be monitoring a different `Tenant` than the default `Resources Tenant` that usually corresponds to a `dbid` of `101`.

timeZone

Type:	Optional
Default Value:	Time zone of the application’s host server.
Valid Values:	Listed in Appendix A.
Dependencies:	None.

This option is used to report data for a time zone other than the time zone where the G*plus* Adapter instance is installed. It is not required if the server time zone is correct for the locale.

Note: See Appendix A - Time Zone ID List.

useLocalTime

Type:	Optional
Default Value:	`auto`
Valid Values:	`auto`, `true`, `false`
Dependencies:	None

The `useLocalTime` option allows the G*plus* Adapter to set the event time for received events using the local server time. This provides coordination between two or more servers where the event times create synchronization problems. One example would be with the Interaction Server that retains event time in seconds while `voice` TServers maintain event time to a resolution of milliseconds. Synchronization between the two server types is necessary for monitoring events in Outbound Push Preview campaigns.

If the option is left as `auto`, the adapter will automatically use local time if it is connected to both an Interaction Server and a `voice` TServer. This can be disabled by setting the option to `false`.

Note: Contact Professional Services prior to configuring this option.

verint.includeEmptyQueues

Type:	Optional
Default Value:	`false`
Valid Values:	`true`, `false`
Dependencies:	None.

Enabling this option results in all of the monitored `Groups` being included in the Contact Statistic reports even if there was no call or agent activity associated with those `Groups`.

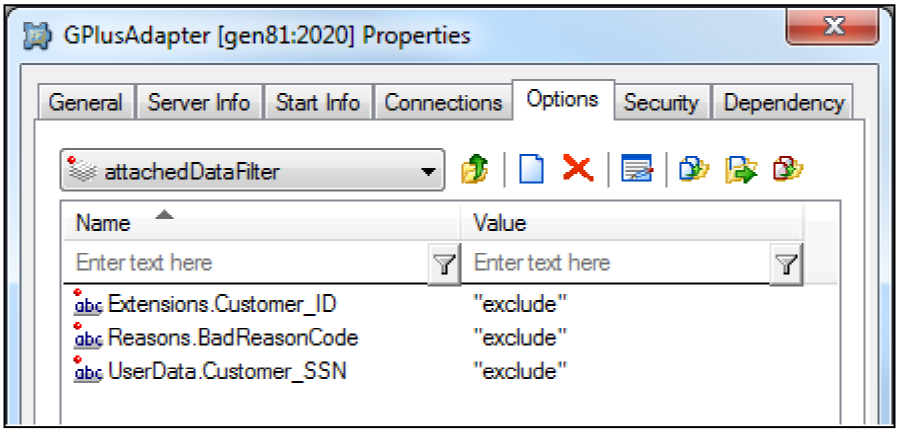
attachedDataFilter Section

This Option Section is used to remove sensitive or private information from the received events prior to those events being processed or logged. Key/value pairs can be removed from the following Key/Value List attributes:

- `UserData`
- `Reasons`
- `Extensions`

The only allowed Option value is "`exclude`" and the Option name is the event attribute to excluded. Nested attributes will be removed if their *root* is identified in Option name.

It should be noted that the attribute will not be available to the Adapter if removed. For example, if a `UserData.VQ_Name` attribute was being used to identify a target for a customer's queued call and all of the `UserData` had been removed, the target determiner would fail.



Name	Value
Extensions.Customer_ID	"exclude"
Reasons.BadReasonCode	"exclude"
UserData.Customer_SSN	"exclude"

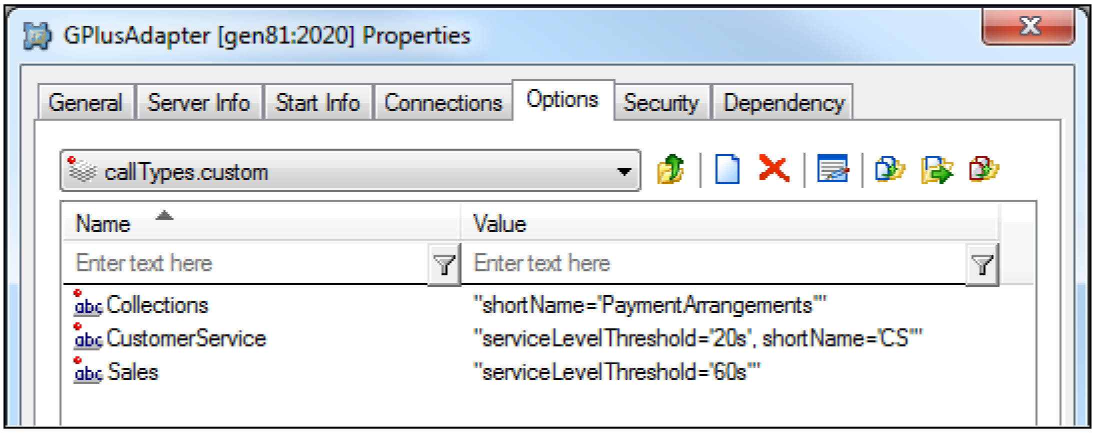
callTypes.custom Section

This optional Section is only used if the `application/callType` Option has been set to `custom`. The section contains a list of WFM `voice` contact type/queue names that will be reported by the G*plus* Adapter. This list is required when there is no one-to-one mapping of expected WFM contact types/queues to specific Configuration Objects (`Skills` or `Virtual Queues`).

The `Option Name` (or `shortName`) must match the exact value generated by the G*plus* Adapter when applying the event filter specified by the `callType.determiner` Option against the TServer queued events. The `Option Value` can be left blank or a configuration text string specified. Since this is essentially a custom list of contact types rather than Genesys Configuration Objects, the `serviceLevelThreshold` (see Threshold Format) and/or a `shortname` can be set in a manner similar to that of the Virtual Queues.

If no `serviceLevelThreshold` is specified, the G*plus* Adapter will use the `defaultServiceThreshold` Option from the `voice` Section of the adapter Options. If no `shortname` is specified, the G*plus* Adapter will report the `Option Name`.

Type:	Optional
Default Value:	*Section is not used*
Valid Values:	`serviceLevelThreshold` and `shortName`
Dependencies:	`application/callType`, `event.properties/callType.determiner`



Name	Value
Collections	"shortName='PaymentArrangements'"
CustomerService	"serviceLevelThreshold='20s', shortName='CS'"
Sales	"serviceLevelThreshold='60s'"

chat Section

chatACWReason

Type:	Optional
Default Value:	`ACW`
Valid Values:	Any unique text value.
Dependencies:	Agent's `chat` client must be able to set the `Reason` field when going `NotReady`.

This option allows users to specify a `NotReady ReasonCode` that will be reported as the equivalent of `AfterCallWork` for the `chat` statistics in the reports and will be reported as the `afterChatWork(1502)` RTA state..

The text string must be contained in the `Reason.ReasonDescription` attribute of a `chat NotReadyForMedia` event.

defaultAbandonedThreshold

Type:	Optional
Default Value:	`0s`
Valid Values:	See Threshold Format
Dependencies:	`historical.reports/serviceLevelFormula`

This option specifies the default abandoned threshold for the service level calculations that form part of the interval reports. Only two of the service level formulae actually use the abandoned threshold so this option has no effect unless one of those formulae (1 or 4) has been selected. A `chat` is considered abandoned if the customer leaves the `chat` session prior to the agent joining the session.

This option should be set when a single abandoned threshold is used as a default for all `chat` targets. Individual abandoned thresholds can be set for each `chat` target/queue by adding `Annex Data` to the `chat Virtual Queues` as shown in a later section or adding the threshold to an `Interaction Queue` as described in the interactionTargets Section. These individual thresholds will override the default specified with this option.

defaultServiceThreshold

Type:	Optional
Default Value:	`0s`
Valid Values:	See Threshold Format
Dependencies:	None

This option specifies the default service threshold for the service level calculations that form part of the interval reports. A `chat` is considered *serviced* at the time that the agent first joins the `chat` session with a customer.

This option should be set when a single service threshold is used as a default for all `chat` targets. Individual service thresholds can be set for each `chat` target/queue by adding `Annex Data` to the `chat` `Virtual Queues` as shown in a later section or adding the threshold to an `Interaction Queue` as described in the interactionTargets Section. These individual thresholds will override the default specified by this option.

defaultTarget

Type:	Optional
Default Value:	`UnknownTarget`
Valid Values:	Any text
Dependencies:	`application/countUnattachedACW`

This option allows users to define a separate `chat` target/queue other than the default value used by the G*plus* Adapter to report on an agent's time outside of actual `chat` activity (including unattached `ACW`) when that time cannot be associated with an actual `chat` target/queue.

enabled

Type:	Mandatory
Default Value:	`false`
Valid Values:	`true, false`
Dependencies:	A connection to a Genesys Interaction Server.

Set to `true` to capture and report on `chat` activity (historical and RTA) in the contact center.

interaction.reason.determiner

Type:	Optional
Default Value:	*Not set*

Valid Values:	`eventAttribute:` {Event attribute name}
	`class:` {custom Java class name}

Dependencies:	`chat/enabled, rta/enabled`

This option specifies the attribute of a `chat` Interaction Server Event that will be included as the Reason field of an RTA state message generated by one of the following events:

- `EventPartyAdded`
- `EventPropertiesChanged` while the agent is a party to a routed `chat`.

Certain implementations may require that the combination of an `InRoutedChat` RTA state and the reason code map to a different RTA state and reason text string. This mapping can be configured in the interactionCodeMappings Section.

Note: Contact Professional Services before using this option.

reason.notReady.determiner

Type:	Optional
Default Value:	`eventAttribute: ReasonDescription`

Valid Values:	`eventAttribute:` {Event attribute name}
	`class:` {custom Java class name}

Dependencies:	`chat/enabled, rta/enabled`

This option specifies the attribute of a `chat` Interaction Server Event that will be included as the Reason field of an RTA state message generated by one of the following events:

- `EventNotReadyForMedia`
- `EventMediaStateReasonChanged` while the agent is NotReady to receive a routed `chat`.

Certain implementations may require that the combination of an `Unavailable` RTA state and the reason code map to a different RTA state and reason text string. This mapping can be configured in the reasonCodeMappings Section

Note: Contact Professional Services before using this option.

reason.ready.determiner

Type:	Optional
Default Value:	*Not set*

Valid Values:	`eventAttribute:` {Event attribute name}
	`class:` {custom Java class name}

Dependencies:	`chat/enabled, rta/enabled`

This option specifies the attribute of a `chat` Interaction Server Event that will be included as the Reason field of an RTA state message generated by one of the following events:

- `EventReadyForMedia`

Certain implementations may require that the combination of an `Available` RTA state and the reason code map to a different RTA state and reason text string. This mapping can be configured in the readyCodeMappings Section.

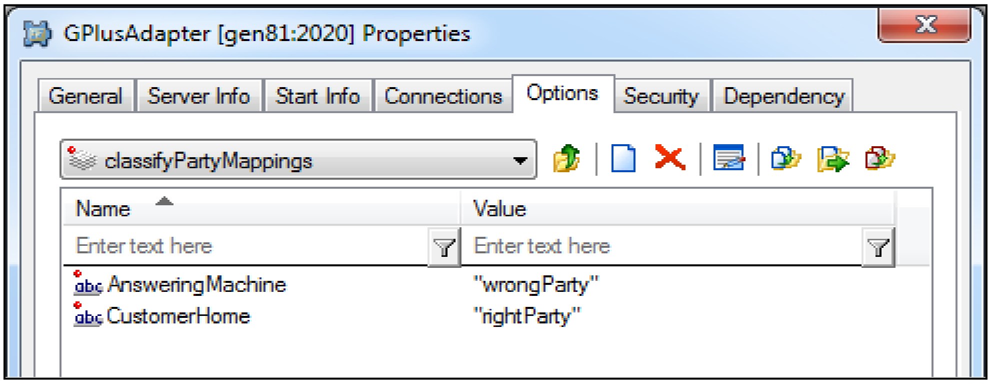
`Note:` Contact Professional Services before using this option.

classifyPartyMappings Section

This section is optional and must be added manually as it is not included in the Application Template. When the section is included, the Outbound Campaign disposition codes parsed by the outbound/classifyPartyDeterminer will be compared against the Option key names in this section. If there is a match, the disposition code will be mapped to one of two alternatives specified in the Option value:

- `rightParty`
- `wrongParty`

If the disposition code does not match one of the listed Options then the call is considered to be `wrongParty` and will not be counted against the "Right Party" columns in the Outbound Statistics report.



Name	Value
AnsweringMachine	"wrongParty"
CustomerHome	"rightParty"

email Section

defaultServiceThreshold

Type:	Optional
Default Value:	`0s`
Valid Values:	See Threshold Format
Dependencies:	None

This option specifies the default service threshold for the service level calculations that form part of the interval reports. An `email` is considered *serviced* at the time that the agent sends the reply to a customer.

This option should be set when a single service threshold is used as a default for all `email` targets. Individual service thresholds can be set for each `email` target/queue by adding `Annex Data` to the `email` `Virtual Queues` as shown in a later section or adding the threshold to an `Interaction Queue` as described in the interactionTargets Section. These individual thresholds will override the default specified by this option.

defaultTarget

Type:	Optional
Default Value:	`UnknownTarget`
Valid Values:	Any text
Dependencies:	`application/countUnattachedACW`

This option allows users to define a separate `email` target/queue other than the default value used by the G*plus* Adapter to report on an agent's time outside of actual `email` activity (including unattached `ACW`) when that time cannot be associated with an actual `email` target/queue.

emailACWReason

Type:	Optional
Default Value:	`ACW`
Valid Values:	Any unique text value.
Dependencies:	Agent’s `email` client must be able to set the `Reason` field when going `NotReady`.

This option allows users to specify a `NotReady ReasonCode` that will be reported as the equivalent of `AfterCallWork` for the `email` statistics in the reports and will be reported as the `afterEmailWork (1104)` RTA state..

The text string must be contained in the `Reason.ReasonDescription` attribute of an `email NotReadyForMedia` event.

enabled

Type:	Mandatory
Default Value:	`false`
Valid Values:	`true, false`
Dependencies:	A connection to a Genesys Interaction Server.

Set to `true` to capture and report on `email` activity (historical and real-time adherence) in the contact center.

interaction.reason.determiner

Type:	Optional
Default Value:	*Not set*

Valid Values:	`eventAttribute:` {Event attribute name}
	`class:` {custom Java class name}

Dependencies:	`email/enabled, rta/enabled`

This option specifies the attribute of an `email` Interaction Server Event that will be included as the Reason field of an RTA state message generated by one of the following events:

- `EventPartyAdded`
- `EventPropertiesChanged` while the agent is a party to a routed `email`.

Certain implementations may require that the combination of a `ReadingCustomerEmail` or `ReplyingToCustomerEmail` RTA state and the reason code map to a different RTA state and reason text string. This mapping can be configured in the interactionCodeMappings Section.

`Note:` Contact Professional Services before using this option.

reason.notReady.determiner

Type:	Optional
Default Value:	`eventAttribute: ReasonDescription`

Valid Values:	`eventAttribute:` {Event attribute name}
	`class:` {custom Java class name}

Dependencies:	`email/enabled, rta/enabled`

This option specifies the attribute of an `email` Interaction Server Event that will be included as the Reason field of an RTA state message generated by one of the following events:

- `EventNotReadyForMedia`
- `EventMediaStateReasonChanged` while the agent is NotReady to receive a routed `email`.

Certain implementations may require that the combination of an `Unavailable` RTA state and the reason code map to a different RTA state and reason text string. This mapping can be configured in the reasonCodeMappings Section

Note: Contact Professional Services before using this option.

reason.ready.determiner

Type:	Optional
Default Value:	*Not set*

Valid Values:	`eventAttribute:` {Event attribute name}
	`class:` {custom Java class name}

Dependencies:	`email/enabled, rta/enabled`

This option specifies the attribute of a `email` Interaction Server Event that will be included as the Reason field of an RTA state message generated by one of the following events:

- `EventReadyForMedia`

Certain implementations may require that the combination of an `Available` RTA state and the reason code map to a different RTA state and reason text string. This mapping can be configured in the readyCodeMappings Section.

Note: Contact Professional Services before using this option.

event.properties Section

callType.determiner

Type:	Optional
Default Value:	`eventAttribute: ThisQueue`

Valid Values:	`eventAttribute:` {`EventQueued` attribute name}
	`class:` {custom Java class name}

Dependencies:	`voice/enabled`, `application/callType`

This option specifies what part (attribute) of an `EventQueued` will be used to determine the contact queue/call target for a routed inbound call.

Examples:

```
eventAttribute: ThisQueue
eventAttribute: UserData.VQ_Name
```

campaignTarget.determiner

Type:	Optional
Default Value:	`eventAttribute: UserData.GSW_CAMPAIGN_NAME`
Valid Values:	`eventAttribute:` {`EventEstablished` attribute name}
Dependencies:	`outbound/enabled`

This option specifies what part (attribute) of an `EventEstablished` on an agent's `outbound` campaign call will be used to determine the contact queue/call target for that call. There will rarely be a reason to use anything other than the default.

See Outbound Campaign Calls for more information.

Note: Contact Professional Services prior to configuring this option.

ignoreAbandonOnRinging

Type:	Optional
Default Value:	`false`
Valid Values:	`true, false`
Dependencies:	None

This option specifies whether the adapter will continue to track an inbound voice call that is Abandoned while ringing at an agent's Place. Normal behavior would see the call records terminated at that point but in certain *Route On No Answer* (RONA) situations, the Abandoned event is generated when the call is pulled back and requeued. This option allows for those calls to be reported correctly.

Note: Calls that are actually abandoned by the customer releasing the call while ringing will be counted incorrectly because the call is not terminated if this option is enabled. This option should only be enabled after considering the effect on the report data.

ignoreRONAEventAttribute

Type:	Optional
Default Value:	*Not set*
Valid Values:	`UserData.`{RONA identifier field name}
Dependencies:	None

This option specifies the attached data field that the G*plus* Adapter will look for to avoid double counting calls in a Route on No Answer (RONA) scenario. This option is required when a RONA call is answered at the agent's desktop and immediately transferred back to the queue to reroute the call. Attaching and removing this `UserData` field for specific events is critical for the accurate reporting of routing with RONA calls.

Note: Contact Professional Services prior to configuring this option.

interactionTarget.determiner

Type:	Optional
Default Value:	`eventAttribute: ThisQueue`

Valid Values:	`eventAttribute:` {`EventPlacedInQueue` attribute name}
	`class:` {custom Java class name}

Dependencies:	`chat/enabled`, `email/enabled`

This option specifies how an `email` and/or `chat` target is determined from the attributes associated with the Interaction event.

Examples:	`eventAttribute: ThisQueue`
	`eventAttribute: interactionProperties.interactionQueue`
	`eventAttribute: UserData.VQ_Name`

reason.logout.determiner

Type:	Optional
Default Value:	`eventAttribute: Extensions.ReasonCode`
Valid Values:	`eventAttribute:` {`EventAgentLogout` attribute name}
Dependencies:	`voice/enabled`, `rta/enabled`

This option specifies one of the attributes of a `voice EventAgentLogout` that identifies the reason code associated with an agent logging out. If the reason code is present, it will be included as the reason field in the agent's RTA state transition.

reason.notReady.determiner

Type:	Optional
Default Value:	`eventAttribute: Reasons.ReasonCode`

Valid Values:	`eventAttribute:` {`EventAgentNotReady` attribute name}
	`class:` {custom Java class name}

Dependencies:	`voice/enabled`, `rta/enabled`, `reasonCodeMappings`

This option specifies attribute(s) of an `EventAgentNotReady` that identify the reason associated with an agent entering an `Unavailable` RTA state. Certain implementations may require that the combination of `EventAgentNotReady` and the reason code map to a different RTA state and reason text string. This mapping can be configured in the reasonCodeMappings Section

> Note: Contact Professional Services before changing this option from its Default Value.

serviceDate.determiner

Type:	Optional
Default Value:	*Not set*
Valid Values:	`eventAttribute:` {attribute name}
Dependencies:	None

This option specifies attribute of in a queued event that contains a “to be serviced by” timestamp for the queued interaction. If the attribute is not present, then the defaultServiceThreshold for the media or the configured serviceLevelThreshold for the queue will be used.

The attribute value is a date/time combination that indicates the time that interaction must be serviced by to meet the Service Level expectations. The attribute string must be in the following format and the time is assumed to GMT rather than the local time:

- Format - `"YYYY-MM-DDThh:mm:ssZ"`
- Example - `"2013-12-31T09:30:00Z"`

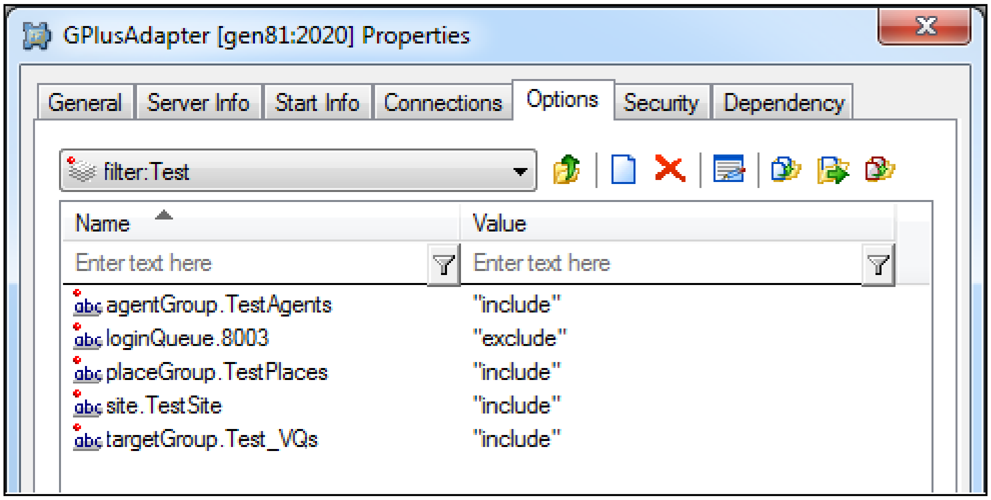
filter:<filterName> Section

There can be more than one instance of this Section type and each is differentiated and made unique by the `<filterName>` portion of the Section name. The purpose of this Section is to define a subset of the monitored objects that have already been defined in the genesys.agentGroups , genesys.placeGroups and genesys.queueGroups Sections. Filters combined with the Stream configurations (historical and/or rta) allow the G*plus* Adapter to support multiple Verint servers if required.

The filter is equivalent to the group sections listed above with `Agent Groups`, `Place Groups` and `VQ DN Groups` included or excluded as required. The addition of the `loginQueue` option provides an alternative method for including or excluding groups of Agents. The `site` option allows a set of `Place Groups` to be defined in much the same way. Sites and `Place Groups` are discussed below. The syntax for the group options is as follows:

- `agentGroup.<AgentGroupName>` : `include` or `exclude`
- `placeGroup.<PlaceGroupName>` : `include` or `exclude`
- `loginQueue.<ACDQueue DN>` : `include or exclude`
- `site.<SiteName>` : `include` or `exclude`
- `targetGroup:<DNGroupName>` : `include` or `exclude`

Note: The groups that are included or excluded must correspond to groups already included in the three `genesys.*Groups` Sections. This will require attention as to how the `AgentGroups`, `PlaceGroups` and `DNGroups` (`VQs`) are defined. . The `loginQueue` option(s) is the exception as each option references a DN rather than an `AgentGroup`.



filter:Test

Name	Value
agentGroup.TestAgents	"include"
loginQueue.8003	"exclude"
placeGroup.TestPlaces	"include"
site.TestSite	"include"
targetGroup.Test_VQs	"include"

genesys.agentGroups Section

This Section is a list of the Agent Groups that will be included in or excluded from the group of agents being monitored by the G*plus* Adapter. Valid Option names are either an asterisk (*) or the name of an Agent Group defined in the Configuration Server. The Option value is the action to be taken with respect to the named Agent Group: include or exclude.

*****

Type:	Optional
Default Value:	include
Valid Values:	include
Dependencies:	None

OR

<Agent Group Name>

Type:	Optional
Default Value:	*None*
Valid Values:	include, exclude
Dependencies:	Configuration Server Agent Groups.

The Option Name determines the agents that the G*plus* Adapter will report upon. An agent that is a member of an excluded or not included Agent Group will have no activity statistics reported in the Agent Scorecard Metric report. Additionally, an excluded agent's call activity will not be included in the Contact Statistics reports; a queued call would still be counted as "Offered" but it would not be counted as "Handled".

An excluded agent's activity will not be reported in the RTA data stream.

The asterisk (*) as the Option Name can be considered to be an Agent Group containing all of the Persons configured as an Agent with an enabled state in the Configuration Server. Including all of the agents is the default configuration and there is little reason to change that unless there is a specific configuration requirement.

The only alternative is to have an Option Name that corresponds to one of the defined Agent Groups in the Configuration Server. Any of those Agent Groups can be excluded or included.

genesys.placeGroups Section

This Section is a list of the Place Groups that will be included in or excluded from the group of Places being monitored by the G*plus* Adapter. Valid Option names are either an asterisk (*) or the name of a Place Group defined in the Configuration Server. The Option value is the action to be taken against the named Place Group: include or exclude.

*

Type:	Optional
Default Value:	include
Valid Values:	include, exclude
Dependencies:	None

OR

<Place Group Name>

Type:	Optional
Default Value:	*None*
Valid Values:	include, exclude
Dependencies:	Configuration Server Place Groups.

The Option Name determines the Place Group that the G*plus* Adapter will monitor or not monitor. This will mean that all of the DNs associated with the Places in those groups will be registered with the connected voice TServer(s) and all of the Places will be registered with the connected Interaction Server(s).

Specifying an asterisk (*) as the Option Name can be considered to be a Place Group containing all of the Places configured in the Configuration Server. The only alternative is to have an Option Name that corresponds to one of the defined Place Groups in the Configuration Server.

Place Groups have two roles in the configuration of the G*plus* Adapter: identifying the Places that are to be monitored by the application and associating groups of Places with Site names. Sites and Place Groups are discussed in another section.

Note: When IVR Ports or other non-Agent Places are defined within the Configuration Server, it is common to exclude these places with the use of a Place Group so that IVR (non-Agent) activity is not reported.

genesys.queueGroups Section

This Section is a list of the DN Groups that will be included in or excluded from the group of Virtual Queue DNs (VQs) being monitored by the G*plus* Adapter. Valid Option names are either an asterisk (*) or the name of a DN Group defined in the Configuration Server. The Option value is the action to be taken against the named DN Group: include or exclude.

* (Option Name)

Type:	Optional
Default Value:	include
Valid Values:	include
Dependencies:	None

OR

<VQ DN Group Name>

Type:	Optional
Default Value:	*None*
Valid Values:	include, exclude
Dependencies:	VQ DN Groups in the Configuration Server.

The Option Name determines the VQs that the G*plus* Adapter will monitor. Specifying an asterisk (*) as the Option Name can be considered to be a DN Group containing all of the VQ DNs configured in the Configuration Server. The only alternative is to have an Option Name that corresponds to one of the defined DN Groups in the Configuration Server.

This option is used to specify monitored VQs that are defined for an actual Switch object in the Configuration Server. VQs defined on a Multimedia_Switch can be included but multimedia Interaction Queues defined in a Business Process routing strategy cannot.

historical.ftp Section

This section lists the configuration options for transferring historical report files from the server hosting the G*plus* Adapter to a folder on the server hosting the WFM Application that uses those files. These options allow the transfer protocol to be configured as plain FTP or to use SSH (Secure Shell) to transfer the files.

With the addition of multiple Streams, this Section should be set up to transfer the report files associated with the Stream defined in the historical.reports Section.

backupRetentionDays

Type:	Mandatory
Default Value:	30
Valid Values:	Any positive integer
Dependencies:	`localBackupDirectory`

This option determines the number of days that historical report files created by the G*plus* Adapter will be retained in the directory specified in the `localBackupDirectory` Option. The report files are removed permanently from the back up folder as they expire.

connectionMode

Type:	Optional
Default Value:	`active`
Valid Values:	`active, passive`
Dependencies:	Ignored if `secureTransfer` is enabled.

This option specifies the ftp connection mode used by the ftp client to connect to the remote ftp server. Some network firewalls will not allow active ftp connections and the connection mode can be changed to passive to allow the report files to transfer through the firewall.

enabled

Type:	Mandatory
Default Value:	`true`
Valid Values:	`true, false`
Dependencies:	None

This option can disable the ftp client that forwards the report files to the WFM server. This option is often set to `false` for backup G*plus* Adapter applications that do not need to send report files on a continuous basis but only in the event of a mishap.

keyPath

Type:	Optional
Default Value:	*Not set*
Valid Values:	Path to SSH private key file
Dependencies:	`secureTransfer`

This option defines the full path (including the file name) for the private key file used to establish the SSH connection with the remote server when the `secureTransfer` Option is enabled. The key file should usually be placed in the Adapter's `config` folder.

keyPassphrase

Type:	Optional
Default Value:	*Not set*
Valid Values:	Valid text string.
Dependencies:	`secureTransfer`

This pass phrase provides additional security for the SSH connection as establishing the connection requires the providing the pass phrase as well as possessing the private key file. The pass phrase must be specified when the SSH keys are created.

localBackupDirectory

Type:	Mandatory
Default Value:	`ftp/reportsBackup`
Valid Values:	Any existing folder on the G*plus* Adapter server.
Dependencies:	None

This option identifies the directory on the G*plus* Adapter server where copies of the historical report files will be saved after those files have been successfully transferred. If the option value is left blank, the report files will be deleted locally after being transferred to the remote WFM server.

localSourceDirectory

Type:	Mandatory
Default Value:	`ftp/reports`
Valid Values:	Any existing folder path.
Dependencies:	`historical.reports/directory`

This option identifies the directory where files will be temporarily stored at each reporting interval prior to being transferred to the WFM Server and subsequently moved to the backup directory or deleted if the backup directory is not configured.

The default location is the default report location specified in the `historical.reports` Option section and setting the two options to the same folder is how the association between the ftp client and the historical Stream is established.

remoteDestinationDirectory

Type:	Mandatory
Default Value:	`./`
Valid Values:	Any folder path on the remote server relative to the ftp user’s default directory
Dependencies:	None

This option identifies the relative directory on the remote WFM Server where the G*plus* Adapter report files will be transferred. This directory is relative to the configured default directory for the ftp user identified in the `userName` Option.

Note: The ftp user must have sufficient access privileges to write the transferred files to the folder identified by the relative path.

remoteHost

Type:	Mandatory
Default Value:	*Not set*
Valid Values:	Host Name or IP Address of the Verint Server
Dependencies:	None

This option specifies the host name or IP address of the server to which the report files will be transferred.

remotePort

Type:	Mandatory
Default Value:	21
Valid Values:	FTP or SSH Port on the remote WFM Server
Dependencies:	`secureTransfer` Option

This option allows users to define the port on the remote WFM Server to something other than the default FTP port of `21`. For instance, the default SSH port is `22` and the option should be set to that when `secureTransfer` is enabled.

secureTransfer

Type:	Mandatory
Default Value:	`false`
Valid Values:	`true, false`
Dependencies:	`keyPath, keyPassphrase, remotePort`

This option specifies that the file transfer will be made over an SSH (Secure Shell) connection to the remote server rather than the default FTP protocol. A private key file must be created and stored on the server hosting the G*plus* Adapter and the public key must be registered on the server hosting the WFM reporting application. Configuring an SSH connection is discussed below and Appendix B contains troubleshooting tips when problems occur.

userName

Type:	Mandatory
Default Value:	*Not set*
Valid Values:	Any alphanumeric user name registered on the remote server.
Dependencies:	None

This option specifies the username used to log in to the destination FTP server.

userPassword

Type:	Mandatory
Default Value:	*Not set*
Valid Values:	Any alphanumeric password
Dependencies:	User identified by `userName`.

This option specifies the password for the user identified by the `userName` Option and is required to establish the connection to the ftp server.

historical.reports Section

dailyReportEnabled

Type:	Optional
Default Value:	`true`
Valid Values:	`true, false`
Dependencies:	None

This option specifies whether the daily Agent Scorecard Metric report will be generated by the G*plus* Adapter. Daily reports are not required for all WFM integrations. The reports are contained in the `Agent_YYYYMMdd.txt` files.

directory

Type:	Mandatory
Default Value:	`ftp/reports`
Valid Values:	Any valid folder on the G*plus* Adapter server
Dependencies:	None

This option specifies the directory where the report files will be saved as they are generated. Typically, this will be the directory monitored by the Adapter’s file transfer client and the files will be immediately transferred to the remote server.

enabled

Type:	Mandatory
Default Value:	`true`
Valid Values:	`true, false`
Dependencies:	None

This option allows users to disable all historical reports for this particular Stream when only RTA data is required from the G*plus* Adapter.

filter

Type:	Optional
Default Value:	*Not set*
Valid Values:	A `filterName` (`filter:<filterName>` Section)
Dependencies:	A filter Section must be defined.

This option allows users to associate the options defined in this Section with a defined `filter` section by setting the value to `filterName`. If multiple streams are not defined or the reports are going to include all of the `Agents`, `Places` and `VQs` defined in the `genesys.*Groups` Sections, then this option can be left blank.

includeOutboundInternalInNOC

Type:	Optional
Default Value:	`false`
Valid Values:	`true, false`
Dependencies:	None

This option specifies which non-routed calls made by an agent will be reported on agent reports as outbound calls. The default behavior is report only the Genesys `CallType` of `CallOutbound` (external to the switch) as outbound calls. If this option is set to true, all non-routed calls (e.g. `CallInternal`) dialed by the agent are considered outbound calls.

intervalReportEnabled

Type:	Mandatory
Default Value:	`true`
Valid Values:	`true, false`
Dependencies:	None

This option specifies whether the four following reports will be generated by the G*plus* Adapter on the interval specified by the `intervalReport.interval` Option:

- Voice Contact Statistics Report
- Outbound Statistics Report
- Email Contact Statistics Report
- Chat Contact Statistics Report

intervalReport.interval

Type:	Mandatory
Default Value:	`15`
Valid Values:	`15, 30, 60`
Dependencies:	None

This option specifies the period in minutes between interval reports. This option should be set to whatever the duration the WFM solution has been configured for. The reports are generated at set times in the hour depending upon the interval so a `15` minute interval would see reports created at `0:00`, `0:15`, `0:30` and `0:45` in each hour

reportRetentionDays

Type:	Mandatory
Default Value:	`30`
Valid Values:	Any positive integer
Dependencies:	`historical.ftp/localBackupDirectory`

This option specifies the number of days that copies of the historical reports will be retained in the `localBackupDirectory` after the originals have been transferred to the remote server. If the `localBackupDirectory` has not been set, the local copies of the report files will not be backed up after being successfully transferred.

serviceLevelFormula

Type:	Mandatory
Default Value:	`3`
Valid Values:	`1 - 4` (formula index listed below)
Dependencies:	`serviceLevelThreshold, abandonedThreshold`

This option specifies which formula the G*plus* Adapter will be used to calculate the Service Level Percentage in the Contact Statistics reports.

Formulae:

1. `(callsAnsweredInThreshold + callsAbandonedInThreshold) / (callsAnswered + callsAbandoned)`
2. `callsAnsweredInThreshold / callsAnswered`
3. `callsAnsweredInThreshold / (callsAnswered + callsAbandoned)`
4. `callsAnsweredInThreshold / (callsAnswered + callsAbandonedAfterThreshold)`

The `serviceLevelThreshold` that can be set in the related media section and/or individually for each Target. An `abandonedThreshold` can also be set in the same way to facilitate using Formula `1` or Formula `4` above.

verint.campaignReportEnabled

Type:	Optional
Default Value:	`true`
Valid Values:	`true, false`
Dependencies:	`outbound/enabled, intervalReportEnabled`

This option can be used to disable the reporting of outbound campaign calls in the Outbound Statistics report. The Outbound Campaign reports are contained in the `Campaign_MMddhhmm.txt` files.

verint.chatReportEnabled

Type:	Optional
Default Value:	`true`
Valid Values:	`true, false`
Dependencies:	`chat/enabled, intervalReportEnabled`

This option can be used to disable the reporting of `chat` interactions in the Chat Contact Statistics report. The reports are contained in the `Chat_MMddhhmm.txt` files.

verint.emailReportEnabled

Type:	Optional
Default Value:	`true`
Valid Values:	`true, false`
Dependencies:	`email/enabled, media:<name>/enabled, intervalReportEnabled`

This option can be used to disable the reporting of `email` and /or `iWD/OpenMedia` interactions in the Email Contact Statistics report. The generic media are included in the `email` reports because Verint does not yet have a report type for generic media.

These reports are contained in the `Email_MMddhhmm.txt` files.

verint.voiceReportEnabled

Type:	Optional
Default Value:	`true`
Valid Values:	`true, false`
Dependencies:	`voice/enabled, intervalReportEnabled`

This option can be used to disable the reporting of inbound voice calls in the Voice Contact Statistics report. The reports are contained in the `Voice_MMddhhmm.txt` files.

historical:<streamName> Section

This Section is optional and is used when the G*plus* Adapter instance is providing more than one Stream. This Section is used to configure the historical component of the data feed to the Verint Server. The Options contained in this section combine the Options from both the historical.reports and historical.ftp Sections with a few additions. The Options are listed below and the additions are discussed.

Report Options

- dailyReportEnabled
- enabled
- filter
- includeOutboundInternalInNOC
- intervalReportEnabled
- intervalReport.interval
- reportRetentionDays
- serviceLevelFormula
- siteBased

  The application/siteBased Option only applies to the stream defined in the `historical.reports` Section. This stream Option allows the site based reports to be configured on a stream by stream basis.
- verint.campaignReportEnabled
- verint.chatReportEnabled
- verint.emailReportEnabled
- verint.voiceReportEnabled

There is no `directory` option for the Stream configuration as the adapter automatically creates subdirectories in the directory configured by the historical.reports/directory Option. The created stream directory uses this Section's `streamName` as its name.

FTP Options

- ftp.backupRetentionDays
- ftp.connectionMode
- ftp.enabled

  This Option is the surrogate for the `historical.ftp/enabled` Option that enables (value = `true`) or disables (value = `false`) the transfer of the historical reports to the remote Verint server.
- ftp.keyPath
- ftp.keyPassphrase
- ftp.localBackupDirectory
- ftp.localSourceDirectory
- ftp.remoteDestinationDirectory
- ftp.remoteHost
- ftp.remotePort
- ftp.secureTransfer
- ftp.userName
- ftp.userPassword

interactionCodeMappings Section

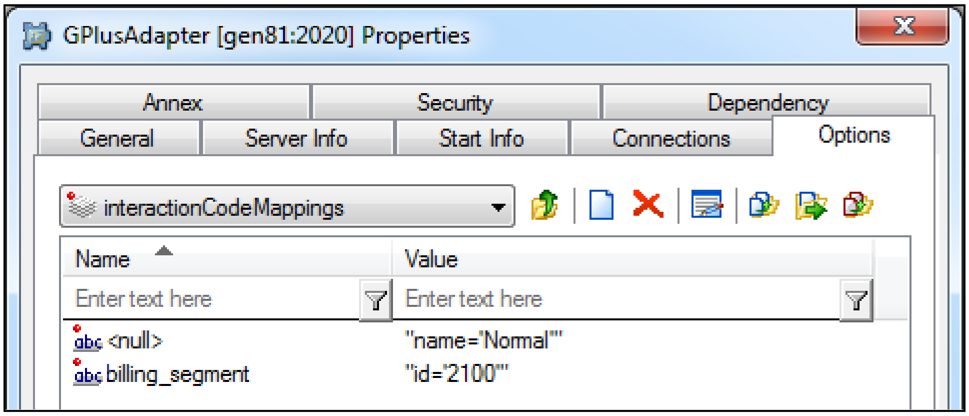
This section is optional and must be added manually as it is not included in the Application Template. When the section is included, the reasons parsed by the `interaction.reason.determiner(s)` will be compared against the Option key names in this section. If there is a match, the parameters listed in the Option value will be used. There are two possible parameters:

1.	`id`	new RTA state.
2.	`name`	new reason code.

The `id` field can be either the new RTA State Name or State ID as defined in the RTA State table. The `name` field is the new Reason that will be included with the RTA state message sent to the Verint server.

The `<null>` Option key signifies an event with no reason attached and it is used to configure a default behavior for the *Interaction* events.

Note: Contact Professional Services before configuring this section.



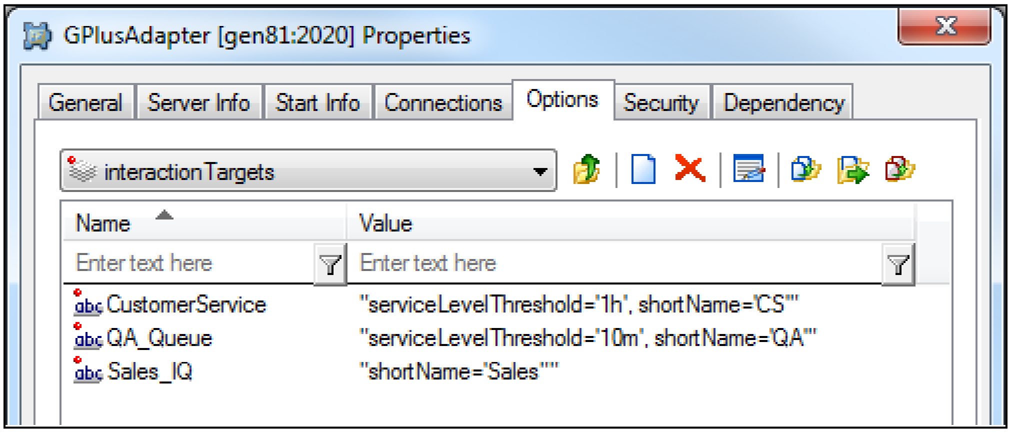
Name	Value
<null>	"name='Normal'"
billing_segment	"id='2100'"

interactionTargets Section

This Section is optional but it is required when Interaction Queues are used within Business Process Routing Strategies as the reporting objects. If only Virtual Queues defined on a Switch are used as reporting objects, this Section is not required. The situation is slightly different for email or other media where the interactions must also be tracked after they leave the agent's desktop. In that situation, outbound queues or Quality Assurance queues must also be monitored. This is explained in more detail in the following section.

The G*plus* Adapter will report on the list of Interaction Queues contained within this section. Each Option name represents the specific name of an actual Interaction Queue defined in the Configuration Server. For each Interaction Queue, the Option value can be used to specify the serviceLevelThreshold (See Threshold Format) and the shortName for the Interaction Queue that will be used in the reports sent to the WFM application. It should be noted that the shortName field does not actually have to be shorter than the Interaction Queue name.

Type:	Optional
Default Value:	*Not Set*
Valid Values:	See example below
Dependencies:	email/enabled, chat/enabled, media:<name>/enabled



Name	Value
CustomerService	"serviceLevelThreshold='1h', shortName='CS'"
QA_Queue	"serviceLevelThreshold='10m', shortName='QA'"
Sales_IQ	"shortName='Sales'"

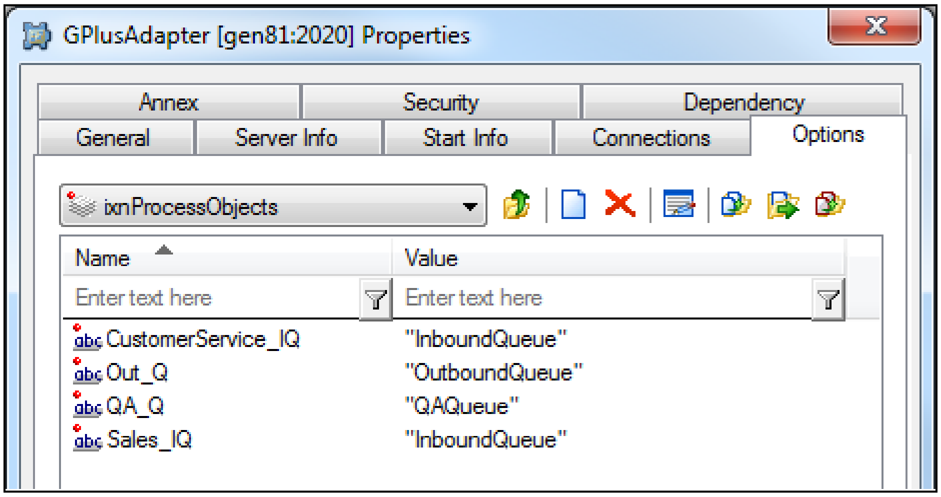
ixnProcessObjects Section

The G*plus* Adapter uses the list of `Interaction Queues` contained in this Section to determine how to report on media activity at various routing stages. The Option values categorize the monitored `Interaction Queues` as to their role in the Genesys `Business Process` Routing Strategies.

Each Option name within this section represents the specific name of an `Interaction Queue` configured as a `Script` in the Configuration Server. The Option Value associates the queue with one of the routing strategy roles defined by the G*plus* Adapter.

All `Interaction Queues` involved with outbound media interactions, transferring a routed media interaction or holding an interaction for quality assurance routing must be added and categorized in this Section with one exception: `Interaction Queues` where media interactions are parked and waiting to be routed to an agent (e.g. an `InboundQueue`) should only be added if `Interaction Queues` instead of `Virtual Queues` will be used as the reporting "queue" object on the interval report.

Type:	Mandatory for `email`
Default Value:	*Not Set*
Valid Values:	`InboundQueue, OutboundQueue, QAQueue, AgentWorkbin, AgentGroupWorkbin, PlaceWorkbin, PlaceGroupWorkbin`
Dependencies:	`email/enabled`



Name	Value
CustomerService_IQ	"InboundQueue"
Out_Q	"OutboundQueue"
QA_Q	"QAQueue"
Sales_IQ	"InboundQueue"

media:<name> Section

The present version of the G*plus* Adapter has defined configuration Options for each of the following media:

- voice
- email
- outboundpreview
- chat

This section allows configuration for a generic media supported by the Interaction Server that may be unique to one call center. The <name> portion of the section name must correspond to the Media Type attribute contained in the Interaction Server events.

Besides a straightforward configuration of how the adapter will deal with the new media, this section also has options that define associations with the other configured media types. It should be noted that this section should be used to configure the media that wraps or initiates the associated media interactions.

Note: ThisMedia is used as an identifier in the following discussion for the media that is configured within this section. OtherMedia describes any media that may be associated with ThisMedia in multiple media interactions.

acwReason

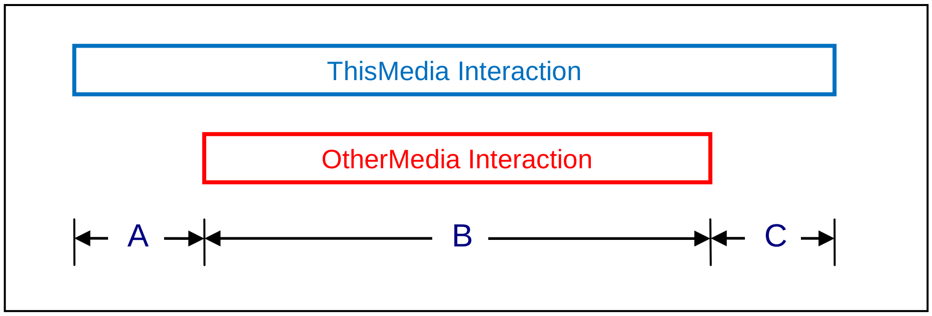
Type:	Optional
Default Value:	ACW
Valid Values:	Any text string
Dependencies:	NotReady Reason is attached by agent desktop

This option identifies the Reason string attached to a NotReady event that will signify the start of a Wrap (AfterInteractionWork) state rather than an Unavailable state.

associationType

Type: Optional
Default Value: none
Valid Values: handle, work, target or none
Dependencies: OtherMedia

The diagram below demonstrates two associated media interactions and the time intervals defined by the association. This diagram will be used to clarify the alternative values that deal with the non-overlapping time (A and C) for the two media.



- none
  - there is no association with the OtherMedia type
  - all of the interaction time for ThisMedia is reported as handle time
  - all of the interaction time for OtherMedia is reported as handle time if that interaction was routed

- handle
  - non-overlapping time (A and C) will be reported as handle time for ThisMedia

- work
  - non-overlapping time (A and C) will be reported as wrap up (AfterInteractionWork) time for ThisMedia

- target
  - the target determined for ThisMedia interaction will be used for the OtherMedia interaction
  - none of the non-overlapping time (A and C) will be reported

Note:	Contact Professional Services prior to configuring this option as it will require a detailed understanding of the association between the two related media.

defaultAbandonedThreshold

Type:	Optional
Default Value:	`0s`
Valid Values:	See Threshold Format
Dependencies:	`historical.reports/serviceLevelFormula`

This option specifies the default abandoned threshold for the service level calculations that form part of the interval reports. It is also quite possible that there is no mechanism to actually abandon a `ThisMedia` interaction and this option can be ignored.

This option should be set when a single abandoned threshold is used as a default for all `ThisMedia` targets. Individual abandoned thresholds can be set for each `ThisMedia` target/queue by adding `Annex Data` to the `ThisMedia Virtual Queues` as shown in a later section or adding the threshold to an `Interaction Queue` as described in the interactionTargets Section. These individual thresholds will override the default specified by this option.

defaultServiceThreshold

Type:	Optional
Default Value:	`0s`
Valid Values:	See Threshold Format
Dependencies:	`markServicedOn`

This option specifies the default service threshold for the service level calculations that form part of the interval reports. A `ThisMedia` is considered *serviced* at either the start or the end of the agent's involvement with the interaction depending on the `markServicedOn` Option.

This option should be set when a single service threshold is used as a default for all `ThisMedia` targets. Individual service thresholds can be set for each `ThisMedia` target/queue by adding `Annex Data` to the `ThisMedia Virtual Queues` as shown in a later section or adding the threshold to an `Interaction Queue` as described in the interactionTargets Section. These individual thresholds will override the default specified by this option.

defaultTarget

Type:	Optional
Default Value:	`UnknownTarget`
Valid Values:	Any text string
Dependencies:	`application/countUnattachedACW`

This option allows users to define a separate target/queue for this media other than the default value used by the G*plus* Adapter to report on unattached wrap up (`AfterInteractionWork`) time. This option is only applicable when the `application/countUnattachedACW` Option is set to `true`.

Unattached `AfterOpenMediaWork` time is defined as the time that the agent spends in a wrap up state when the agent has not yet received a routed interaction of the `ThisMedia` type since logging in.

deleteOnTransfer

Type:	Optional
Default Value:	`false`
Valid Values:	`true, false`
Dependencies:	`OtherMedia`, `associationType`

This option deals with a transfer scenario where the `OtherMedia` interaction is transferred to another agent but there is no "End of Interaction" event received for `ThisMedia`. If this option is set to `true`, the `ThisMedia` interaction will be terminated when the transfer is completed.

Note: Contact Professional Services prior to configuring this option as it will require a detailed understanding of the association between the two related media.

enabled

Type:	Mandatory
Default Value:	`false`
Valid Values:	`true, false`
Dependencies:	None

This option must be set to `true` to capture and report on historical `ThisMedia` activity in the contact center. It is possible to disable the tracking of media activity if required.

interaction.reason.determiner

Type:	Optional
Default Value:	*Not set*

Valid Values:	`eventAttribute:` {Event attribute name}
	`class:` {custom Java class name}

Dependencies:	`enabled, rta/enabled`

This option specifies the attribute of a `ThisMedia` Interaction Server Event that will be included as the Reason field of an RTA state message generated by one of the following events:

- `EventPartyAdded`
- `EventPropertiesChanged` while the agent is a party to a routed interaction of the `ThisMedia` type.

Certain implementations may require that the combination of a `RoutedOpenMediaWork` RTA state and the reason code map to a different RTA state and reason text string. This mapping can be configured in the interactionCodeMappings Section.

Note: Contact Professional Services before using this option.

markServicedOn

Type:	Mandatory
Default Value:	`answered`
Valid Values:	`answered, released`
Dependencies:	None

This option indicates when an interaction of the `ThisMedia` type is considered to be serviced for the Service Level Threshold calculations. Either when the agent first *answers* the interaction or when the agent has completed handling the interaction.

otherIdDeterminer

Type:	Optional
Default Value:	*Not Set*
Valid Values:	Any valid determiner.
Dependencies:	`OtherMedia, associationType, thisIdDeterminer`

This option identifies the determiner used to uniquely identify the `OtherMedia` interaction and provide the association back to the `ThisMedia` interaction that forms the other half of connected interactions.

Note: Contact Professional Services prior to configuring this option as it will require a detailed understanding of the association between the two related media.

otherMediaTypes

Type:	Optional
Default Value:	*Not Set*
Valid Values:	Comma delimited list of media names.
Dependencies:	`OtherMedia`, `associationType`

This option lists the names of the OtherMedia that could be associated with `ThisMedia`.

Note: Contact Professional Services prior to configuring this option as it will require a detailed understanding of the association between the two related media.

reason.notReady.determiner

Type:	Optional
Default Value:	`eventAttribute: ReasonDescription`

Valid Values:	`eventAttribute:` {Event attribute name}
	`class:` {custom Java class name}

Dependencies:	`enabled, rta/enabled`

This option specifies the attribute of a `ThisMedia` Interaction Server Event that will be included as the Reason field of an RTA state message generated by one of the following events:

- `EventNotReadyForMedia`
- `EventMediaStateReasonChanged` while the agent is NotReady to receive a routed interaction of a `ThisMedia` type.

Certain implementations may require that the combination of an `Unavailable` RTA state and the reason code map to a different RTA state and reason text string. This mapping can be configured in the reasonCodeMappings Section

Note: Contact Professional Services before using this option.

reason.ready.determiner

Type:	Optional
Default Value:	*Not set*

Valid Values:	`eventAttribute`: {Event attribute name}
	`class`: {custom Java class name}

Dependencies:	`enabled`, `rta/enabled`

This option specifies the attribute of a `ThisMedia` Interaction Server Event that will be included as the Reason field of an RTA state message generated by one of the following events:

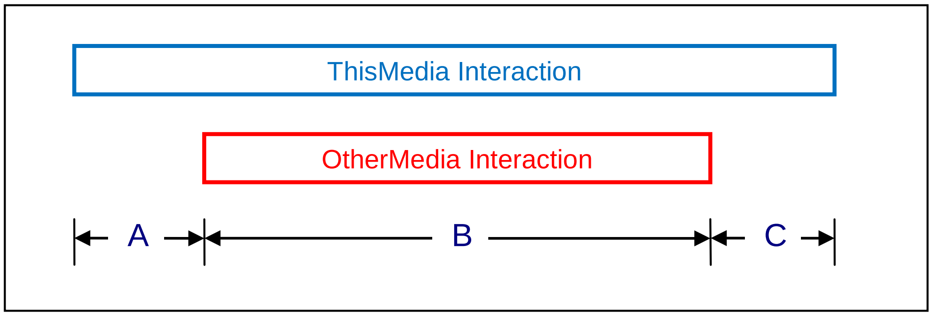
- `EventReadyForMedia`

Certain implementations may require that the combination of an `Available` RTA state and the reason code map to a different RTA state and reason text string. This mapping can be configured in the readyCodeMappings Section.

Note: Contact Professional Services before using this option.

reportingType

Type:	Optional
Default Value:	`thisInteraction`
Valid Values:	`thisInteraction`, `otherInteraction`
Dependencies:	`OtherMedia`, `associationType`



This option determines whether the `ThisMedia` or the `OtherMedia` activity is reported upon for the overlapping interval (B) in the offered and handled metrics.

Note: Contact Professional Services prior to configuring this option as it will require a detailed understanding of the association between the two related media.

rta.routedInteractionState

Type:	Optional
Default Value:	`1700 (interactionInbound)`
Valid Values:	Any 4 digit integer not already defined.
Dependencies:	`rta.nonRoutedInteractionState`, `rta.acwInteractionState`

This option identifies the agent state value that will be passed to the Verint RTA server when the agent is handling a `ThisMedia` interaction that was routed through a monitored queue to that agent.

The defined RTA states are integer codes as shown in the RTA State table.

rta.nonRoutedInteractionState

Type:	Optional
Default Value:	`1701 (interactionInternal)`
Valid Values:	Any 4 digit integer not already defined.
Dependencies:	`rta.routedInteractionState`, `rta.acwInteractionState`

This option identifies the agent state value that will be passed to the Verint RTA server when the agent is handling a `ThisMedia` interaction that was not routed through a monitored queue to that agent.

The defined RTA states are integer codes as shown in the RTA State table.

rta.acwInteractionState

Type:	Optional
Default Value:	`1702 (afterInteractionWork)`
Valid Values:	Any 4 digit integer not already defined.
Dependencies:	`rta.routedInteractionState`, `rta.nonRoutedInteractionState`

This option identifies the agent state value that will be passed to the Verint RTA server when the agent is wrapping up a concluded `ThisMedia` interaction that was routed through a monitored queue to that agent.

The defined RTA states are integer codes as shown in the RTA State table.

targetProperty

Type:	Mandatory
Default Value:	`eventAttribute: interactionProperties.interactionQueue`
Valid Values:	Any valid Interaction target determiner.
Dependencies:	None

This option identifies the event attribute that the adapter will use to identify the target/queue for the `ThisMedia` in the same way as the `event.properties/interactionTarget.determiner` is defined for `email`.

thisIdDeterminer

Type:	Optional
Default Value:	*Not Set*
Valid Values:	Any valid determiner
Dependencies:	`OtherMedia`, `associationType, otherIdDeterminer`

This option identifies the determiner used to uniquely identify the `ThisMedia` interaction and provide the association back to the `OtherMedia` interaction that forms the other half of connected interactions.

Note: Contact Professional Services prior to configuring this option as it will require a detailed understanding of the association between the two related media.

outbound Section

classifyPartyDeterminer

Type:	Optional
Default Value:	`eventAttribute: UserData.GSW_DISPOSITION_CODE`
Valid Values:	Any `Event` attribute
Dependencies:	`classifyPartyMappings` Option Section

This option specifies which event attribute contains the disposition code for the associated Outbound Campaign call. Once the disposition code has been parsed from the call event, it can be compared against the Option keys listed in the `classifyPartyMappings` Option **S**ection to determine if the call was answered by the “right” or “wrong” party.

defaultTarget

Type:	Optional
Default Value:	`UnknownTarget`
Valid Values:	Any text
Dependencies:	`application/countUnattachedACW`

This option allows users to define a separate `outbound` target/queue or campaign name other than the default value used by the G*plus* Adapter for all media. This value is used to report on an agent's time outside of actual `outbound` activity (including unattached ACW) when that time cannot be associated with an actual `outbound` target/queue or campaign name.

dialingDN

Type:	Optional
Default Value:	*Not Set*
Valid Value:	Name of an actual `Trunk Group`
Dependencies:	Predictive or Progressive Campaign Type

This option specifies the name of the `Trunk Group` that is used in either Predictive or Progressive campaigns to dial the `outbound` calls. Only calls dialed from a monitored `DN` will be included in the `Dials` column of the Outbound Statistics report.

dialingRouteGroup

Type:	Optional
Default Value:	*Not Set*
Valid Value:	Name of an actual `Routing Point DN Group`
Dependencies:	Predictive or Progressive Campaign Type

This option specifies the name of the DN Group that contains the `Routing Points` (`RPs`) used in either Predictive or Progressive campaigns to dial the `outbound` calls. Only calls dialed from a monitored `RPs` will be included in the `Dials` column of the Outbound Statistics report.

enabled

Type:	Deprecated
Default Value:	`false`
Valid Values:	`true, false`
Dependencies:	`event.properties/campaignTarget.determiner.`

This option allowed users to enable reporting of outbound campaign calls in the Outbound Statistics report

See Outbound Campaign Calls for more information.

recordProcessedOnRelease

Type:	Optional
Default Value:	`true`
Valid Values:	`true, false`
Dependencies:	Predictive or Progressive Campaign Type OCS `OCServer/record_processed` Option

This option sets when the G*plus* Adapter considers the Outbound Contact Server (OCS) `outbound` call record to be finished. If this option is set to `true`, then the record is marked as processed when the `outbound` call is released. If the option is set to `false`, the record is considered processed when a specific `EventUserEvent` is received after the call has been released.

readyCodeMappings Section

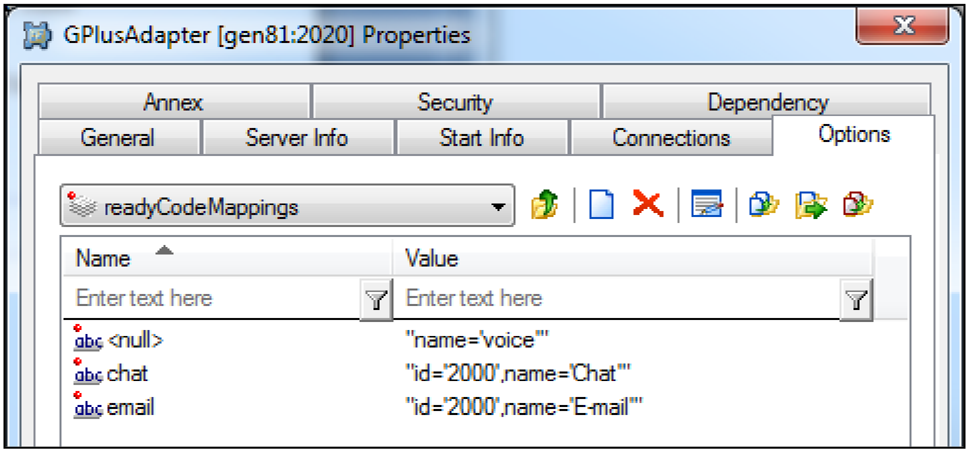
This section is optional and must be added manually as it is not included in the Application Template. When the section is included, the reasons parsed by the `reason.ready.determiner(s)` will be compared against the Option key names in this section. If there is a match, the parameters listed in the Option value will be used. There are four possible parameters:

1.	`id`	new RTA state.
2.	`name`	new reason code.

The `id` field can be either the new RTA State Name or State ID as defined in the RTA State table. The `name` field is the new Reason that will be included with the RTA state message sent to the Verint server.

The `<null>` Option key signifies an event with no reason attached and it is used to configure a default behavior for the *Ready* events.

Note:	Contact Professional Services before configuring this section.



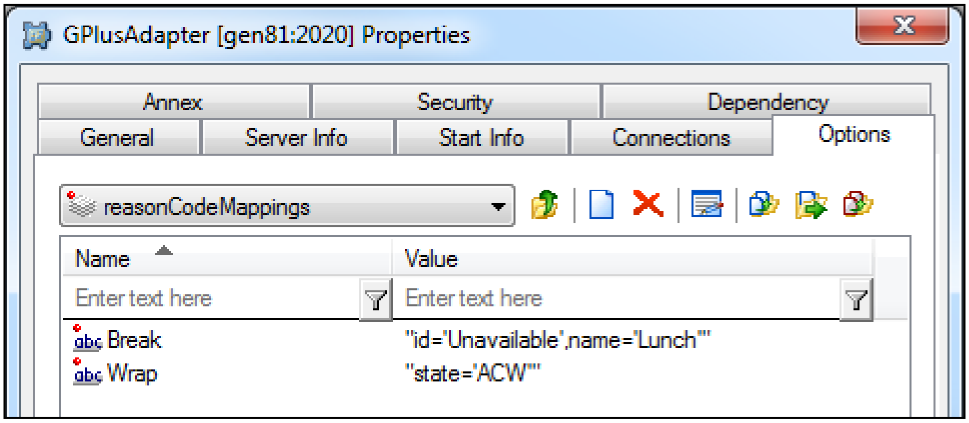
readyCodeMappings

Name	Value
<null>	"name='voice'"
chat	"id='2000',name='Chat'"
email	"id='2000',name='E-mail'"

reasonCodeMappings Section

This section is optional and must be added manually as it is not included in the Application Template. When the section is included, the NotReady reasons parsed by the `reason.notReady.determiner` will be compared against the Option key names in this section. If there is a match, the parameters listed in the Option value will be used. There are three possible parameters:

1. `id`	new RTA state.
2. `name`	new NotReady reason code.
3. `state`	`NotReady` or `ACW`



reasonCodeMappings

Name	Value
Break	"id='Unavailable',name='Lunch'"
Wrap	"state='ACW'"

The `id` field is the new RTA State Name as defined in the RTA State table. The `name` field is the new Reason that will be included with the RTA state. The `state` field allows the event to be interpreted as either `Unavailable` or `ACW` in the historical reports.

The `<null>` Option key signifies an event with no reason attached and it is used to configure a default behavior for the NotReady events.

Note: Contact Professional Services before configuring this section.

rta Section

enabled

Type:	Mandatory
Default Value:	`true`
Valid Values:	`true, false`
Dependencies:	None

The option allows users to disable the RTA data stream for G*plus* Adapter installations where RTA data is not required.

filter

Type:	Optional
Default Value:	*Not set*
Valid Values:	A `filterName` (`filter:<filterName>` Section)
Dependencies:	A filter Section must be defined.

This option allows users to associate the options defined in this Section with a defined `filter` section by setting the value to `filterName`. If multiple streams are not defined, then this option can be left blank and the RTA stream will be associated with the `Agents`, `VQs` and `Places` defined by the `genesys.*Groups` sections.

reportDndAsNotReady

Type:	Optional
Default Value:	`true`
Valid Values:	`true`, `false`
Dependencies:	None

The default behavior is for the Adapter to report Do Not Disturb time as the `NotReady` RTA state. This option allows the Adapter to report that time as the `DndOn` RTA state if set to `false`.

stateTransformer.class

Type:	Mandatory
Default Value:	`com.ariasolutions.iconnect.rta.VerintRTAStateTransformer`
Valid Values:	`com.ariasolutions.iconnect.rta.VerintRTAStateTransformer`
Dependencies:	None

This option specifies the Java class that defines the mapping of Genesys events to state transitions in the RTA data stream. The default is the standard mapping for Verint and should not have to be changed.

verint.listenPort

Type:	Optional
Default Value:	`12000`
Valid Values:	Any open port on the G*plus* Adapter host.
Dependencies:	None

This option specifies the port that the G*plus* Adapter listens on while waiting for a connection request from the Verint RTA application.

rta:<streamName> Section

This Section is optional and is used when the G*plus* Adapter instance is providing more than one data Stream. This Section is used to configure the Real Time Adherence (RTA) component of the data feed to the Verint Server. The Options are listed below but are not discussed here as they directly correspond to those listed in the rta section.

Options

- enabled
- filter
- reportDndAsNotReady
- stateTransformer.class
- verint.listenPort

voice Section

defaultAbandonedThreshold

Type:	Optional
Default Value:	`0s`
Valid Values:	See Threshold Format
Dependencies:	`historical.reports/serviceLevelFormula`

This option specifies the default abandoned threshold for the service level calculations that form part of the interval reports. Only two of the service level formulae actually use the abandoned threshold so this option has no effect unless one of those formulae (1 or 4) has been selected. An inbound `voice` call is considered *abandoned* if the customer releases the call prior to an agent answering it.

This option should be set when a single abandoned threshold is used as a default for all `voice` targets. Individual abandoned thresholds can be set for each `voice` target/queue by adding `Annex Data` to the `voice` `Virtual Queues` as shown in a later section. These individual thresholds will override the default specified with this option.

defaultServiceThreshold

Type:	Optional
Default Value:	`0s`
Valid Values:	See Threshold Format
Dependencies:	None

This option specifies the default service threshold for the service level calculations that form part of the interval reports. An inbound `voice` call is considered *serviced* at the time that the agent answers the call.

This option should be set when a single service threshold is used as a default for all inbound `voice` call targets. Individual service thresholds can be set for each inbound `voice` call target/queue by adding `Annex Data` to the `voice` `Virtual Queues` as shown in a later section. These individual thresholds will override the default specified by this option.

defaultTarget

Type:	Optional
Default Value:	`UnknownTarget`
Valid Values:	Any text string
Dependencies:	`application/countUnattachedACW`

This option allows users to define a separate contact type/queue other than the default value used by the G*plus* Adapter to report on unattached After Call Work time. This option is applicable if the `application/countUnattachedACW` Option in the `application` Section is set to `true`.

Unattached After Call Work time is defined as the time that the agent spends in a NotReady state initiated by an `EventAgentNotReady` containing a `WorkMode` attribute of `3` during the portion of a login session when the agent has not yet answered a routed call.

enabled

Type:	Mandatory
Default Value:	`false`
Valid Values:	`true, false`
Dependencies:	None

This option is set to `true` to capture and report on inbound `voice` call activity in the contact center. It is possible to disable the tracking of `voice` activity if the G*plus* Adapter was installed to only track the agents other media interactions.

interaction.reason.determiner

Type:	Optional
Default Value:	*Not set*

Valid Values:	`eventAttribute:` {Event attribute name}
	`class:` {custom Java class name}

Dependencies:	`voice/enabled, rta/enabled`

This option specifies the attribute of a TServer/SIPServer `voice` Event that will be included as the Reason field of an RTA state message generated by one of the following events:

- `EventEstablished`
- `EventHeld`
- `EventPartyAdded`
- `EventPartyChanged`
- `EventPartyDeleted`
- `EventRetrieved`

Certain implementations may require that the combination of a `voice` call RTA state and its reason code map to a different RTA state and reason text string. This mapping can be configured in the interactionCodeMappings Section.

`Note:` Contact Professional Services before using this option.

isAfterCallWork.class

Type:	Optional
Default Value:	*Not Set*
Valid Values:	Check with Technical support.
Dependencies:	None

The option overrides the default behavior for determining whether an agent has entered an ACW state. One, `PresumptiveAfterCallWorkProperty,` can be used to report all NotReady time as After Call Work.

parallelQueuesEnabled

Type:	Optional
Default Value:	`false`
Valid Values:	`true, false`
Dependencies:	Routing strategy

This option is set to `true` to report a call against a different queue than the first monitored queue in one particular circumstance. Genesys routing can place a call in additional queues to broaden the target agent group if the call has been waiting in the original queue for a set period. This option, if set to `true`, allows the target for the customer call to be associated with the queue that the call was diverted from rather than defaulting to the original queue.

Note: Contact Professional Services prior to changing this option.

reason.notReady.determiner

Type:	Optional
Default Value:	`eventAttribute: Reasons.ReasonCode`

Valid Values:	`eventAttribute:` {Event attribute name}
	`class:` {custom Java class name}

Dependencies:	`voice/enabled, rta/enabled`

This option specifies the attribute of a `voice` TServer/SIPServer event that will be included as the Reason field of an RTA state message generated by one of the following events:

- `EventAgentNotReady`

Certain implementations may require that the combination of an `Unavailable` RTA state and the reason code map to a different RTA state and reason text string. This mapping can be configured in the reasonCodeMappings Section

Note: Contact Professional Services before using this option.

reason.ready.determiner

Type:	Optional
Default Value:	*Not set*

Valid Values:	`eventAttribute:` {Event attribute name}
	`class:` {custom Java class name}

Dependencies:	`enabled, rta/enabled`

This option specifies the attribute of a `ThisMedia` Interaction Server Event that will be included as the Reason field of an RTA state message generated by one of the following events:

- `EventAgentReady`

Certain implementations may require that the combination of an `Available` RTA state and the reason code map to a different RTA state and reason text string. This mapping can be configured in the readyCodeMappings Section.

Note: Contact Professional Services before using this option.

Chapter

4 Additional Configuration

Adding T-Servers and Interaction Servers

The application establishes connections with the TServers, SIP Servers, Interaction Servers and Message Servers listed under the `Connections` tab of the `Properties` window. Adding a server is accomplished in exactly the same way that any other Genesys application would add connections to servers in the Genesys environment.

Select the `Add` button and then the `Browse` button at the end of the `Server` line on the `Connection Info Properties` window. The adapter supports connections to multiple TServers/SIP Servers/Interaction Servers and using the `addp` connection protocol for those connections. Only the Primary servers should be added to the `Connections` tab, as the Backup servers will be identified from the Primary's configuration and included automatically.

Transport Layer Security (TLS) can also be added to a server connection by checking the `Secure` checkbox. This assumes that the *Security Deployment Guide* has been followed to properly install certificates etc.

Threshold Time Format

The service level calculations are based upon calls/interactions that meet or do not meet configured service or abandoned time thresholds. These thresholds can be set as a default per media or added to the following configuration objects: `Skills`, `Virtual Queues` and `interactionTargets`. The default for setting the threshold time is to specify it in milliseconds. However, a more user friendly version is also available where the threshold time is specified in terms of days, hours, minutes and seconds. A typical call center would have a service (answered) threshold for inbound calls that would be in the range of 20 to 30 seconds whereas a threshold for emails may be measured in hours or days. The following format allows a range of time values to be specified without having to calculate the millisecond equivalent:

```
{0-31}d{0-24}h{0-60}m{0-60}s
```

Examples:

15000	15000 milliseconds
60s	60000 milliseconds
45m	2700000 milliseconds
1d12h30m	131400000 milliseconds

Note: Each field size is restricted to two characters so none of the numbers can be larger than 99. It would be advisable to use the ranges specified above.

Adding Annex Data for Virtual Queues and Skills

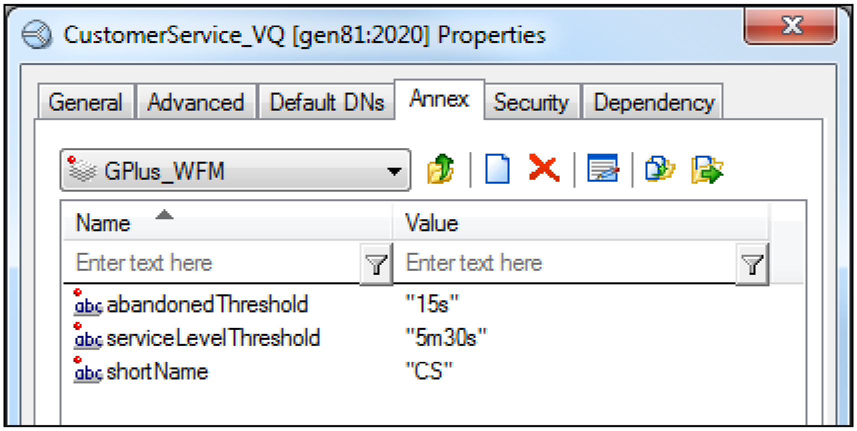
This section discusses adding individual configuration to a `Skill` or `Virtual Queue` with the end result being additional configuration of the associated call/interaction target. Three parameters can be added to these Configuration Objects to modify the reported results:

- `abandonedThreshold`
- `serviceLevelThreshold`
- `shortName`

The Contact Statistics reports deal with statistics for each Contact Group and the two thresholds can differ between Groups as the business expectations for each type of customer call will vary. Associating the thresholds with a specific Skill or Virtual Queue allows the reports to take into account the different response expectations.

Two of the service level formulae supported by this adapter (see serviceLevelFormula Option in historical.reports Section, Formula 1 and Formula 4) have components that take into account whether the call was abandoned before reaching a threshold. This abandoned threshold can also be set for each Skill or Virtual Queue but it only affects the service level calculation and not the abandoned count in the reports. Configuring abandoned thresholds for an email Virtual Queue while allowed would have little effect as emails are not abandoned.

The figure below shows an example with the opened Annex tab for a Virtual Queue object in the Configuration Manager. Adding the threshold settings to a Skill object is essentially the same process. The question of which of the two object types will be configured with the thresholds is answered by the application/callType Option.



Name	Value
Enter text here	Enter text here
abandonedThreshold	"15s"
serviceLevelThreshold	"5m30s"
shortName	"CS"

The following steps should be followed to add a serviceLevelThreshold or abandonedThreshold to a Virtual Queue.

1. Select the Virtual Queue in the Configuration Manager and open the Properties frame.
2. Select the Annex tab.
3. Add a GPlus_WFM Section or if the application/annexSection Option has been set, add a Section with the name configured there.
4. In the new Section, add an Option with the Name set to serviceLevelThreshold or abandonedThreshold with value equal to the new threshold as described above.

If the Skill or Virtual Queue (Number or Alias fields) name exceeds the report field size or the WFM application requires something different than the actual name, the shortName option can be added to the Annex tab of the Virtual Queue as follows:

1. Create a GPlus_WFM section in the Annex tab if it does not exist already.
2. Add a new option to this section named shortName with the Value set to the new text string.
3. This value will now be used in place of the original Virtual Queue/Skill name in all reports.

Adding A PlaceGroup to a Site

All of the Contact Statistics reports can include Site information if required. The G*plus* Adapter creates an association between Place Groups and Sites by adding a siteName option to a Place Group's Annex Data. Only the Place Groups included in the genesys.placeGroups Section of the Options will be included and, therefore, only those Place Groups should be modified.

To add a Site name for use in reporting:

1. Add a GPlus_WFM Section or if the application/annexSection Option has been set, add a Section with the name configured there.
2. Add a new option to this section named siteName with the new Site name as the value.
3. This name will be used as the Site identifier in the reports.

Chapter

5 Setting Up Secure File Transfer

The file transfer protocol that forwards the historical reports to the remote WFM server can be configured to use SSH (Secure Shell) as the underlying network channel. Three options are included in the historical.ftp section of the application configuration to set up the G*plus* Adapter as an SSH client. The configuration of the client side of the SSH channel is reasonably simple as most of the complexity is reserved for the configuration of the server side.

A description of some of the possible errors due to a misconfigured client or server has been included as Appendix B.

Generating the Private-Public Key Pair

The authentication of the SSH login to the remote server can be carried out using a public-private key pair. The private key is stored on the G*plus* Adapter server and the public key is stored on the remote Verint server. When the keys are generated, it is also possible to specify a key passphrase to further limit access to the private key.

The embedded SSH client supports SSH2 RSA keys. The SSH2 protocol does not specify a standard key format and there are key formats that are not compatible with the SSH client used in the Adapter. Keys generated by the PuTTY `gen` key generation tool are not compatible. Keys generated by the OpenSSH `ssh-keygen` tool are compatible. Tools using the following ciphers should be compatible:

- `aes256-ctr`
- `aes192-ctr`
- `aes128-ctr`
- `aes256-cbc`

- `aes192-cbc`
- `aes128-cbc`
- `3des-ctr`
- `arcfour`
- `arcfour128`
- `arcfour256`

As mentioned, the private key is stored with the G*plus* Adapter where the embedded SSH client can access it. This key is usually stored in the `config` folder of the Adapter installation.

Storing the public key will be specific to the SSH server that has been used on the Verint server. The public key will have to be placed in a certain folder and the key will have to be registered with the SSH server. Both of these actions are implementation specific and should be fully described in the SSH server documentation.

Client Configuration

The following options in the `historical.ftp` section must be configured or at least considered:

`enabled`	`true` or the files are not transferred
`remoteHost`	host name of the destination server
`remotePort`	host port on the destination server
`secureTransfer`	`true` to use SSH as a secure connection
`userName`	user registered on the destination server
`userPassword`	password for the user identified by `userName`
`keyPath`	path and name of the private key file
`keyPassphrase`	pass phrase that allows access to the private key

Server Configuration

The SSH server should support configuration for the authentication modes that it allows and which mode it will attempt first. The two supported options for the Adapter are public key or password. Password authentication is not as secure because the `userPassword` option is retained as plain text in the Configuration Manager. The public key authentication is more secure and more difficult to implement.

Chapter

6 Outbound Voice Campaigns

The G*plus* Adapter now supports the Outbound Campaign types that did not fit the original inbound voice call model used to define a customer call. This chapter will discuss the Outbound Campaign types, the Outbound Statistics report and how the Adapter classifies the calls associated with the campaigns.

There are four types of Genesys outbound campaigns that can be monitored by the G*plus* Adapter:

1. Predictive
2. Progressive
3. Preview
4. Push Preview

The first two, Predictive and Progressive, are routed to the agent after being dialed by the dialer while the second two, Preview and Push Preview are essentially dialed from the agent's phone. All four types have a campaign name included in the event attributes that the `event.properties/campaignTarget.determiner` can parse.

Predictive and Progressive Campaigns

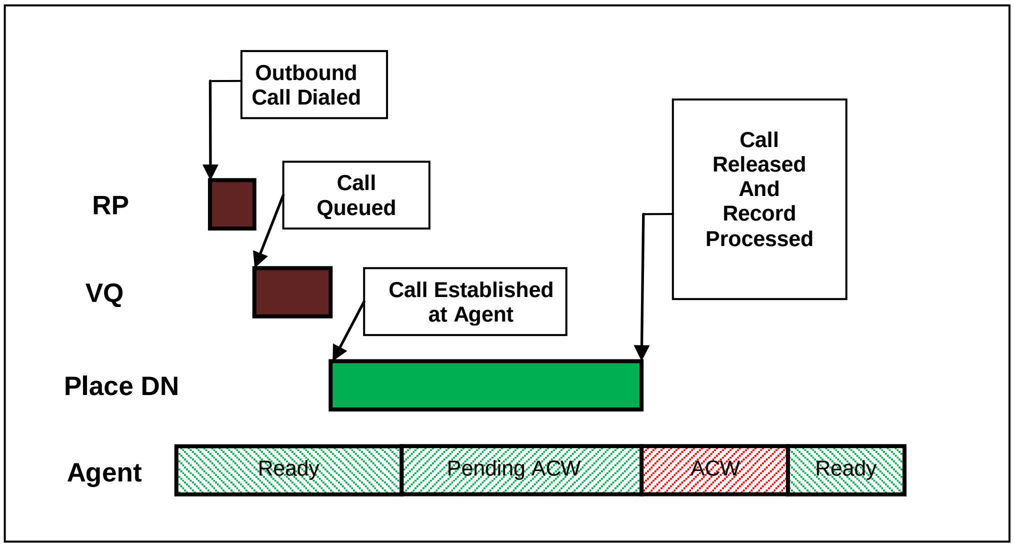
Both the Predictive and Progressive campaign types have call flows that start as a dialing event on a Routing Point, queued on a Virtual Queue, diverted to and then answered by an available agent with an optional ACW interval after the call is released. The adapter does not differentiate between the two Campaign types.

The Adapter attempts to determine the campaign name at two points in the call flow: first by parsing the dialing event associated with the monitored Routing Point and if that fails, parsing the queued event on the monitored Virtual Queue. It is recommended that the dialing Routing Points be monitored and the Campaign Name is one of the attributes attached to the dialing event. Even if the Campaign Name is parsed in the dialing event, it is still important that the Virtual Queue associated with the targeted Agent Group also be monitored. The time that the call waits in queue for an available agent is still required for the Outbound Statistics report.

The Figures below illustrate the reason for including the recordProcessedOnReleased Option depending upon the events generated at the conclusion of the call. This Option specifies whether the Outbound Contact Server (OCS) record associated with the call is closed at the conclusion of the call or at a subsequent point marked by a record processed UserEvent.

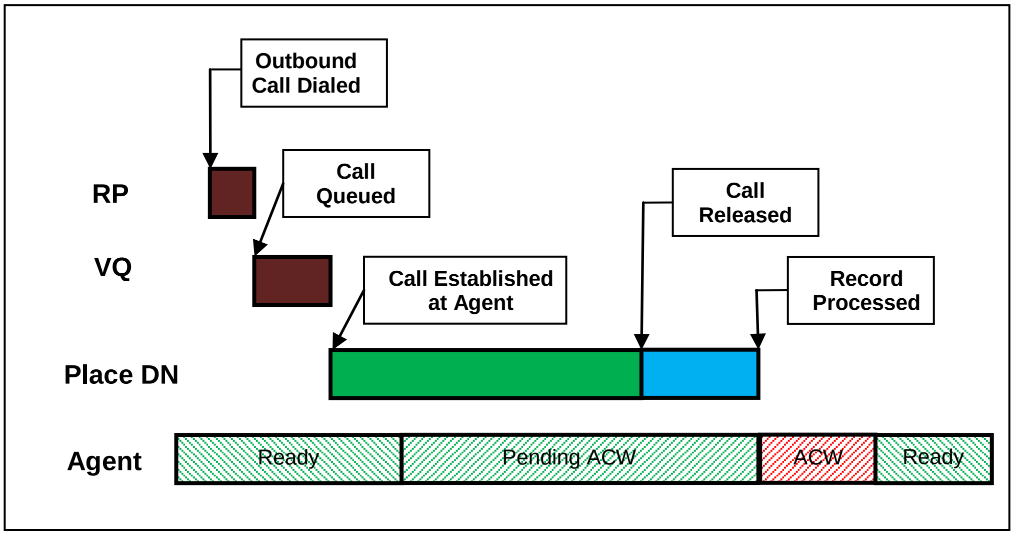
recordProcessedOnReleased = true

The first Figure shows the record being processed at the conclusion of the call. There is no wrap time associated with an open OCS record and any disposition codes must be attached to the call prior to the call being released.



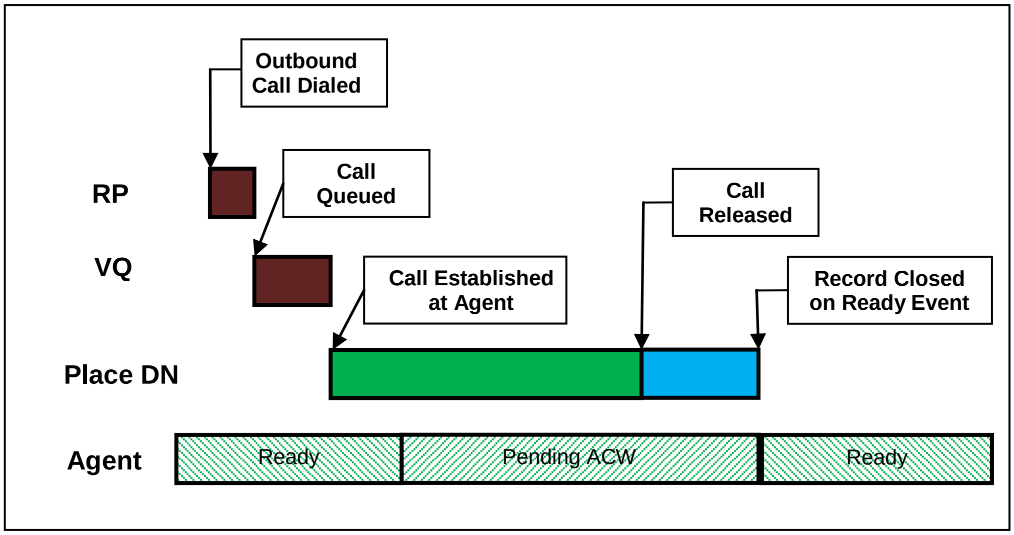
recordProcessedOnReleased = false

The second Figure shows the record being processed after the conclusion of the call. There is wrap time associated with an open OCS record and also ACW time associated with the call that is counted after the record is processed. Any disposition codes must be added to the call prior to the record being processed.



recordProcessedOnReleased = false – Record Closed Prematurely

The third Figure shows the record being closed on an error generated by the OCS when the agent goes into a Ready state rather than marking the OCS record as done. Any errors caused by state changes or problems with the OCS will result in the Adapter considering the record closed. There is wrap time associated with an open OCS record up to the agent state change but any disposition codes attached to the call after the state change are discarded by OCS.



Preview and Push Preview Campaigns

The Preview and Push Preview campaigns are tracked properly by the Adapter if the `campaignTarget.determiner` and the `classifyPartyDeterminer` are configured to identify the `Campaign Name` and disposition code respectively.

Non Campaign Outbound Calls

If the `campaignTarget.determiner` fails to identify a `Campaign Name` on a Progressive or Predictive call but the `callType.determiner` parses a target name from a queued event on a monitored `VQ`, the outbound call will be reported in the Contact Statistics report. Calls that do not have an associated target will still be regarded as outbound calls but they will be counted in the Agent Scorecard Metric report as non-customer outbound activity.

Chapter

7 Streams and Filters

Multiple Streams

Prior versions of the adapter were limited to a single Stream where there was one set of historical reports generated and one RTA connection supported. Multiple Streams allow subsets of the accumulated data to be sent to different Verint servers through the use of Filters that identify the data subsets and Streams that identify the destination for that data.

Filters are configured as separate Option Sections in the G*plus* Adapter Configuration as described in the filter:<filtername> Section. A Filter defines a set of monitored `Agents`, `Places` and `Virtual Queues` (`VQs`) that any associated Stream will report upon. Only `Agents`, `Places` and `VQs` that are part of the groups configured in the `genesys.*Groups` Sections are candidates to be included in a filter. It is not possible to include `Agent`, `Place` or `VQ` groups that are not included in the `genesys.*Groups` Sections.

Streams are associated with Filters with a singular reference to the Filter name but multiple Streams can be associated with a single Filter. The obvious configuration would be a historical and a rta stream "connected" to the same Filter. Another configuration might see two historical streams: one `siteBased` and the other not.

Each Stream can be configured to generate a single report or a subset of the five available reports.

Appendix

A Time Zone ID List

The list of acceptable values to specify the G*plus* Adapter `timeZone` Option mirrors the list of acceptable Time Zone Ids for the Java JVM. For reference the supported option values are listed in the table below. The recommended value is to use one of the `Region/Locale` formats if possible as they deal with Daylight Savings Time transparently.

ACT	AET	AGT
ART	AST	Africa/Abidjan
Africa/Accra	Africa/Addis_Ababa	Africa/Algiers
Africa/Asmara	Africa/Asmera	Africa/Bamako
Africa/Bangui	Africa/Banjul	Africa/Bissau
Africa/Blantyre	Africa/Brazzaville	Africa/Bujumbura
Africa/Cairo	Africa/Casablanca	Africa/Ceuta
Africa/Conakry	Africa/Dakar	Africa/Dar_es_Salaam
Africa/Djibouti	Africa/Douala	Africa/El_Aaiun
Africa/Freetown	Africa/Gaborone	Africa/Harare
Africa/Johannesburg	Africa/Kampala	Africa/Khartoum
Africa/Kigali	Africa/Kinshasa	Africa/Lagos
Africa/Libreville	Africa/Lome	Africa/Luanda
Africa/Lubumbashi	Africa/Lusaka	Africa/Malabo
Africa/Maputo	Africa/Maseru	Africa/Mbabane
Africa/Mogadishu	Africa/Monrovia	Africa/Nairobi

Africa/Ndjamena	Africa/Niamey	Africa/Nouakchott
Africa/Ouagadougou	Africa/Porto-Novo	Africa/Sao_Tome
Africa/Timbuktu	Africa/Tripoli	Africa/Tunis
Africa/Windhoek	America/Adak	America/Anchorage
America/Anguilla	America/Antigua	America/Araguaina
America/Argentina/Buenos_Aires	America/Argentina/Catamarca	America/Argentina/ComodRivadavia
America/Argentina/Cordoba	America/Argentina/Jujuy	America/Argentina/La_Rioja
America/Argentina/Mendoza	America/Argentina/Rio_Gallegos	America/Argentina/San_Juan
America/Argentina/Tucuman	America/Argentina/Ushuaia	America/Aruba
America/Asuncion	America/Atikokan	America/Atka
America/Bahia	America/Barbados	America/Belem
America/Belize	America/Blanc-Sablon	America/Boa_Vista
America/Bogota	America/Boise	America/Buenos_Aires
America/Cambridge_Bay	America/Campo_Grande	America/Cancun
America/Caracas	America/Catamarca	America/Cayenne
America/Cayman	America/Chicago	America/Chihuahua
America/Coral_Harbour	America/Cordoba	America/Costa_Rica
America/Cuiaba	America/Curacao	America/Danmarkshavn
America/Dawson	America/Dawson_Creek	America/Denver
America/Detroit	America/Dominica	America/Edmonton
America/Eirunepe	America/El_Salvador	America/Ensenada
America/Fort_Wayne	America/Fortaleza	America/Glace_Bay
America/Godthab	America/Goose_Bay	America/Grand_Turk
America/Grenada	America/Guadeloupe	America/Guatemala
America/Guayaquil	America/Guyana	America/Halifax
America/Havana	America/Hermosillo	America/Indiana/Indianapolis
America/Indiana/Knox	America/Indiana/Marengo	America/Indiana/Petersburg
America/Indiana/Vevay	America/Indiana/Vincennes	America/Indianapolis
America/Inuvik	America/Iqaluit	America/Jamaica
America/Jujuy	America/Juneau	America/Kentucky/Louisville
America/Kentucky/Monticello	America/Knox_IN	America/La_Paz
America/Lima	America/Los_Angeles	America/Louisville
America/Maceio	America/Managua	America/Manaus
America/Martinique	America/Mazatlan	America/Mendoza

America/Menominee	America/Merida	America/Mexico_City
America/Miquelon	America/Moncton	America/Monterrey
America/Montevideo	America/Montreal	America/Montserrat
America/Nassau	America/New_York	America/Nipigon
America/Nome	America/Noronha	America/North_Dakota/Center
America/North_Dakota/New_Salem	America/Panama	America/Pangnirtung
America/Paramaribo	America/Phoenix	America/Port-au-Prince
America/Port_of_Spain	America/Porto_Acre	America/Porto_Velho
America/Puerto_Rico	America/Rainy_River	America/Rankin_Inlet
America/Recife	America/Regina	America/Rio_Branco
America/Rosario	America/Santiago	America/Santo_Domingo
America/Sao_Paulo	America/Scoresbysund	America/Shiprock
America/St_Johns	America/St_Kitts	America/St_Lucia
America/St_Thomas	America/St_Vincent	America/Swift_Current
America/Tegucigalpa	America/Thule	America/Thunder_Bay
America/Tijuana	America/Toronto	America/Tortola
America/Vancouver	America/Virgin	America/Whitehorse
America/Winnipeg	America/Yakutat	America/Yellowknife
Antarctica/Casey	Antarctica/Davis	Antarctica/DumontDUrville
Antarctica/Mawson	Antarctica/McMurdo	Antarctica/Palmer
Antarctica/Rothera	Antarctica/South_Pole	Antarctica/Syowa
Antarctica/Vostok	Arctic/Longyearbyen	Asia/Aden
Asia/Almaty	Asia/Amman	Asia/Anadyr
Asia/Aqtau	Asia/Aqtobe	Asia/Ashgabat
Asia/Ashkhabad	Asia/Baghdad	Asia/Bahrain
Asia/Baku	Asia/Bangkok	Asia/Beirut
Asia/Bishkek	Asia/Brunei	Asia/Calcutta
Asia/Choibalsan	Asia/Chongqing	Asia/Chungking
Asia/Colombo	Asia/Dacca	Asia/Damascus
Asia/Dhaka	Asia/Dili	Asia/Dubai
Asia/Dushanbe	Asia/Gaza	Asia/Harbin
Asia/Hong_Kong	Asia/Hovd	Asia/Irkutsk
Asia/Istanbul	Asia/Jakarta	Asia/Jayapura
Asia/Jerusalem	Asia/Kabul	Asia/Kamchatka

Asia/Karachi	Asia/Kashgar	Asia/Katmandu
Asia/Krasnoyarsk	Asia/Kuala_Lumpur	Asia/Kuching
Asia/Kuwait	Asia/Macao	Asia/Macau
Asia/Magadan	Asia/Makassar	Asia/Manila
Asia/Muscat	Asia/Nicosia	Asia/Novosibirsk
Asia/Omsk	Asia/Oral	Asia/Phnom_Penh
Asia/Pontianak	Asia/Pyongyang	Asia/Qatar
Asia/Qyzylorda	Asia/Rangoon	Asia/Riyadh
Asia/Riyadh87	Asia/Riyadh88	Asia/Riyadh89
Asia/Saigon	Asia/Sakhalin	Asia/Samarkand
Asia/Seoul	Asia/Shanghai	Asia/Singapore
Asia/Taipei	Asia/Tashkent	Asia/Tbilisi
Asia/Tehran	Asia/Tel_Aviv	Asia/Thimbu
Asia/Thimphu	Asia/Tokyo	Asia/Ujung_Pandang
Asia/Ulaanbaatar	Asia/Ulan_Bator	Asia/Urumqi
Asia/Vientiane	Asia/Vladivostok	Asia/Yakutsk
Asia/Yekaterinburg	Asia/Yerevan	Atlantic/Azores
Atlantic/Bermuda	Atlantic/Canary	Atlantic/Cape_Verde
Atlantic/Faeroe	Atlantic/Faroe	Atlantic/Jan_Mayen
Atlantic/Madeira	Atlantic/Reykjavik	Atlantic/South_Georgia
Atlantic/St_Helena	Atlantic/Stanley	Australia/ACT
Australia/Adelaide	Australia/Brisbane	Australia/Broken_Hill
Australia/Canberra	Australia/Currie	Australia/Darwin
Australia/Eucla	Australia/Hobart	Australia/LHI
Australia/Lindeman	Australia/Lord_Howe	Australia/Melbourne
Australia/NSW	Australia/North	Australia/Perth
Australia/Queensland	Australia/South	Australia/Sydney
Australia/Tasmania	Australia/Victoria	Australia/West
Australia/Yancowinna	BET	BST
Brazil/Acre	Brazil/DeNoronha	Brazil/East
Brazil/West	CAT	CET
CNT	CST	CST6CDT
CTT	Canada/Atlantic	Canada/Central
Canada/East-Saskatchewan	Canada/Eastern	Canada/Mountain

Canada/Newfoundland	Canada/Pacific	Canada/Saskatchewan
Canada/Yukon	Chile/Continental	Chile/EasterIsland
Cuba	EAT	ECT
EET	EST	EST5EDT
Egypt	Eire	Etc/GMT
Etc/GMT+0	Etc/GMT+1	Etc/GMT+10
Etc/GMT+11	Etc/GMT+12	Etc/GMT+2
Etc/GMT+3	Etc/GMT+4	Etc/GMT+5
Etc/GMT+6	Etc/GMT+7	Etc/GMT+8
Etc/GMT+9	Etc/GMT-0	Etc/GMT-1
Etc/GMT-10	Etc/GMT-11	Etc/GMT-12
Etc/GMT-13	Etc/GMT-14	Etc/GMT-2
Etc/GMT-3	Etc/GMT-4	Etc/GMT-5
Etc/GMT-6	Etc/GMT-7	Etc/GMT-8
Etc/GMT-9	Etc/GMT0	Etc/Greenwich
Etc/UCT	Etc/UTC	Etc/Universal
Etc/Zulu	Europe/Amsterdam	Europe/Andorra
Europe/Athens	Europe/Belfast	Europe/Belgrade
Europe/Berlin	Europe/Bratislava	Europe/Brussels
Europe/Bucharest	Europe/Budapest	Europe/Chisinau
Europe/Copenhagen	Europe/Dublin	Europe/Gibraltar
Europe/Guernsey	Europe/Helsinki	Europe/Isle_of_Man
Europe/Istanbul	Europe/Jersey	Europe/Kaliningrad
Europe/Kiev	Europe/Lisbon	Europe/Ljubljana
Europe/London	Europe/Luxembourg	Europe/Madrid
Europe/Malta	Europe/Mariehamn	Europe/Minsk
Europe/Monaco	Europe/Moscow	Europe/Nicosia
Europe/Oslo	Europe/Paris	Europe/Podgorica
Europe/Prague	Europe/Riga	Europe/Rome
Europe/Samara	Europe/San_Marino	Europe/Sarajevo
Europe/Simferopol	Europe/Skopje	Europe/Sofia
Europe/Stockholm	Europe/Tallinn	Europe/Tirane
Europe/Tiraspol	Europe/Uzhgorod	Europe/Vaduz
Europe/Vatican	Europe/Vienna	Europe/Vilnius

Europe/Volgograd	Europe/Warsaw	Europe/Zagreb
Europe/Zaporozhye	Europe/Zurich	GB
GB-Eire	GMT	GMT0
Greenwich	HST	Hongkong
IET	IST	Iceland
Indian/Antananarivo	Indian/Chagos	Indian/Christmas
Indian/Cocos	Indian/Comoro	Indian/Kerguelen
Indian/Mahe	Indian/Maldives	Indian/Mauritius
Indian/Mayotte	Indian/Reunion	Iran
Israel	JST	Jamaica
Japan	Kwajalein	Libya
MET	MIT	MST
MST7MDT	Mexico/BajaNorte	Mexico/BajaSur
Mexico/General	Mideast/Riyadh87	Mideast/Riyadh88
Mideast/Riyadh89	NET	NST
NZ	NZ-CHAT	Navajo
PLT	PNT	PRC
PRT	PST	PST8PDT
Pacific/Apia	Pacific/Auckland	Pacific/Chatham
Pacific/Easter	Pacific/Efate	Pacific/Enderbury
Pacific/Fakaofo	Pacific/Fiji	Pacific/Funafuti
Pacific/Galapagos	Pacific/Gambier	Pacific/Guadalcanal
Pacific/Guam	Pacific/Honolulu	Pacific/Johnston
Pacific/Kiritimati	Pacific/Kosrae	Pacific/Kwajalein
Pacific/Majuro	Pacific/Marquesas	Pacific/Midway
Pacific/Nauru	Pacific/Niue	Pacific/Norfolk
Pacific/Noumea	Pacific/Pago_Pago	Pacific/Palau
Pacific/Pitcairn	Pacific/Ponape	Pacific/Port_Moresby
Pacific/Rarotonga	Pacific/Saipan	Pacific/Samoa
Pacific/Tahiti	Pacific/Tarawa	Pacific/Tongatapu
Pacific/Truk	Pacific/Wake	Pacific/Wallis
Pacific/Yap	Poland	Portugal
ROK	SST	Singapore
Turkey	UCT	US/Alaska

US/Aleutian	US/Arizona	US/Central
US/East-Indiana	US/Eastern	US/Hawaii
US/Indiana-Starke	US/Michigan	US/Mountain
US/Pacific	US/Pacific-New	US/Samoa
UTC	Universal	VST
W-SU	WET	Zulu

Appendix

B Secure File Transfer Problems

The following excerpt is an example of the logging generated by a successful file transfer using the Secure File Transfer Protocol. Each line shown was actually prefaced with something similar to the following:

```
01 Jan 2011 12:00:00,000 0000 [Connect thread gen75 session] DEBUG SftpTransferAgent.class  -
```

Connecting to remote server:

```
Connecting to gen75 port 22
Connection established
Remote version string: SSH-2.0-OpenSSH_3.8.1p1
Local version string: SSH-2.0-JSCH-0.1.42
```

Checking ciphers:

```
CheckCiphers:       aes256-ctr,aes192-ctr,aes128-ctr,aes256-cbc,aes192-cbc,aes128-cbc,
                    3des-ctr,arcfour,arcfour128,arcfour256
aes256-ctr is not available.
aes192-ctr is not available.
aes256-cbc is not available.
aes192-cbc is not available.
arcfour256 is not available.
```

Initializing:

```
SSH_MSG_KEXINIT sent
SSH_MSG_KEXINIT received
kex: server->client aes128-ctr hmac-md5 none
kex: client->server aes128-ctr hmac-md5 none
SSH_MSG_KEXDH_INIT sent
expecting SSH_MSG_KEXDH_REPLY
ssh_rsa_verify: signature true
Permanently added 'gen75' (RSA) to the list of known hosts.
SSH_MSG_NEWKEYS sent
SSH_MSG_NEWKEYS received
```

```
SSH_MSG_SERVICE_REQUEST sent
SSH_MSG_SERVICE_ACCEPT received
```

Iterating through available authentication methods:

```
Authentications that can continue: publickey,keyboard-interactive,password
Next authentication method: publickey
```

Successful on *publickey*:

```
Authentication succeeded (publickey).
```

File transfer:

```
Attempting to transfer file: 'C:\testFiles\textFile.txt' to 'sftpDestination'
Successfully transferred file: 'C:\testFiles\textFile.txt'
```

Closing connection to remote server:

```
Disconnecting from gen75 port 22
Caught an exception, leaving main loop due to socket closed
```

Incomplete host information

The following are required for the destination server:

- Host name.
- Port
- Destination path relative to the user default.

Incorrect Host Name

An incorrect host name will see the transfer failing as the client attempts to connect to the remote server.

Connecting to remote server:

```
Connecting to something port 22
Error while connecting to SSH server (host: something user: Administrator port: 22)
com.jcraft.jsch.JSchException: java.net.UnknownHostException: something
```

Incorrect Port

An incorrect port will see the transfer failing as the client attempts to connect to the remote server.

Connecting to remote server:

```
Error while connecting to SSH server (host: 127.0.0.1 user: Administrator port: 1234)
com.jcraft.jsch.JSchException:  java.net.ConnectException:  Connection refused:
```

Missing Port

If the port has not been specified in the G*plus* Adapter options, the transfer will fail prior to attempting to connect to the remote server.

```
Incomplete host information. Host, username and destination port must be defined and valid in the CME
```

Public Key Authentication

Public Key Authentication Not Enabled

Public key authentication was not enabled on the destination SSH server.

Iterating through available authentication methods:

```
Authentications that can continue: keyboard-interactive, password
Next authentication method: keyboard-interactive
Authentications that can continue: password
Next authentication method: password
Disconnecting from gen75 port 22
Error while connecting to SSH server (host: gen75 user: Administrator port: 22)
com.jcraft.jsch.JSchException: Auth fail
```

Public Key Authentication Failed

1. The generated private key did not use one of the following supported ciphers:
   - aes256-ctr
   - aes192-ctr
   - aes128-ctr
   - aes256-cbc
   - aes192-cbc
   - aes128-cbc
   - 3des-ctr
   - arcfour
   - arcfour128
   - arcfour256

2. The public key associated with the private key has not been added to the authorized key file on the SSH server.
3. The private key did not have a matching public key on the SSH server.

Iterating through available authentication methods:

```
Authentications that can continue: publickey,keyboard-interactive,password
Next authentication method: publickey
Authentications that can continue: keyboard-interactive, password
Next authentication method: keyboard-interactive
Authentications that can continue: password
Next authentication method: password
Disconnecting from gen75 port 22
Error while connecting to SSH server (host: gen75 user: Administrator port: 22)
com.jcraft.jsch.JSchException: Auth fail
```

Incorrect Public Key Path

The private key path configured in the G*plus* Adapter options is incorrect and the transfer will fail prior to attempting to connect to the remote server.

```
Private key not found at 'wrongPath'
```

Public Key Passphrase Authentication Failed

The private key was generated with a passphrase but the passphrase was missing from the configuration or it was incorrect.

Iterating through available authentication methods:

```
Authentications that can continue: publickey,keyboard-interactive,password
Next authentication method: publickey
Disconnecting from gen75 port 22
Error while connecting to SSH server (host: gen75 user: Administrator port: 22)
com.jcraft.jsch.JSchException: Auth fail
```

Username-Password Authentication

Password Authentication Not Enabled

Password authentication was not enabled in the SSH server configuration.

Iterating through available authentication methods:

```
Authentications that can continue: publickey,keyboard-interactive
Next authentication method: publickey
Authentications that can continue: keyboard-interactive
Next authentication method: keyboard-interactive
Disconnecting from gen75 port 22
Error while connecting to SSH server (host: gen75 user: Administrator port: 22)
com.jcraft.jsch.JSchException: Auth fail
```

Failed Authentication

1. The username configured for the client does not match one of the usernames configured on the SSH server.
2. The plain text password configured for the client is incorrect for the user identified by the username.
3. The password configured for the client is blank and `PermitEmptyPasswords` has not been set in the SSH server configuration.

Iterating through available authentication methods:

```
Authentications that can continue: publickey,keyboard-interactive,password
Next authentication method: publickey
Authentications that can continue: keyboard-interactive,password
Next authentication method: keyboard-interactive
Authentications that can continue: password
Next authentication method: password
Disconnecting from gen75 port 22
Error while connecting to SSH server (host: gen75 user: Admin port: 22)
com.jcraft.jsch.JSchException: Auth fail
```

Problems with Transfer

Inaccessible Destination Directory or Insufficient Access Privileges

1. The destination directory cannot be reached as a sub-directory from the user home directory on the SSH server.
2. The user identified by the username has the wrong or insufficient privileges for the destination directory on the SSH server.

File transfer:

```
Attempting to transfer file: 'C:\testFiles\textFile.txt' to 'c:/'
Unable to transfer file: 'C:\testFiles\textFile.txt', check that destination directory exists
Failure
        at com.jcraft.jsch.ChannelSftp.throwStatusError(ChannelSftp.java:2289)
        ....
```

Non-Existent Destination Directory

The destination directory on the SSH server does not exist.

There will be no error indication for a missing destination directory if it was configured as a subdirectory of the user's home directory on the remote server. Despite the logged statements, the file will not have been transferred to the destination directory as the directory will not be created. It is likely that the file was transferred but its location and name may be a combination of the valid and invalid portions of the destination path.

File transfer:

```
Attempting to transfer file: 'C:\testUser\testFiles\textFile.txt' to 'sftpDestination'
Successfully transferred file: 'C:\testUser\testFiles\textFile.txt'
```

Closing connection to remote server:

```
Disconnecting from gen75 port 22
Caught an exception, leaving main loop due to socket closed
```

Using the above as an example, if the `testFiles` directory did not exist in the `testUser` directory, a successful transfer might see a new file, `testFilestextFile.txt`, located in the `testUser` directory. The actual file location and name will be dependent upon the portions of the destination path that are incorrect, the type of SSH server and the underlying operating system for the SSH server

Appendix

C Managing Windows Service

prunmgr Application

The Apache Commons™ *daemon* ™ project also provides a Windows GUI application for managing an installed Service without having to resort to editing Register parameters or reinstalling the Service. This *prunmgr* application will be discussed below with respect to the G*plus* Adapter installation and there is additional information available on the Apache website. The application is contained in the `wrappers` folder of the Adapter installation and is started with the following:

```
prunmgr.exe //ES//GPlusAdapterServiceName
```

Any changes to the configuration can be established by stopping and restarting the Service after applying the changes. The options and parameters appropriate to the Adapter Service wrapper are discussed below.

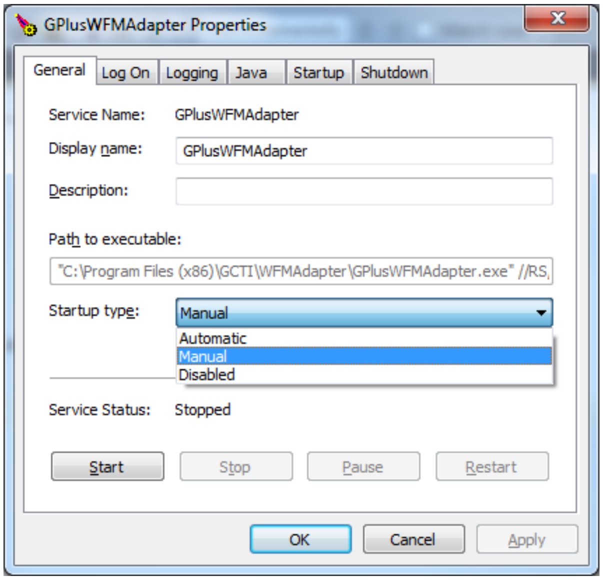
General Tab

Startup type:

This field should be set to *Automatic* so that the Service will restart if the host server is restarted.

Start, Stop and ***Restart*** Buttons

The Service can be started, stopped and restarted. Pausing the Service would have indeterminate results that would probably warrant a restart.



GPlusWFMAdapter Properties

Service Name: GPlusWFMAdapter

Display name: GPlusWFMAdapter

Description:

Path to executable:

"C:\Program Files (x86)\GCTI\WFMAdapter\GPlusWFMAdapter.exe" //RS,

Startup type: Manual

Automatic
Manual
Disabled

Service Status: Stopped

Logging Tab

It is possible to log the Adapter Service which is helpful when encountering problems with getting the Service properly installed and running.

Log prefix:

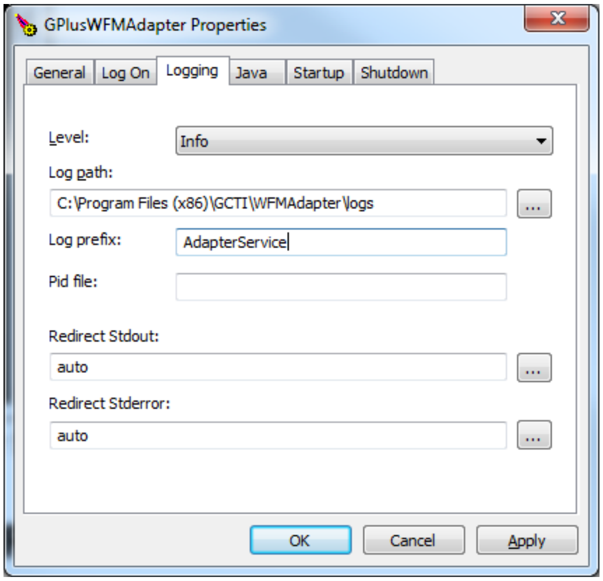
Specifying this parameter will enable logging to the *Log path* folder with the prefix as the start of the log name combined with a `.YEAR-MONTH-DAY.log` suffix.

Log path:

The default is the Adapter's `logs` folder but this parameter can be set to another folder if required.

Redirect Stdout: and ***Redirect Stderr:***

Both `stdout` and `stderr` can be redirected to files in the *Log path* folder if there is a need to debug JVM problems that are not logged in the regular adapter logs. Both are set to `auto` in the `svcinstall.bat` file and that enables the logging with the default file names.



GPlusWFMAdapter Properties

Level: Info

Log path: C:\Program Files (x86)\GCTI\WFMAdapter\logs

Log prefix: AdapterService

Pid file:

Redirect Stdout: auto

Redirect Stderror: auto

Java Tab

The *Java* tab configures the operating parameters for the Java Virtual Machine associated with the Service. The default configuration is shown below with the exception of the *Java Virtual Machine* parameter that is installation specific.

Java Virtual Machine*:*

This parameter identifies the path to the `jvm.dll` file that will be linked by the Service wrapper to host the Adapter application. The `.dll` file should be a “server” version as described in Chapter 2.

Java Classpath:

This parameter must not be modified from the default, `bootstrap.jar`.

Initial memory pool:

This parameter specifies the amount of system memory that will be allocated to the JVM at startup. This parameter is set to the same amount as the *Maximum memory pool* so that all of the required memory is requested at startup and if the system cannot provide it at that time, the JVM shuts down. This avoids memory problems arising hours after startup because there is insufficient system memory.

Default = `1024 MB`

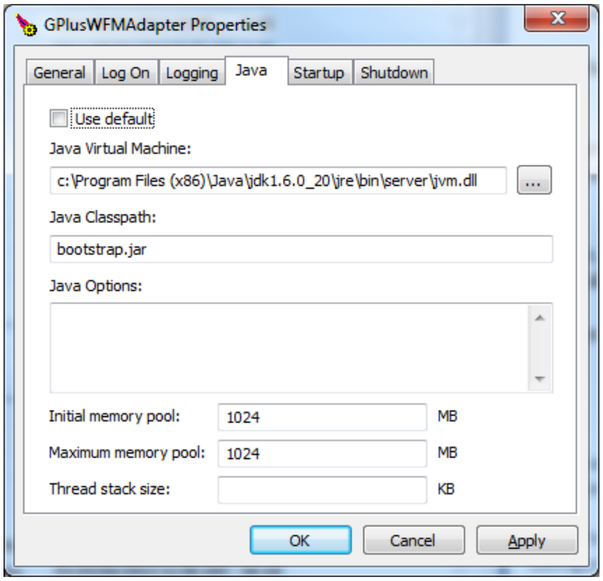
Maximum memory pool:

This parameter specifies the maximum amount of system memory that will be requested by the JVM.

Default = `1024 MB`

Java Options:

This section allows any number of parameters to be added to the JVM configuration. Presently, the parameters that might be added are related to configuring the memory Garbage Collection (GC) and adding logging of the GC activity.



☐ Use default

Java Virtual Machine: c:\Program Files (x86)\Java\jdk1.6.0_20\jre\bin\server\jvm.dll

Java Classpath: bootstrap.jar

Java Options:

Initial memory pool: 1024 MB

Maximum memory pool: 1024 MB

Thread stack size: KB

Startup Tab

Class:

This field specifies the Java class that contains the main method that will be invoked at startup. This cannot be changed.

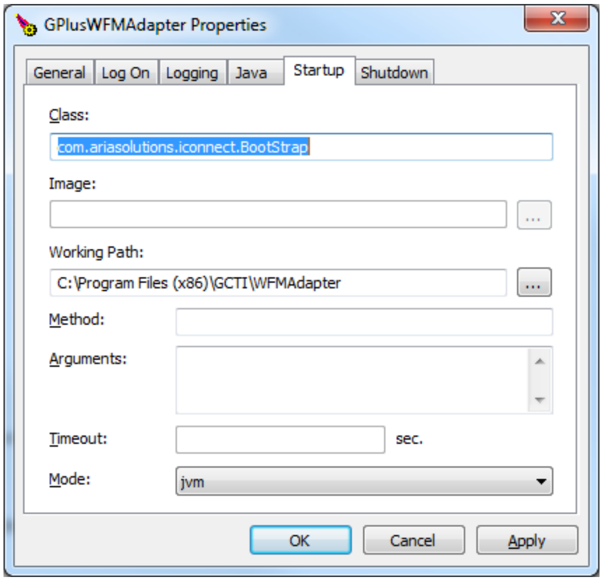
Default = `com.ariasolutions.iconnect.BootStrap`

Working Path:

This field specifies the root folder for the Adapter installation. It is dependent upon the *Path to executable* field on the *General Tab* and should not be changed.

Mode:

The proper operation of the Service wrapper is dependent upon this field remaining set to `jvm`.



Class: com.ariasolutions.iconnect.BootStrap

Image:

Working Path: C:\Program Files (x86)\GCTI\WFMAdapter

Method:

Arguments:

Timeout: sec.

Mode: jvm

Genesys

Appendix

D Log Files

Files

Depending upon the logging configuration discussed above, there may be more than one set of logs contained in the logging folder. This chapter is concerned with the event logs that also include logging of state changes for Agents and media interactions.. There will typically be two versions of the event logs contained in the `logs` folder: `ROLLING` and `GZIP`. The actual names of the logs are specified in the `log4j.xml` file but unless they have been changed in that file, the log files will have the prefix of `GPlusAdapterWFM`.

The default size for each of the `ROLLING` logs is `50 MB` and they will be rolling over quickly in a large call center. These logs give a snapshot of activity at the time that they are checked but may not be suitable for tracking down problems with the adapter if they cover too short an interval. The most recent of the log files will have a name starting with the prefix described above and then the `.log` extension. Earlier logs will have numbers added to the extension with `.log.1` being the next most recent and `.log.2` being earlier again.

Each of the zipped logs contain exactly the same information as the `ROLLING` logs but summarize an hour of activity and the time at the start of that hour is contained in the log file name:

```
GPlusAdapterWFM.log.2012.01.01-16.00.00.gz
```

The time field will always be an even hour except for the log created at startup. That file will have a time stamp indicating when the logging started:

```
GPlusAdapterWFM.log.2012.01.01-15.50.00.gz
```

This log will contain useful startup information not available in the other logs. If the Adapter has been running for an extended period of time, the startup log will have been removed as older zipped logs are also deleted to prevent filling up the disk storage.

Description of Fields

The log files are intended as an aid in debugging the application when problems do occur. The files have a simple format that is described below using the first line of the log file as it starts up as an example.

```
01 Jan 2012 12:00:02,000 · 2000 [main] INFO com.ariasolutions.iconnect.Application - Launching GPlus Adapter
```

1. `01 Jan 2012 12:00:02,000`
2. `2000`
3. `[main]`
4. `INFO`
5. `com.ariasolutions.iconnect.Application`
6. `Launching GPlus Adapter`

The **first** field shows the date and time that the event or update occurred.

The **second** field shows the number of milliseconds that the application has been running.

The **third** field is the thread name that produced the message.

The **fourth** field is the logging level that the message has been assigned.

The **fifth** field is the Java class (and stream) that has generated the message.

The **sixth** field is the description of what has occurred to generate the message.

Logging Threshold and Messages

The logging threshold has more relevance as the types of messages that are generated by the Adapter have been rationalized. The `info` mode tracks significant events such as report generation and TServer connections but is restricted to logging agent activity by the changes in RTA states. The `debug` mode adds more detail to the agent activity as well as logging the progress of interactions (voice calls, emails etc.) from monitored queues to the agent desktops. The `trace` mode provides the most detail with full logging of the incoming events and includes the `PlaceListenerLogger` and the `QueueListenerLogger` discussed below.

Both the `info` and `debug` modes result in log files that are significantly smaller than those generated with the `trace` mode.

info mode example

The following is an example of what is logged when the logThreshold is set to info. It shows the messages generated when an agent receives an inbound customer call.

```
01 Jan 2014 12:01:00,000 90000 [main] INFO  com.ariasolutions.iconnect.rta.StateHolder Stream1 -
StateHolder.notifyStateChange: msmith: StateInfo(CallInbound, Mon Jan 01 12:01:00 MDT 2014, null,
2500, VoiceSales_VQ, )
```

Just the single line is generated showing the change in the agent's state. It should be noted that the stream name of Stream1 has been added to the class name that generated the message. If the message was not associated with a configured stream, then a stream name of default would have been logged.

debug mode example

As can be seen, the debug mode adds call tracking.

```
01 Jan 2014 12:00:45,000 75000 [main] DEBUG
com.ariasolutions.iconnect.eventprocessing.VoiceEventProcessor  - Call 00b102420f4e2002 placed in
Queue VoiceSales_VQ
01 Jan 2014 12:00:55,000 85000 [main] DEBUG
com.ariasolutions.iconnect.eventprocessing.VoiceEventProcessor  - Call 00b102420f4e2002 ringing on
Place Place_2500 (Agent msmith)
01 Jan 2014 12:00:59,000 89000 [main] INFO  com.ariasolutions.iconnect.rta.StateHolder default -
StateHolder.notifyStateChange: msmith: StateInfo(CallInbound, Mon Jan 01 12:00:59 MDT 2014, null,
2500, VoiceSales_VQ, )
01 Jan 2014 12:01:00,000 90000 [main] DEBUG
com.ariasolutions.iconnect.eventprocessing.VoiceEventProcessor  - Call 00b102420f4e2002 established
on Place Place_2500 (Agent msmith) Pegged to Queue VoiceSales_VQ
```

trace mode

The trace mode logging for the above example would be several pages long as the full events are logged from the initial queued event to the EventEstablished on the agent's desktop. The extra logging is discussed below in the next three sections.

Events and Server Identification

Events are logged in `trace` mode just as they are received from the connected Genesys Server: Interaction Server, SIPServer or TServer. Configuration Server events are logged but not with the detail described below. The following shows a TServer event generated when an Agent answered an incoming call:

```
24 May 2012 21:59:37,183 - 168858 ... - Received event from 'TServer1'
EventEstablished
  ANI: 4032500001
  AgentID: 200000
  CallState: 0
  CallType: Inbound
  ConnID: 000000000005c85e
  EventSequenceNumber: 112318
  GPlusWFM.LocalTime: 1337885977167
  GPlusWFM.RemoteTime: 1337885977201
  GPlusWFM.SourceId: 133
  GPlusWFM.SourceName: TServer1
  GPlusWFM.Time: 1337885977201
  OtherDN: 4032500002
  OtherDNRole: RoleOrigination
  ThisDN: Test_2000
  ThisDNRole: RoleDestination
  ThisTrunk: 9999
  TimeinSecs: 1337885977
  TimeinuSecs: 201000
  UserData.SampleBinary1: [B@1c668df
  UserData.SampleInt1: 1
  UserData.inner.inner sample: sample
  UserData.inner.inner int: 3
  UserData.SampleString1: Sample
```

Some of the enumerated event attributes such as `CallType` are replaced with the value name to make the logged event easier to read. The adapter also adds attributes with the `GPlusWFM` prefix for the Adapter's host time (`GPlusWFM.LocalTime`) and the Genesys Server time (`GPlusWFM.RemoteTime`). The `GPlusWFM.Time` attribute indicates which of the two time sources the Adapter is actually using for the event time as selected by the application/useLocalTime configuration option.

The `GPlusWFM.SourceId` and `GPlusWFM.SourceName` for the originating Genesys Server are also included.

PlaceListenerLogger

Prior to the event listed above, the Adapter added some state change notifications that were associated with the `EventEstablished`. The first notification is logged by the `PlaceListenerLogger` and indicates the answering Agent's state:

```
24 May 2012 21:59:37,183 - 168858 ... PlaceListenerLogger  - PlaceState updated to:
PlaceState for Place: [PlaceName: Place_2500001, site: Site_1]
  LoginSession: [LoginID: 200000, AgentUsername: Agent_200000, DoNotDisturb: off]
    MediaSession: [MediaType: voice, ReadyState: Ready]
      Interaction: [ID: 000000000005c85e, Target: VQ_Test]
```

Although the receiving agent was not identified in the `EventEstablished` attributes, the `PlaceListenerLogger` has identified the `Place` and its associated `Site` for the logged in agent. The agent is indicated as in a `Ready` state for `voice` calls and is presently handling one call that was queued on a monitored `VQ`. If the agent was logged in to additional media (`email`, `chat` etc.), there would be additional media sessions listed as well as any ongoing interactions associated with those sessions.

Note: The absence of any of the above information (with the exception of `Site`) would indicate a configuration problem as there was insufficient information to create and maintain the required internal associations.

QueueListenerLogger

The second state change notification associated with the `EventEstablished` was generated by the `QueueListenerLogger` which logs the state changes for the incoming call. There is not a lot of detail other than logging the target with which the call had been associated and logging the state of the call in generic terms that apply to routing of any media:

```
24 May 2012 21:59:37,183 - 168858 ... QueueListenerLogger  - Interaction 'answered':
QueuedInteraction [id=000000000005c85e, target=VQ_4847]
```

There are six available states for a queued interaction and they are sufficient to indicate the part of a "call flow" the interaction had reached when the event was generated:

- `offered`
- `abandoned`
- `expired`
- `answered`
- `released`
- `exitQueue`

The QueueListenerLogger had already been logging the interaction state of the call prior to the agent answering it:

```
24 May 2012 21:59:33,613 - 165288 ... QueueListenerLogger  -
Interaction 'offered': QueuedInteraction [id=000000000005c85e,
target=VQ_4847]
24 May 2012 21:59:33,613 - 165288 ... - Received event from 'TServer2'
EventQueued
  CallType: Inbound
  ConnID: 000000000005c85e
  EventSequenceNumber: 112574
  GPlusWFM.LocalTime: 1337885973613
  GPlusWFM.RemoteTime: 1337885973638
  GPlusWFM.SourceId: 135
  GPlusWFM.SourceName: TServer2
  GPlusWFM.Time: 1337885973638
  OtherDN: 4032350227
  OtherDNRole: RoleOrigination
  ThisDN: VQ_4847
  ThisDNRole: RoleDestination
  ThisQueue: VQ_4847
  TimeinSecs: 1337885973
  TimeinuSecs: 638000
```

Note: The QueueListenerLogger only logs calls or interactions that the adapter has identified as a "customer call". If the logger has not identified an interaction as 'offered' prior to the first queued event for the interaction, then none of the activity for that interaction will be associated with a VQ or target.

Genesys

Appendix

E Windows Installation Script

Files

The Windows installation script retains the configuration parameters in two files:

- `config/jvm.properties`
  - contains the location of the jvm.dll and the memory setting
  - required to install the Adapter as a Windows Service
- `config/GPlusWFM.properties`
  - contains the Application name and the Configuration Server connection parameters
  - the Application name is required for the Service installation
  - the connection parameters are required for the Adapter to run but not for the Service installation

A third file, `config/product.properties`, contains the WFM vendor, Verint, as well as the version number for the specific release that has been installed. This file is not modified by the installation script but the vendor and release version are displayed.

The Adapter also displays the relevant contents of the license file.

Invoking the Installation Script

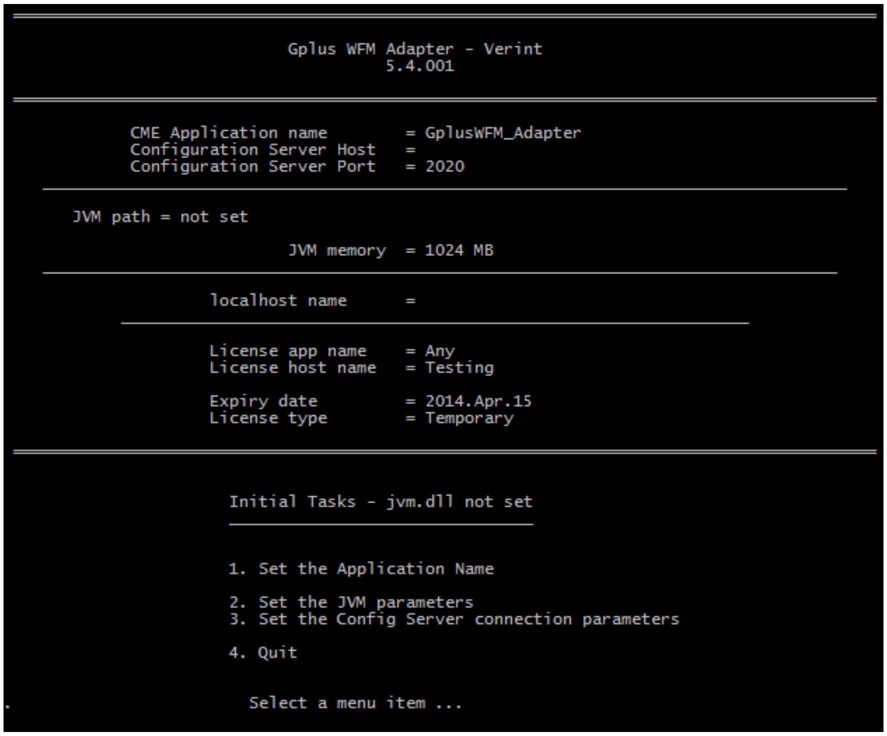
It is important that the script execute with Administrator privileges so that the Windows service can be installed and the properties files modified. This can be accomplished by right clicking on the `windows_install.bat` file and selecting *Run as administrator*.

Initial Screen and Status Display

There are three main menus that will be displayed depending on the context of the Adapter installation:

- *Initial Tasks – jvm.dll not set*
- *Installation Tasks – adapter service not installed*
- *Management Tasks – adapter service installed*

The first, *Initial Tasks*, will appear until the path to the `jvm.dll` file has been set at which time the second menu will be used.

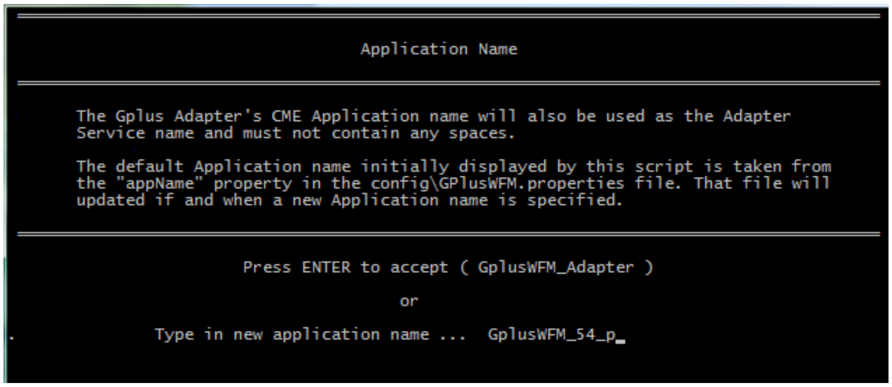


```
                    Gplus WFM Adapter - Verint
                             5.4.001

        CME Application name        = GplusWFM_Adapter
        Configuration Server Host   =
        Configuration Server Port   = 2020

    JVM path = not set

                    JVM memory  = 1024 MB

                localhost name      =

                License app name    = Any
                License host name   = Testing

                Expiry date         = 2014.Apr.15
                License type        = Temporary

                  Initial Tasks - jvm.dll not set

                  1. Set the Application Name

                  2. Set the JVM parameters
                  3. Set the Config Server connection parameters

                  4. Quit

.                   Select a menu item ...
```

The screenshot above shows the default settings from the properties files as well as the parameters for the temporary license provided with the Adapter release.

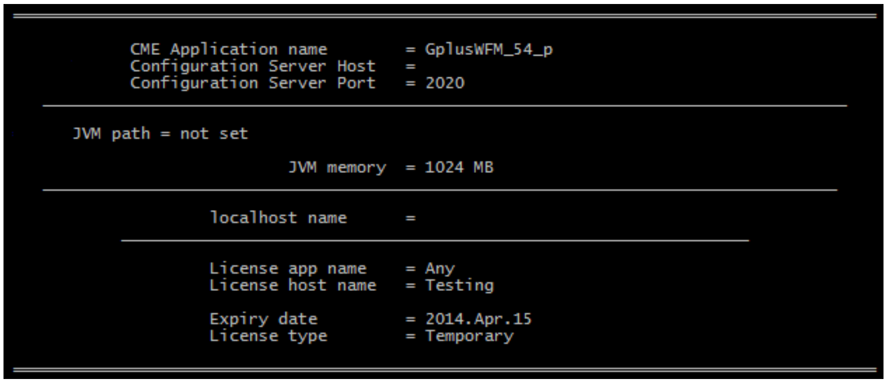
Set the Application name

The Application name should be set first. This name must match the name of the G*plus* Adapter's Application object in the Configuration Server. The name will be used for the name of the installed service and the renamed `prunsrv.exe` (Java service wrapper) that will be copied to the Adapter's main folder.



```
                              Application Name

      The Gplus Adapter's CME Application name will also be used as the Adapter
      Service name and must not contain any spaces.

      The default Application name initially displayed by this script is taken from
      the "appName" property in the config\GPlusWFM.properties file. That file will
      updated if and when a new Application name is specified.

                      Press ENTER to accept ( GplusWFM_Adapter )

                                        or

              Type in new application name ...  GplusWFM_54_p_
```

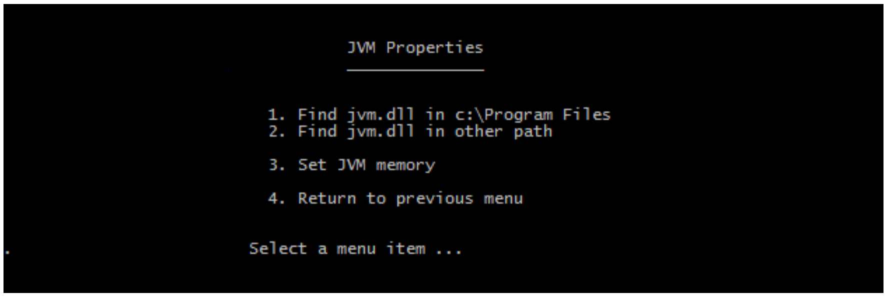
The new name is immediately written to the `GPlusWFM.properties` file and will be displayed in the status section as shown below:



```
            CME Application name        = GplusWFM_54_p
            Configuration Server Host   =
            Configuration Server Port   = 2020
      ---------------------------------------------------------------------------------
      JVM path = not set

                      JVM memory  = 1024 MB
      ---------------------------------------------------------------------------------
                    localhost name      =
              -----------------------------------------------------------------
                    License app name    = Any
                    License host name   = Testing

                    Expiry date         = 2014.Apr.15
                    License type        = Temporary
```

Set the JVM parameters

This menu item brings up a second menu that allows the Java Runtime Environment to be configured. Once this has been completed, the Adapter can be installed as a service.

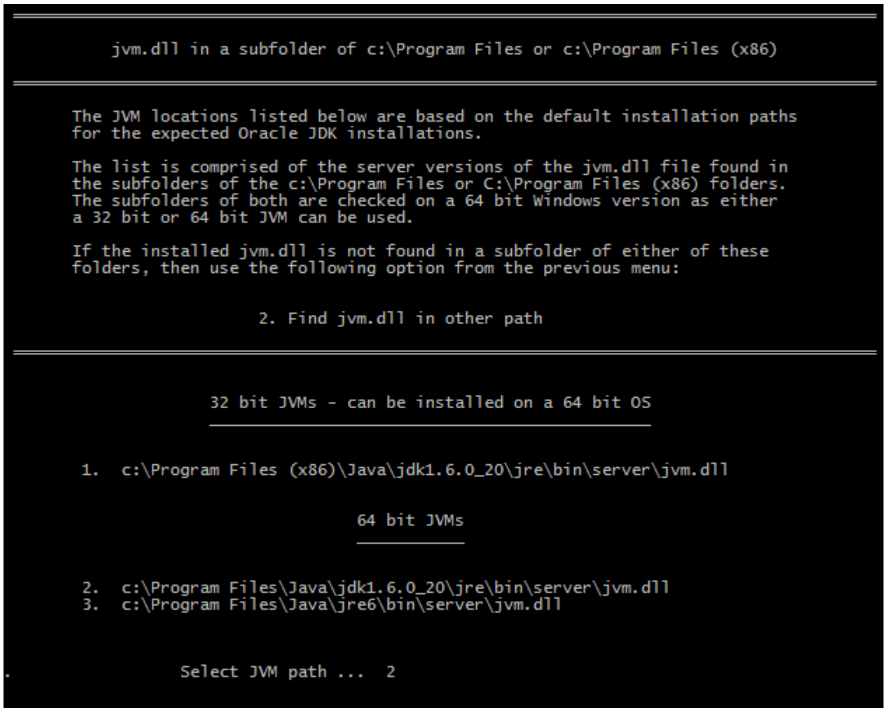


```
                          JVM Properties
                          ______________

                  1. Find jvm.dll in c:\Program Files
                  2. Find jvm.dll in other path

                  3. Set JVM memory

                  4. Return to previous menu

.                Select a menu item ...
```

There are two different menu options for finding the path to the jvm.dll file. The first just looks in the default locations that the Oracle installation application will place a 32 bit or 64 bit version while the second option allows a path to be set with all of the subfolders being searched.

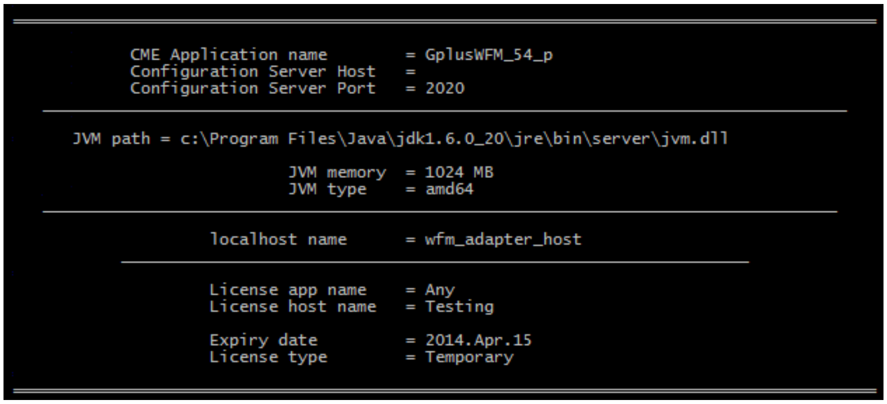
Default JVM Locations

The first menu selection will list all of the jvm.dll files found in the default installation locations. Both 32 bit and 64 bit JVM versions will be listed if present on a 64 bit OS. The Adapter will run in either.



```
==========================================================================

      jvm.dll in a subfolder of c:\Program Files or c:\Program Files (x86)

==========================================================================

      The JVM locations listed below are based on the default installation paths
      for the expected Oracle JDK installations.

      The list is comprised of the server versions of the jvm.dll file found in
      the subfolders of the c:\Program Files or C:\Program Files (x86) folders.
      The subfolders of both are checked on a 64 bit Windows version as either
      a 32 bit or 64 bit JVM can be used.

      If the installed jvm.dll is not found in a subfolder of either of these
      folders, then use the following option from the previous menu:

                        2. Find jvm.dll in other path

==========================================================================

                  32 bit JVMs - can be installed on a 64 bit OS
                  _____________________________________________

       1.  c:\Program Files (x86)\Java\jdk1.6.0_20\jre\bin\server\jvm.dll

                               64 bit JVMs
                               ___________

       2.  c:\Program Files\Java\jdk1.6.0_20\jre\bin\server\jvm.dll
       3.  c:\Program Files\Java\jre6\bin\server\jvm.dll


.                 Select JVM path ...  2
```

The change is immediately written to the `config\jvm.properties` file and the status section reflects the specification of the 64 bit JVM as shown below.



```
          CME Application name        = GplusWFM_54_p
          Configuration Server Host   =
          Configuration Server Port   = 2020

   JVM path = c:\Program Files\Java\jdk1.6.0_20\jre\bin\server\jvm.dll

                          JVM memory  = 1024 MB
                          JVM type    = amd64

                  localhost name      = wfm_adapter_host

                  License app name    = Any
                  License host name   = Testing

                  Expiry date         = 2014.Apr.15
                  License type        = Temporary
```

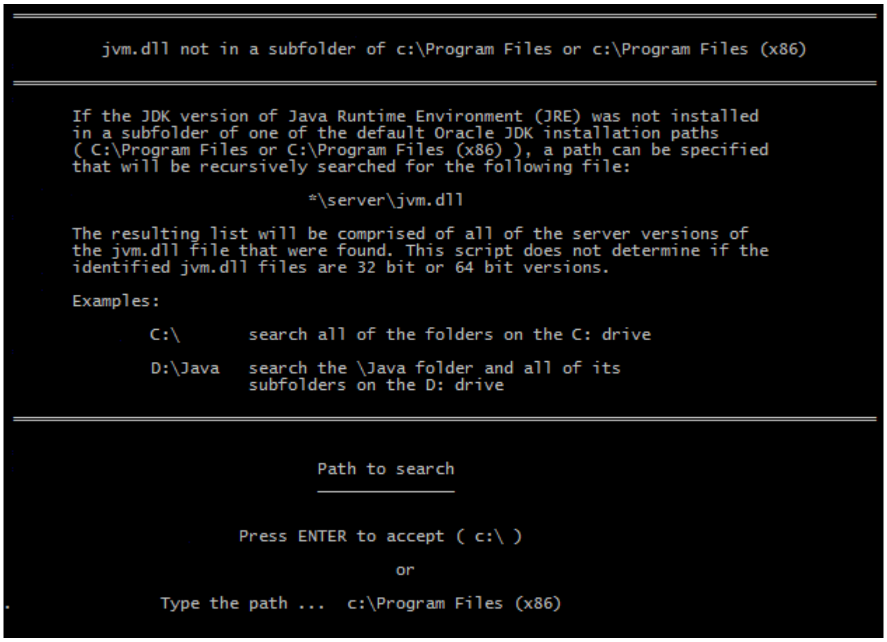
There are two things to note in the status section shown above beyond the fact that the path to the `jvm.dll` file has been identified.

First, a new parameter, `JVM type`, has been added. This indicates which of the prunsrv.exe service wrappers will be used. In the example above, the 64 bit non-Itanium™ version will be used.

The second addition is that the `localhost name` parameter has also been set. This is the host name that should be used when requesting a permanent license for the Adapter installation. Once the JVM path was set, a Java application was run to determine the hostname with exactly the same code that is used to confirm the hostname when validating the license.

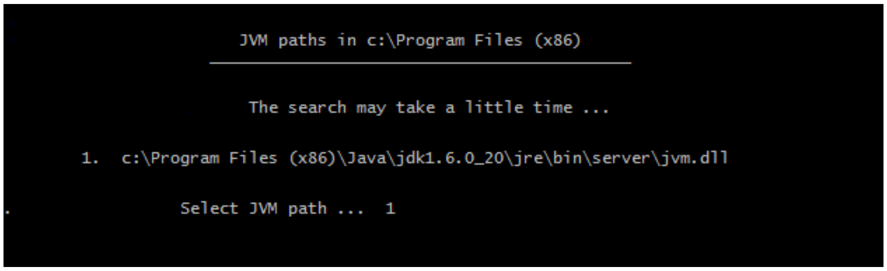
Set path to find `jvm.dll`

If the Java instance has not been installed in a subfolder of either `c:\Program Files` or `c:\Program Files (x86)`, the script will search for the `jvm.dll` file in other locations.



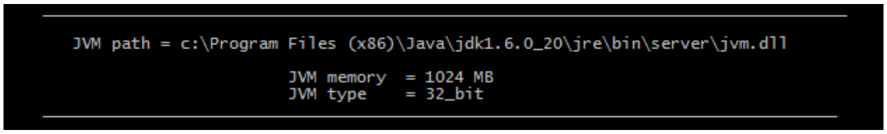
```
 ==============================================================================

          jvm.dll not in a subfolder of c:\Program Files or c:\Program Files (x86)

 ==============================================================================

      If the JDK version of Java Runtime Environment (JRE) was not installed
      in a subfolder of one of the default Oracle JDK installation paths
      ( C:\Program Files or C:\Program Files (x86) ), a path can be specified
      that will be recursively searched for the following file:

                              *\server\jvm.dll

      The resulting list will be comprised of all of the server versions of
      the jvm.dll file that were found. This script does not determine if the
      identified jvm.dll files are 32 bit or 64 bit versions.

      Examples:

              C:\       search all of the folders on the C: drive

              D:\Java   search the \Java folder and all of its
                        subfolders on the D: drive

 ==============================================================================

                               Path to search
                               ______________

                       Press ENTER to accept ( c:\ )

                                     or

.              Type the path ...  c:\Program Files (x86)
```

The result will be a list of the paths to any found `jvm.dll` files but there will no indication as to whether the files are 32 bit or 64 bit versions.



```
                         JVM paths in c:\Program Files (x86)
                     ___________________________________________

                          The search may take a little time ...

        1.  c:\Program Files (x86)\Java\jdk1.6.0_20\jre\bin\server\jvm.dll

.                 Select JVM path ...  1
```

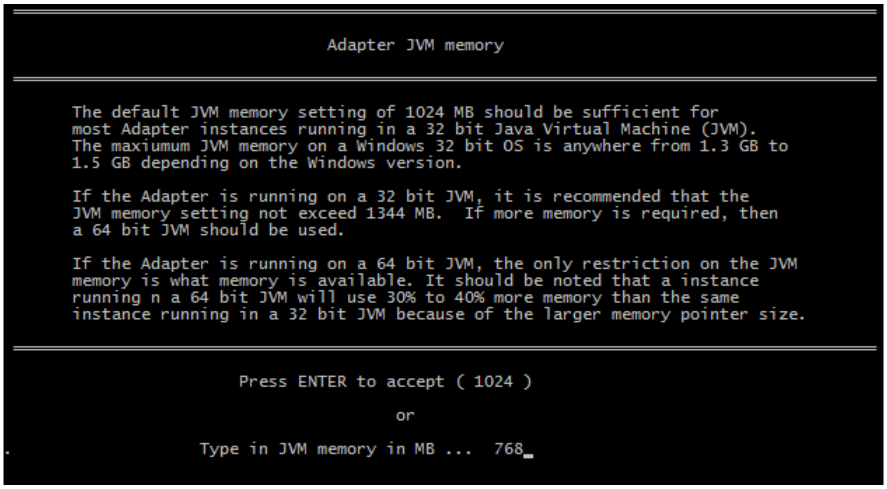
The script can determine the bitness of the JVM once it has been selected as shown below where the `JVM type` has been changed to `32_bit`.



```
    ___________________________________________________________________________

       JVM path = c:\Program Files (x86)\Java\jdk1.6.0_20\jre\bin\server\jvm.dll

                           JVM memory  = 1024 MB
                           JVM type    = 32_bit
    ___________________________________________________________________________
```

Set JVM memory

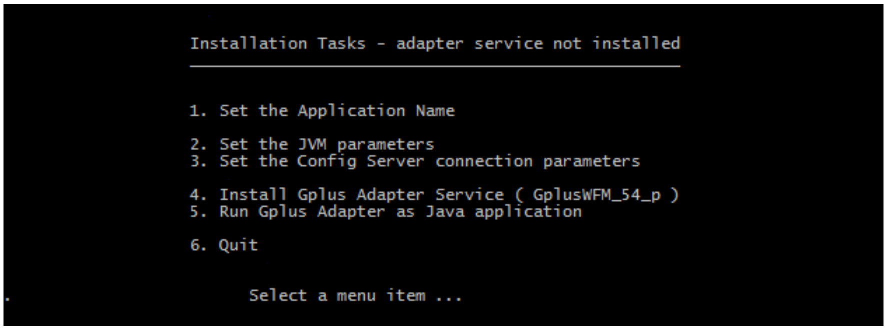
The amount of memory that the JVM will allocate as it starts up can be set to something other than the default of 1024 MBs.



```
                         Adapter JVM memory

      The default JVM memory setting of 1024 MB should be sufficient for
      most Adapter instances running in a 32 bit Java Virtual Machine (JVM).
      The maxiumum JVM memory on a Windows 32 bit OS is anywhere from 1.3 GB to
      1.5 GB depending on the Windows version.

      If the Adapter is running on a 32 bit JVM, it is recommended that the
      JVM memory setting not exceed 1344 MB.  If more memory is required, then
      a 64 bit JVM should be used.

      If the Adapter is running on a 64 bit JVM, the only restriction on the JVM
      memory is what memory is available. It should be noted that a instance
      running n a 64 bit JVM will use 30% to 40% more memory than the same
      instance running in a 32 bit JVM because of the larger memory pointer size.

                    Press ENTER to accept ( 1024 )

                                   or

.                 Type in JVM memory in MB ...  768_
```

Installation Tasks

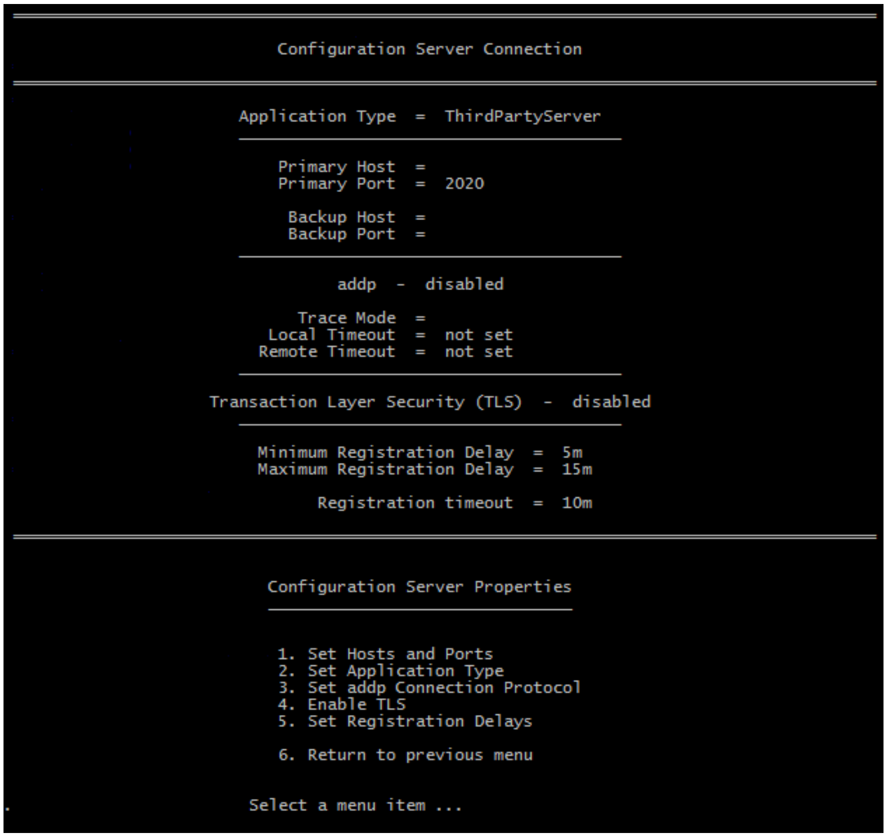
Once the path to the jvm.dll file is set, the Gplus Adapter can be installed as a Windows Service. The main menu reflects this change in status with the addition of two more menu options. It is possible to install the service with no issues but starting that service or running the Adapter as a console application will fail if the Configuration Server connection has not be configured.



```
                   Installation Tasks - adapter service not installed
                   __________________________________________________

                   1. Set the Application Name

                   2. Set the JVM parameters
                   3. Set the Config Server connection parameters

                   4. Install Gplus Adapter Service ( GplusWFM_54_p )
                   5. Run Gplus Adapter as Java application

                   6. Quit

.                        Select a menu item ...
```

Set the Config Server connection parameters

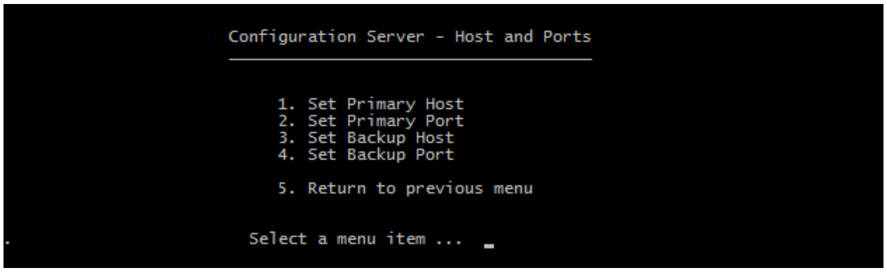
Selecting this menu option allows the script to modify the contents of the `GPlusWFM.properties` file which contains all of the parameters that can set on the connection.



```
                    Configuration Server Connection

                 Application Type  =  ThirdPartyServer

                     Primary Host  =
                     Primary Port  =  2020

                      Backup Host  =
                      Backup Port  =

                           addp  -  disabled

                       Trace Mode  =
                    Local Timeout  =  not set
                   Remote Timeout  =  not set

              Transaction Layer Security (TLS)  -  disabled

                 Minimum Registration Delay  =  5m
                 Maximum Registration Delay  =  15m

                       Registration timeout  =  10m

                    Configuration Server Properties

                     1. Set Hosts and Ports
                     2. Set Application Type
                     3. Set addp Connection Protocol
                     4. Enable TLS
                     5. Set Registration Delays

                     6. Return to previous menu

                  Select a menu item ...
```

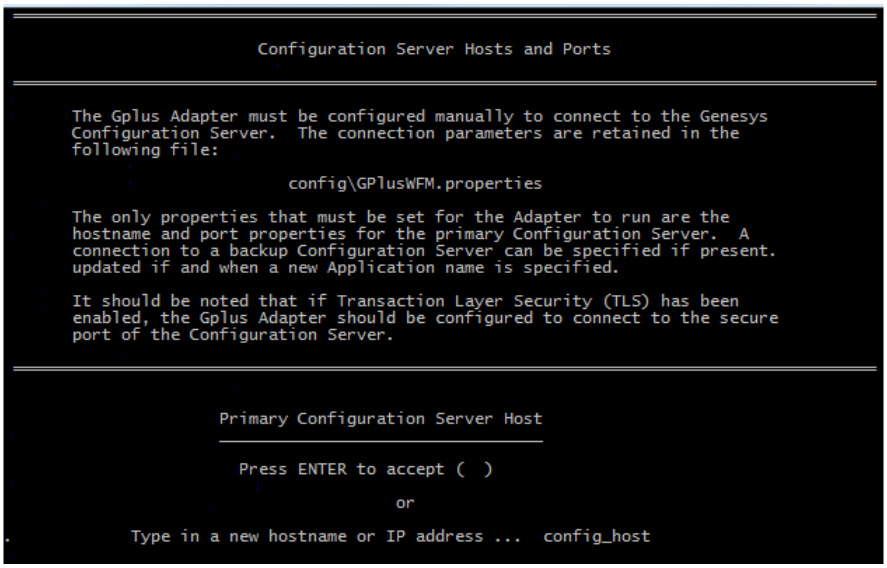
Hosts and ports

The host and port can be set for the primary Configuration Server and its backup.



```
                 Configuration Server - Host and Ports

                      1. Set Primary Host
                      2. Set Primary Port
                      3. Set Backup Host
                      4. Set Backup Port

                      5. Return to previous menu

                   Select a menu item ...
```

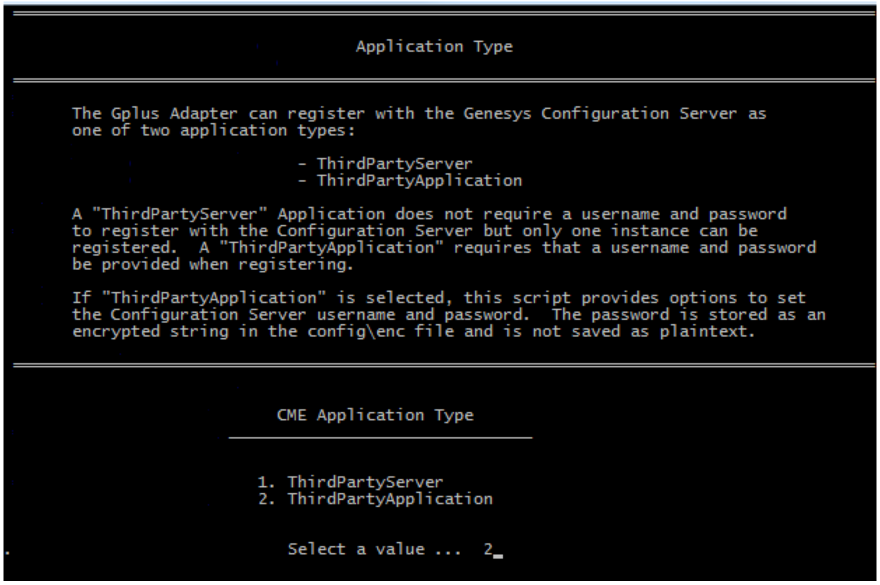
The host can be specified as with the hostname or the IP address. If TLS has been enabled, the configured port should be the Configuration Server's secure port rather than the default.



```
                    Configuration Server Hosts and Ports

    The Gplus Adapter must be configured manually to connect to the Genesys
    Configuration Server.  The connection parameters are retained in the
    following file:

                          config\GPlusWFM.properties

    The only properties that must be set for the Adapter to run are the
    hostname and port properties for the primary Configuration Server.  A
    connection to a backup Configuration Server can be specified if present.
    updated if and when a new Application name is specified.

    It should be noted that if Transaction Layer Security (TLS) has been
    enabled, the Gplus Adapter should be configured to connect to the secure
    port of the Configuration Server.

                  Primary Configuration Server Host
                  _________________________________

                    Press ENTER to accept (  )

                                    or

.            Type in a new hostname or IP address ...  config_host
```

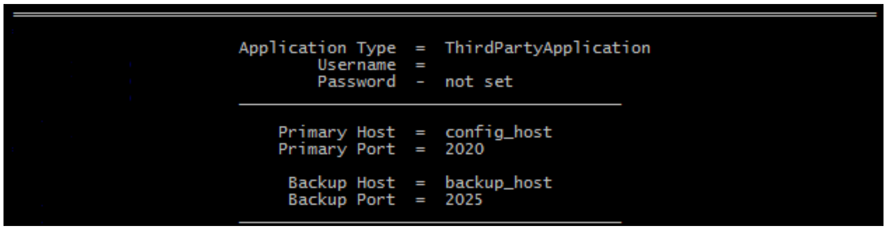
Set Application type

The Adapter's Application type affects the parameters that will be sent to the Configuration Server as part of the Adapter's registration request. A `ThirdPartyServer` type does not provide a `username` or `password` but is only allowed one connection. The alternative, `ThirdPartyApplication`, requires the `username` and `password` but provides better security.



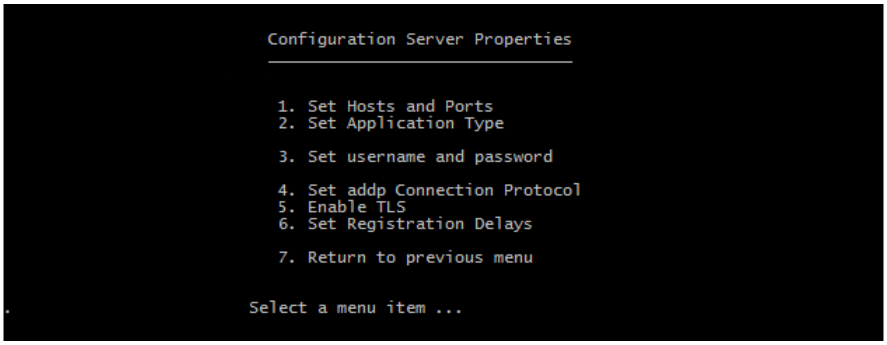
```
                            Application Type

    The Gplus Adapter can register with the Genesys Configuration Server as
    one of two application types:

                       - ThirdPartyServer
                       - ThirdPartyApplication

    A "ThirdPartyServer" Application does not require a username and password
    to register with the Configuration Server but only one instance can be
    registered.  A "ThirdPartyApplication" requires that a username and password
    be provided when registering.

    If "ThirdPartyApplication" is selected, this script provides options to set
    the Configuration Server username and password.  The password is stored as an
    encrypted string in the config\enc file and is not saved as plaintext.

                       CME Application Type
                   _______________________________

                   1. ThirdPartyServer
                   2. ThirdPartyApplication

.                     Select a value ...  2_
```

ThirdPartyApplication



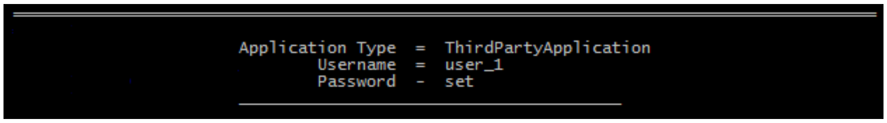
```
Application Type  =  ThirdPartyApplication
        Username  =
        Password  -  not set
_______________________________________

    Primary Host  =  config_host
    Primary Port  =  2020

     Backup Host  =  backup_host
     Backup Port  =  2025
_______________________________________
```

If ThirdPartyApplication is selected as the Application type, the script provides a menu option to set both the `username` and the `password`. It should be noted that the password is not saved as plain text.



```
   Configuration Server Properties
   _______________________________


    1. Set Hosts and Ports
    2. Set Application Type

    3. Set username and password

    4. Set addp Connection Protocol
    5. Enable TLS
    6. Set Registration Delays

    7. Return to previous menu


 Select a menu item ...
```

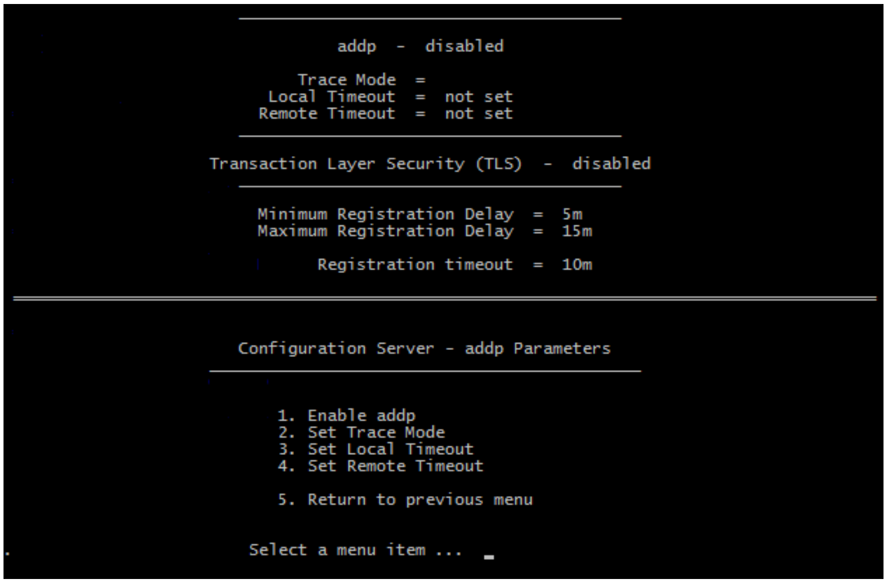
When the password is set, that string is encrypted and stored in the `config/enc` file. If that file is present, the script indicates that the password has been *set* but does not display it.



```
Application Type  =  ThirdPartyApplication
        Username  =  user_1
        Password  -  set
_______________________________________
```

Set addp Connection Protocol

The script provides a menu for configuring the addp connection protocol for the Configuration Server connection.

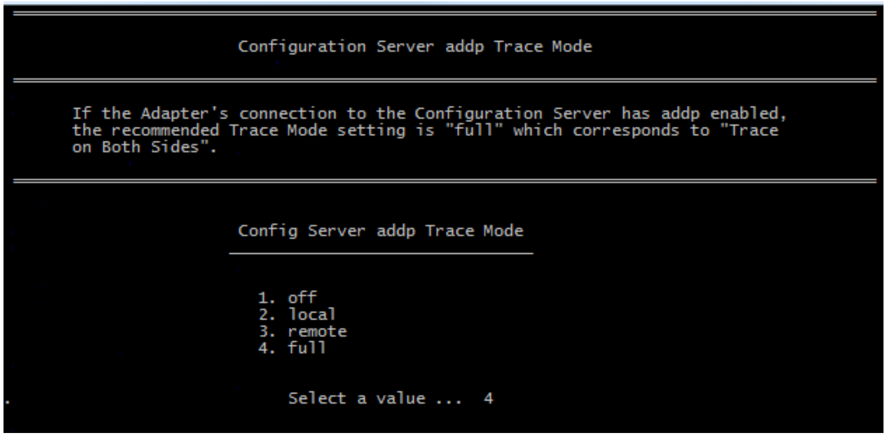


```
                         addp  -  disabled

                     Trace Mode  =
                  Local Timeout  =  not set
                 Remote Timeout  =  not set

           Transaction Layer Security (TLS)  -  disabled

             Minimum Registration Delay  =  5m
             Maximum Registration Delay  =  15m

                   Registration timeout  =  10m

==========================================================================

              Configuration Server - addp Parameters

                 1. Enable addp
                 2. Set Trace Mode
                 3. Set Local Timeout
                 4. Set Remote Timeout

                 5. Return to previous menu

                Select a menu item ...
```

The addp protocol can be enabled or disabled by selecting the first menu option. Both timeouts should be entered as seconds with the Local Timeout being less than the Remote Timeout.

addp Trace mode

The script lists the allowed trace mode settings. It is recommended that the trace mode be set to *full*.



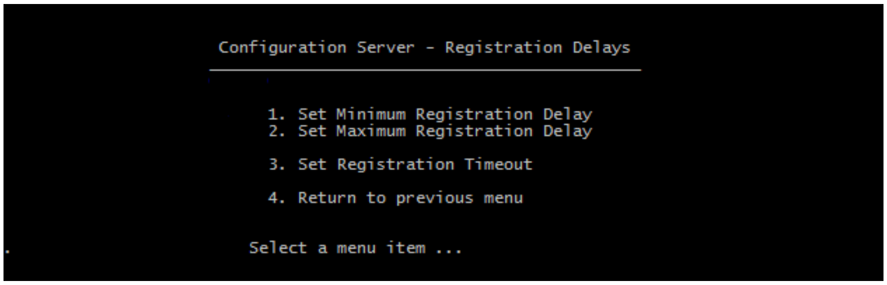
```
==========================================================================
              Configuration Server addp Trace Mode
==========================================================================
      If the Adapter's connection to the Configuration Server has addp enabled,
      the recommended Trace Mode setting is "full" which corresponds to "Trace
      on Both Sides".
==========================================================================

              Config Server addp Trace Mode

               1. off
               2. local
               3. remote
               4. full

                  Select a value ...  4
```

Enable/Disable TLS

TLS (Transaction Layer Security) can be set on the Configuration Server connection but the majority of the configuration work will be on the server side. The extent of the changes on the Adapter side is to enable TLS and set the connection to point to the Configuration Server's secure port.

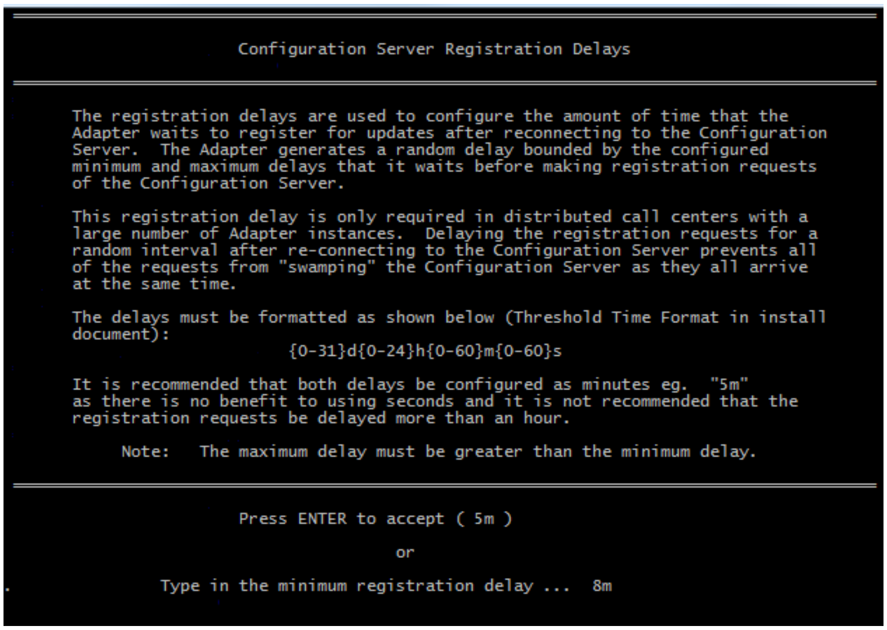
Set Registration delays

This menu section may be necessary in very busy call centers where there are a number of applications connected to the Configuration Server. In that case, the response to requests may be delayed and it may be necessary to delay those requests when the Configuration Server is restarting after a mishap.



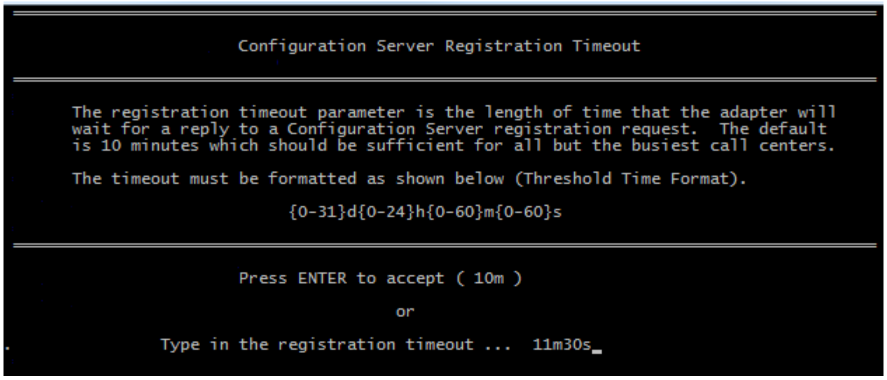
```
                      Configuration Server - Registration Delays
                    ______________________________________________

                            1. Set Minimum Registration Delay
                            2. Set Maximum Registration Delay

                            3. Set Registration Timeout

                            4. Return to previous menu

.                         Select a menu item ...
```

The registration delays are the minimum and maximum bounds of the time that the G*plus* Adapter will wait before making a registration request of the Configuration Server after the connection has been restored. The Adapter will wait a random time between the two limits.



```
                    Configuration Server Registration Delays

      The registration delays are used to configure the amount of time that the
      Adapter waits to register for updates after reconnecting to the Configuration
      Server.  The Adapter generates a random delay bounded by the configured
      minimum and maximum delays that it waits before making registration requests
      of the Configuration Server.

      This registration delay is only required in distributed call centers with a
      large number of Adapter instances.  Delaying the registration requests for a
      random interval after re-connecting to the Configuration Server prevents all
      of the requests from "swamping" the Configuration Server as they all arrive
      at the same time.

      The delays must be formatted as shown below (Threshold Time Format in install
      document):
                            {0-31}d{0-24}h{0-60}m{0-60}s

      It is recommended that both delays be configured as minutes eg.  "5m"
      as there is no benefit to using seconds and it is not recommended that the
      registration requests be delayed more than an hour.

           Note:   The maximum delay must be greater than the minimum delay.

                        Press ENTER to accept ( 5m )

                                       or

.                Type in the minimum registration delay ...  8m
```

As mentioned, the response from the Configuration Server can be significantly delayed in very busy call centers. The registration timeout option sets the amount of time that the Adapter will wait generating a new request.



```
                     Configuration Server Registration Timeout

      The registration timeout parameter is the length of time that the adapter will
      wait for a reply to a Configuration Server registration request.  The default
      is 10 minutes which should be sufficient for all but the busiest call centers.

      The timeout must be formatted as shown below (Threshold Time Format).

                           {0-31}d{0-24}h{0-60}m{0-60}s

                         Press ENTER to accept ( 10m )

                                       or

.              Type in the registration timeout ...  11m30s_
```

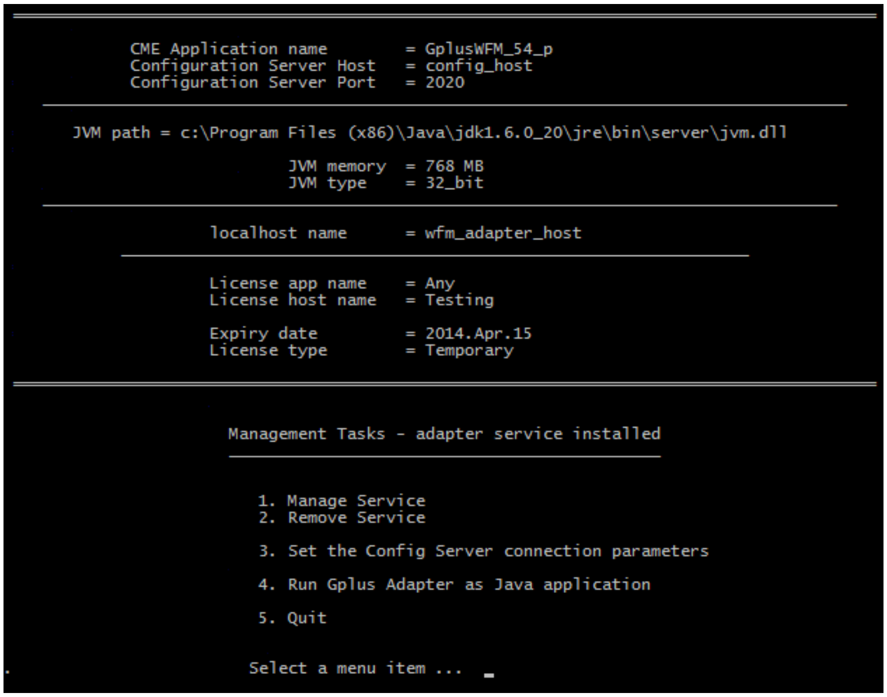
Installing the Adapter as a Service

When the option to install the Adapter as a Service is selected, two actions occur:

1. The appropriate version of the `prunsrv.exe` file is copied to the main Adapter folder from the `wrappers` folder and renamed to the *CME Application name* e.g. `GplusWFM_54_p.exe`.
2. The install mode on the renamed `prunsrv.exe` file is used to install the application as a Windows service with the service name set to the *CME Application name.*

The script will check to see if there is a Service installed with the same name as the *CME Application name*. If that is the case, then the script will display the *Management Tasks* menu.

Management Tasks



```
            CME Application name        = GplusWFM_54_p
            Configuration Server Host   = config_host
            Configuration Server Port   = 2020

   JVM path = c:\Program Files (x86)\Java\jdk1.6.0_20\jre\bin\server\jvm.dll

                      JVM memory  = 768 MB
                      JVM type    = 32_bit

                    localhost name      = wfm_adapter_host

                    License app name    = Any
                    License host name   = Testing

                    Expiry date         = 2014.Apr.15
                    License type        = Temporary

                      Management Tasks - adapter service installed

                         1. Manage Service
                         2. Remove Service

                         3. Set the Config Server connection parameters

                         4. Run Gplus Adapter as Java application

                         5. Quit

                        Select a menu item ...  _
```

The final menu allows for certain tasks to be carried out once the G*plus* Adapter has been installed as a Service.

Manage Service

Selecting the *Manage Service* option invokes the `prunmgr.exe` application that is detailed in Appendix C. Changes can be made to the JVM parameters without having to uninstall the service.

Remove Service

Selecting this option removes the installed service and deletes the renamed `prunsrv.exe` file. The script menu will return to the *Installation Tasks* menu that is detailed above.

Set the Config Server connection parameters

The G*plus* Adapter reads the parameters from the `GPlusWFM.properties` file at start up. It is possible to edit the Configuration Server connection parameters, stop the Adapter, restart the Adapter and have the changes take effect.

Run Gplus Adapter as Java application

This option will start the Adapter as a console application using the same configuration as the Windows Service. It is one way to check that the Adapter configuration is correct if there are problems getting the Service to start.

This is not a viable mode for running the Adapter in a production environment but can be useful for debugging the configuration.