Gplus Adapter for Verint WFM

Installation & Configuration Guide - 7.2

The information contained herein is proprietary and confidential and cannot be disclosed or duplicated without the prior written consent of Genesys Cloud Services, Inc.

Copyright © 2019-2024 Genesys Cloud Services, Inc. All rights reserved.

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 Cloud Services, Inc. cannot assume responsibility for any existing errors. Changes and/or corrections to the information contained in this document may be incorporated in future versions.

Your Responsibility for Your System's Security

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

Trademarks

Genesys and the Genesys logo are registered trademarks of Genesys Cloud Services, Inc. All other company names and logos may be trademarks or registered trademarks of their respective holders. ©2024 Genesys Cloud Services, Inc. All rights reserved.

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 Cloud Services, Inc. http://www.genesys.com/

Document Version: 72gp_icg_verint-wfm_05-2024_v7.2.030.00

Contents

1 About Gplus Adapter

The GPlus WFM Adapter provides an interface between the Genesys CTI Environment and a variety of Workforce Management (WFM) application suites. The adapter supports Nice-IEX, Aspect, Teleopti, and Verint WFM vendors and acts as a translator between Genesys CTI events and the WFM protocols of each vendor.

The adapter provides streaming of Real Time Adherence (RTA) data for monitoring of agent and call status to ensure proper staffing levels and agent schedule compliance. The adapter also generates historical reports to be transferred over FTP to the WFM server.

1.1 Intended Audience

This document is primarily intended for system administrators or other individuals who will be installing the Gplus 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 7.6 - 8.x
  - Configuration Manager
  - Universal Routing 7.0 - 8.x

1.2 Historical Reports

The Verint 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.

Note: The present version of the Adapter does not support the backlog metric for outbound Campaign calls.

1.3 Real Time Adherence (RTA)

Verint does not stipulate a standard set of RTA states so the states tabulated below have been selected as the “standard” used by the Gplus 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
offHook	1307 (see *extendedPhoneStates*)
dialing	1308 (see *extendedPhoneStates*)
ringing	1309 (see *extendedPhoneStates*)
emailInbound	1400
emailReply	1401
emailOutbound	1402
chatInbound	1500
chatInternal	1501
afterChatWork	1502
imInbound	1503
imInternal	1504
afterImWork	1505
outboundPreview	1600
outboundCampaignCall	1601
interactionInbound	1700
interactionInbound	1701
afterInteractionWork	1702

RTA State Priority

When multiple RTA states are present for an agent, the following priorities are applied to determine the most relevant state to appear in the RTA feed. The larger the number, the higher the priority.

For example:

Agent is logged in and ready on voice, email, and chat.

- ready (voice) – priority 60
- ready (email) – priority 60
- ready (chat) – priority 60

Agent is reading a routed email

- emailInbound - priority 150

Agent gets a routed inbound call

- callInbound - priority 400

The agent now has five (5) simultaneously active RTA states, but the agent is reported as CallInbound due to having the highest priority.

RTA State	Priority
afterCallWork	100
afterChatWork	80
afterEmailWork	90
afterIMWork	80
afterInteractionWork	80
callConference	350 (takes precedence over CallConsult)
callConsult	350 when during an inbound or outbound call 200 when during composing collaboration email reply 150 when during reading collaboration email
callInbound	400
callInternal	350
callOnHold	300 - routed call 250 - non-routed call
callOutbound	400 - outboundRoutedCall 350 - outboundNonRoutedCall
chatInbound	130
chatInternal	120
dialing	116
directCallInbound	350
dndOn	450
emailInbound	150
emailOutbound	140

RTA State	Priority
emailReply	200
imInbound	130
imInternal	120
interactionInbound	200
interactionInternal	140
notReady	50
offHook	113
outboundCampaignCall	400
outboundPreview	110
ready	60
ringing	117

2 Installation and Setup

2.1 Requirements

Before proceeding with the installation of the Gplus Adapter, ensure that your system meets the following requirements:

1. **Operating System**: The Gplus Adapter can be installed on both Windows and Unix-like operating systems. Ensure that your system is running a compatible version of Windows or Linux distribution.
   - Microsoft - Windows Server 2012 and higher (64 bit)
   - Linux – Red Hat 6.x – 8.x (64 bit)
2. **User Permissions**: The user performing the installation must have Administrator privileges or root access. This is essential for ensuring proper installation of the adapter and setting up the adapter as a service.
3. **Java Version**: Java 17 is recommended (a minimum of Java 11 is required) to run this version of the Gplus Adapter for optimal performance and compatibility. Make sure to install the appropriate Java version before proceeding with the installation of the adapter.

Once you have verified that your system meets these requirements, you may proceed with the installation and setup process.

2.2 Directory Structure

The application installation package contains all the required components except for the Java Runtime Environment (JRE). Installation of the application is a matter of unzipping the distribution files into a suitable directory. Once you have unzipped the package, you will see the following directories:

Directory	Description
IP	Contains the application files and directories required for running the Gplus Adapter.
templates	Contains the application template file that is required to import the Gplus Adapter application definition into the Configuration Server database.

The directory structure of the IP directory is described below:

Directory	Description
bsd	This is the destination directory for Billing Data Server reports to be written, if enabled. The destination can be configured to a different directory with the `billing/directory` option.
bdsBackup	This is the backup directory for old Billing Data Server reports that have been transferred to an S3 bucket. The backup directory can be configured with the `billing/backupDirectory` option.
config	The Gplus Adapter configuration files, including startup options, service configuration and logging. This directory is also where the GPlus Adapter `license.txt` file will need to be placed.
ftp	The ftp directory contains all of the generated WFM reports that will be transferred to the WFM system over FTP. There are two sub directories: • reports – The destination directory for the WFM reports written by the Gplus Adapter. This destination is configurable in the `historical.reports/directory` option but it should

Directory	Description
	typically match the `historical.ftp/localSourceDirectory` option. • reportsBackup – The backup directory for WFM reports that have been sent to the WFM server over FTP. This destination is configurable in the `historical.ftp/localBackupDirectory` option.
lib	Contains all of the java dependencies required to run the Gplus Adapter.
logs	This is the destination directory for the application logs and Windows service logs. The destination for application logs can be configured in the `IP/config/log4j2.xml` file, see *Apache Logging Services*. The service logs can be configured by editing the service configuration through the install script, see *Appendix E*
recovery	This is the destination directory for the Gplus Adapter recovery logs. Recovery logs record every event that the Gplus Adapter receives from the Genesys servers. They serve to recover lost state after a restart and also as a tool for support. The recovery directory can be configured using the `--recoveryPath` startup option. Java heap dumps and error files generated by runtime errors will also be sent to the recovery directory and can be directed elsewhere by modifying the `JVM -XX:HeapDumpPath` and `-XX:ErrorFile` options in `startup.sh/startup.bat` or in the service configuration.
res	The res directory contains all of the report templates used to format WFM reports.
scripts	The scripts directory contains useful scripts for configuring and starting the adapter. There are sub-directories for Windows and Unix-like operating systems.
wrappers	The wrappers directory contains executables related to Apache Procrun, the Java service wrapper used to install the adapter as a Windows service.

Configuration Files

The config directory contains the configuration files required to install and run the adapter. In a fresh installation, you will see:

Filename	Description
`GPlusWFM.properties`	An example `GPlusWFM.properties` file that contains startup options for running the adapter.
`jvm.properties`	Defines parameters for installing the Gplus Adapter as a Windows service.
`license.txt`	A temporary `license.txt` file. The temporary license is set to expire 60 days after the release date. When you purchase a Gplus Adapter license, you will need to replace this file with the new one.
`log4j2.xml`	An example `log4j2.xml` file, see *Apache Logging Services*
`product.properties`	This file lists the WFM vendor, the PSDK version, as well as the version number for the specific release of Gplus Adapter that has been installed.

Windows Scripts

The `scripts/windows subdirectory` contains startup and installation scripts for configuring and starting the adapter and installing it as a service. The subroutines subdirectory contains scripts used by the windows_install.bat script and should not be modified.

Filename	Description
startup.bat	Script for starting the Gplus Adapter in Windows
windows_install.bat	The Windows installation script is an interactive menu-based console application that guides the user through the configuration of the adapter and installing it as a service. It allows setting the Configuration Server startup parameters that will be stored in IP/config/GPlusWFM.properties. See *Appendix E* for more details.

Unix Scripts

The scripts/unix_bash subdirectory contains startup and utility scripts for configuring and starting the adapter. Unlike for Windows, there is no installation script to aid in configuration. Administrators will need to configure the GPlusWFM.properties file manually before running the adapter. Refer *Configuration* for more information.

Filename	Description
encryptPassword.sh	A utility script for encrypting a password for the Genesys Configuration Server. This is only required if running as a ThirdPartyApplication which is not the recommended method. If running as a ThirdPartyApplication, encryptPassword.sh will encrypt the password and store it in a file named "enc" that needs to be stored in the IP/config directory.
serverHostName.sh	A utility script that will identify the hostname of the host machine. The hostname is used to validate the Hostname property in the license.txt file and it must match exactly.
startup.sh	Script for starting the Gplus Adpater in Unix-like environments.

2.3 Adding an Adapter Instance to the Configuration Server

The Adapter requires an application to be created in the Configuration Server. This is a manual process that involves importing the templates/GPlus_Adapter_Verint_72.apd template file to the Genesys Configuration Manager. Once imported, the template can be used in the creation of the GPlus Application.

Note: The Adapter does not support installation with the Genesys Administrator.

Importing the Application Template

The templates folder contains the GPlus_Adapter_Verint_72.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_72 is one possibility, but alternatives are also acceptable as long as the template name is unique.

Creating the Application in the Configuration Layer

An 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 `Gplus Adapter` template.
4. Once the template is selected, an `Application Properties` window will appear. Input a unique name for the Adapter instance.
5. Click on the `Server Info` tab.
6. Click on the button next to the Host field selector.
7. Select the host on which you will be hosting the adapter and hit OK.
8. Click on the `Start Info` tab.
9. Enter values for the Working Directory and Command Line fields. If you are configuring the adapter for SCS integration, input appropriate values. See *Solutions Control Server Integration*
10. Click on the `Connections` tab.
11. Add connections for the SIP Servers and Interaction Servers that will serve the adapter.
12. Add a connection to the Message Server if desired.
13. Click on the `Options` tab.
14. Modify the application options to suit your needs. Refer *Application Options*
15. Click on the `Annex` tab.
16. If configuring the adapter for SCS integration and you want to run the adapter as a Windows or Linux service, see *Start/Stop a Service with SCS*
17. Hit OK to complete the creation of the application.

Once your application has been created, make note of the application name and application type. These values will be required to configure the adapter, see *Configuration*

2.4 Configuration

Configuring the adapter can be done through the command line or through environment variables, but the most convenient way to quickly configure the adapter is by using the `IP/-config/GPlusWFM.properties` file. The `GPlusWFM.properties` file contains a list of startup options for the adapter.

Here is an example `GPlusWFM.properties` file:

```
# Configuration Server Connection Settings
appName=GPlus_Adapter_{vendor}_72.apt
cmeHost=host1.example.com
cmePort=2020
cmeAppType=ThirdPartyServer
```

```
configTimeout=10m
registrationMinDelay=5m
registrationMaxDelay=15m
cmeBackupHost=hos2.example.com
cmeBackupPort=2020
#cmeUser=
#cmePass=
# ADDP Configuration Settings
addpEnabled=true
#addpLocalTimeout=30
#addpRemoteTimeout=60
#addpTraceMode=None
# HTTP Server Settings
httpServerEnabled = true
httpServerPort = 9765
# IP Settings
#ip-version=
#enable-ipv6=
# TLS Configuration Settings
tls=true
provider=MSCAPI
#trusted-ca=
#trusted-ca-pass=
tls-mutual=true
certificate=f55db0efb800383f488b107a9c0721ce04547a11
#certificate-pass=
#certificate-key=
#certificate-jks-pass=
#tls-crl=
#tls-target-name-check=
#hostname=
# LCA Configuration Settings
#lcaPort=
# Recovery Directory Override
#recoveryPath=
```

The full list of startup options that are available are described in the table below:

Option	Environment Variable	Description	Valid Values
addpEnabled	GPLUSWFM_ADDP_ENABLED	Enable the Advanced Disconnect Detection Protocol (ADDP). ADDP helps detect a connection failure on both the client and the server side	true or false
addpLocalTimeout	GPLUSWFM_ADDP_LOCAL_TIMEOUT	The local timeout for ADDP in seconds	Number
addpRemoteTimeout	GPLUSWFM_ADDP_REMOTE_TIMEOUT	The remote timeout for ADDP in seconds	Number
addpTraceMode	GPLUSWFM_ADDP_TRACE_MODE	Whether ADDP trace logging is enabled on the client, server, both or neither	None, Local, Remote, Both
appName	GPLUSWFM_APP_NAME	The CME application name you defined in. See *Creating the Application in the Configuration Layer*	String
certificate	GPLUSWFM_CERTIFICATE	The TLS certificate specified either as a thumbprint, PEM file or JKS file	String
certificate-jks-pass	GPLUSWFM_KEY_STORE_PASSWORDSTORE_PASSWORD	The JKS password for the certificate when using a JKS provider	String
certificate-key	GPLUSWFM_CERTIFICATE_KEY	The private key for the certificate if using a PEM provider	String
certificate-pass	GPLUSWFM_CERTIFICATE_PASSWORD	The password for the certificate entry in the key store when the provider is set to JKS	String
cmeAppType	GPLUSWFM_CME_APP_TYPE	ThirdPartyServer or ThirdPartyApp	ThirdPartyServer ThirdPartyApp
cmeBackupHost	GPLUSWFM_CME_BACKUP_HOST	The hostname of the backup Configuration Server	Hostname
cmeBackupPort	GPLUSWFM_CME_BACKUP_PORT	The port of the backup Configuration Server	Port number
cmeHost	GPLUSWFM_CME_HOST	The host of the primary Configuration Server	Hostname
cmePass	GPLUSWFM_CME_PASS	The password for the Configuration Server if running as a ThirdPartyApp	String
cmePort	GPLUSWFM_CME_PORT	The port of the primary Configuration Server	Port number
cmeUser	GPLUSWFM_CME_USER	The user for logging into Configuration Server when running as a ThirdPartyApp	String
configTimeout	GPLUSWFM_CONFIG_TIMEOUT	The configuration server timeout for registration requests	Time. See *Threshold Time Format*
enable-ipv6	GPLUSWFM_ENABLE_I_PV6	Enable IPv6	1 or 0
httpServerEnabled	GPLUSWFM_HTTP_SERVER_ENABLED	Enable the HTTP server to create an http endpoint for Prometheus metric scraping	true or false
httpServerPort	GPLUSWFM_HTTP_SERVER_PORT	Port for the HTTP server	Port number

Option	Environment Variable	Description	Valid Values
hostname	GPLUSWFM_HOSTNAME	Hostname to verify against the server certificate when using TLS	Hostname
ip-version	GPLUSWFM_IP_VERSION	IP version preference. The first number is the preferred version.	6,4 or 4,6
lcaPort	GPLUSWFM_LCA_PORT	The port of the Local Control Agent (LCA) if enabled by application/lcaEnabled option	Port number
provider	GPLUSWFM_PROVIDER	The TLS provider to use	MSCAPI PEM JKS
recoveryPath	GPLUSWFM_RECOVERY_PATH	Directory for storing recovery logs	Directory
registrationMaxDelay	GPLUSWFM_REGISTRATION_MAX_DELAY	Maximum registration delay	Time. See *Threshold Time Format*
registrationMinDelay	GPLUSWFM_REGISTRATION_MIN_DELAY	Minimum registration delay	Time. See *Threshold Time Format*
tls	GPLUSWFM_TLS	Enable TLS	true or false
tls-crl	GPLUSWFM_CERTIFICATE_REVOCATION_LIST	TLS certificate revocation list in PEM format	PEM file
tls-mutual	GPLUSWFM_MUTUAL_TLS	Enable mutual TLS	true or false
tls-target-name-check	GPLUSWFM_TLS_TARGET_NAME	If set to “host”, enabled TLS hostname verification	host
trusted-ca	GPLUSWFM_TRUSTED_CA	Trusted certificate authority filename if using PEM or JKS providers	PEM or JKS file
trusted-ca-pass	GPLUSWFM_TRUST_STORE_PASSWORD	The JKS password for the trusted certificate authority when using a JKS provider	String

2.5 Configuration Server Connection

Former versions of the Gplus 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.

Note: Avoid defining the same connection parameter in multiple locations. If a parameter is defined both in a startup script and properties file, the value in the properties file will be ignored.

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 Gplus 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 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.

Note: When TLS is enabled, the `cmePort` parameter should specify the secure port on the Configuration Server. The secure port on the Configuration Server is not necessarily the default port. See *Enabling Transport Layer Security (TLS)*.

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 Gplus 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=Both
```

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 `Both` 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)

tls

Enables the Adapter to use Transport Layer Security (TLS) when connecting to the Configuration Server.

```
tls=<true or false>
```

mutual-tls

Dictates whether connections require a simple or mutual TLS configuration.

```
mutual-tls=<true or false>
```

provider

Identifies the security provider to use. There are three providers available:

```
provider=<MSCAPI or JKS or PEM>
```

cmePort

Specifies the secure port on the Configuration Server when `tls` is enabled.

```
cmePort=<secure port>
```

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.

Setting up the Configuration Server for TLS

The Configuration Server must be properly set up on the server side. Ensure the server certificates and a secure port have been configured. If mutual TLS is enabled, ensure that the Adapter's client certificate can be authenticated.

See Securing Connections Using TLS for how to set up TLS on the Configuration Server.

MSCAPI

Utilizes Microsoft Cryptography API (MS-CAPI) to access the Windows stored certifications.

Parameters:

Note: Install certificates in the Local Computer or Current User and the CA's certificate into Trusted Root Certification Authorities.

```
certificate=<certificate thumbprint, spaces are optional>
```

Example of **simple TLS** using MSCAPI:

```
tls=true
provider=MSCAPI
```

Example of **mutual TLS** using MSCAPI:

```
tls=true
mutual-tls=true
provider=MSCAPI
certificate=49 cc 6a 1b 18 17 38 06 e2 2e 2d f7 40 8d e0 78 71 8c eb 05
```

JKS

Retrieve certificates stored in Java Key Store (JKS).

Parameters:

```
certificate=<path to JKS file>
certificate-jks-pass=<key store password>
certificate-pass=<key/certificate entry password>
trusted-ca=<path to JKS file with the CA's certificate>
trusted-ca-pass=<key store password>
```

Example of **simple TLS** using JKS:

```
tls=true
provider=JKS
trusted-ca=certs\ca.jks
trusted-ca-pass=password987654
```

Example of **mutual TLS** using JKS:

```
tls=true
mutual-tls=true
provider=JKS
certificate=certs\client.jks
certificate-jks-pass=password1234568
certificate-pass=passwordabcdefgh
trusted-ca=certs\ca.jks
trusted-ca-pass=password987654
```

PEM

Utilizes Privacy Enhanced Mail (PEM) files.

Parameters:

```
certificate=<path to PEM file with the certificate>
certificate-key=<path to PEM file with the private key>
trusted-ca=<path to PEM file with the CA's certificate>
```

Example of **simple TLS** using PEM:

```
tls=true
provider=PEM
trusted-ca=certs\ca.pem
```

Example of **mutual TLS** using PEM:

```
tls=true
mutual-tls=true
provider=PEM
certificate=certs\client-cert.pem
certificate-key=certs\client-key.pem
trusted-ca=certs\ca.pem
```

Additional TLS parameters

tls-crl

Specifies a Certificate Revocation List (CRL) file to check during the certificate validation process. This file must be in PEM format.

```
tls-crl=<path to PEM file with the CRL>
```

tls-target-name-check

Enables comparison of certificate subject fields against an expected host name. It checks for matches on the certificate Subject Alternative Name (SAN), if present, or subject Common Name (CN). The `hostname` parameter specifies the expected host name and only supports DNS or IP address values.

```
tls-target-name-check=host
hostname=<expected host name>
```

Note: If the `tls-target-name-check` parameter is set to `host` but the `hostname` parameter is not set, matches are attempted against the `cmeHost` parameter.

IPv6

enable-ipv6

Controls IPv6 support when connecting to the Configuration Server. IPv6 is disabled by default.

```
enable-ipv6 =<0 or 1>
```

ip-version

Defines the order of connection attempts made to the Configuration Server's IPv6 and IPv4 addresses. It has no effect if IPv6 support is disabled. Do not include spaces between the comma-separated numbers.

```
ip-version=<4,6 or 6,4>
```

2.6 License File

A temporary license file (`license.txt`) is included in the `IP/config` folder of the Gplus Adapter installation. This license will allow the adapter to run on any server for a limited number 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 specific to a particular hostname and so moving the adapter to another server will 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

The hostname can be determined by running the `serverHostName.sh` script for Unix-like operating systems. For Windows, the hostname will be displayed along with the other information in the `windows_install.bat` script . See *Appendix: Windows Installation Script*

2.7 Running the GPlus Adapter

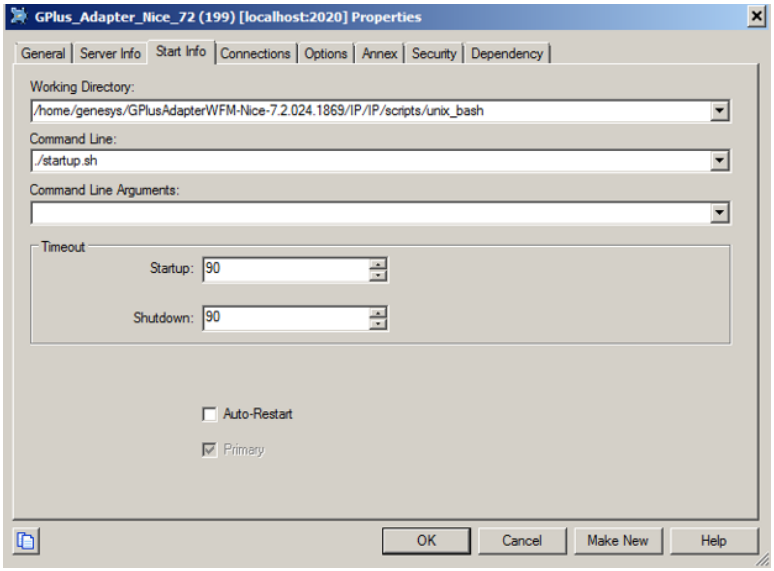
The Gplus Adapter can be run using the startup scripts in the `IP/scripts` directory. For Windows installations, run `windows/startup.bat` and for Unix-like installations run `unix_bash/startup.sh`. All scripts require the following 3 environment variables to be set:

Environment Variable	Description
GPLUSWFM_HOME	The absolute path of the IP directory containing `bootstrap.jar`.
GPLUSWFM_JVM_MEM	The amount of memory to allocate to the Java virtual machine, given in a number of megabytes
GPLUSWFM_JAVA_HOME	The absolute path of the Java 17 or Java 11 home directory

2.8 Solution Control Server Integration

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

The Gplus Adapter can be started and stopped using the Solution Control Interface. This can be done by setting the appropriate values for the Working Directory, Command Line and Command Line Arguments fields in the `Start Info` tab of the application that was created previously. See *Creating the Application in the Configuration Layer*

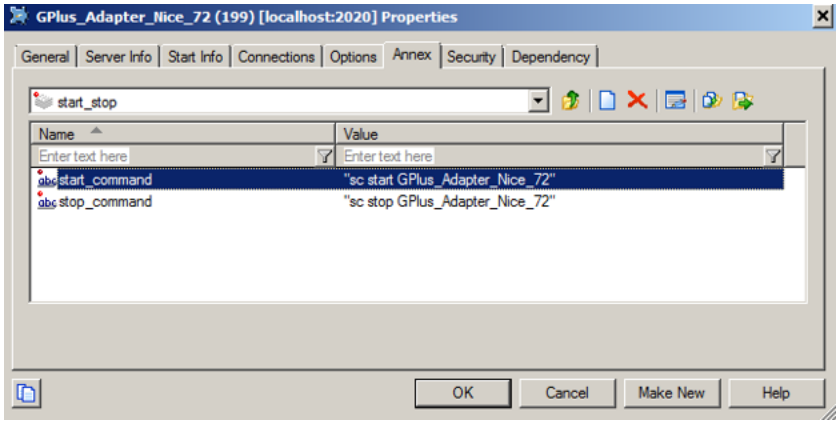


Start/Stop a Service with SCS

If you have configured the Gplus Adapter to run as a Windows or Linux service, you will need to perform additional steps to allow SCS to start and stop the service.

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.

To configure the adapter to start and stop the service with SCS, you will need to create a new annex section named “start_stop”. In this new section, create a key called “`start_command`” and set the value to the command that starts the service in your environment. Also create a key called “`stop_command`” and set the value to the command that stops the service. If running windows, you would use the `sc command` line utility to start and stop the service.



Alternatively, if you have configured the adapter to run as a Linux service you would input the corresponding commands to start and stop the service.

3 Application Options

Each of the Gplus 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.

3.1 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 Gplus 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 Gplus 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.

recoveryLogFailure

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 recovery log files to write, usually due to a full drive.

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 Gplus 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 Gplus Adapter's RTA connection to the WFM server is 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 Gplus 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.

3.2 application Section

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

acwStrategy

Type: Optional

Default Value: `lastRouted`

Valid Values: `lastRouted, pendingOnly`

Dependencies: None

This option defines how the Adapter handles AfterCallWork (ACW) sessions that were not started within a routed interaction. These ACW sessions can either be associated with the last routed interaction that the agent handled (`lastRouted`) or be treated as `Not Ready` (pendingOnly).

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 Gplus 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, chat and other Open Media implementations, only `personUserName` or `personEmployeeId` option values should be used.

annexSection

Note: Contact Professional Services prior to configuring this option.

Type: Optional

Default Value: `GPlus_WFM`

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 Gplus specific options added to a `Place Group`, `VQ` or `Skill` object in the Configuration Server. More than one `Annex` Section containing a site name can be configured for a particular `Place Group`.

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 Gplus 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 Gplus 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 an agent can 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 `Target` in the Gplus 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.

eventLoggingMode

Type: Optional

Default Value: `minimum`

Valid Values: `minimum`, `full`

Dependencies: None

When added and set to the default value of minimum, only the necessary event attributes required for reporting by the Gplus Adapter for WFM are logged. This prevents sensitive PII data from appearing in recovery logs. A value of full provides all event attributes in recovery logs.

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 Gplus 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.

lcaInstalled

Type: Optional

Default Value: `true`

Valid Values: `truefalse`

Dependencies: None

This Option determines whether the Adapter will attempt to connect to an LCA instance at startup. It was added to eliminate the Adapter connection attempts and subsequent failures when Genesys Local Control Agent (LCA) that has not been installed.

logoutOnEmptyMediaTypes

Type: Optional

Default Value: `true`

Valid Values: `truefalse`

Dependencies: None

When set to true, the Adapter will treat an agent who logs out of all media types but remains logged into the Interaction Server as logged out.

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/log4j2.xml` file. This option defines the logging level that the application will use after the configuration information is read from the Configuration Server.

Appendix: Log Files discusses different log file examples based on the configured logThreshold.

pendingLogout

Type: Optional

Default Value: `false`

Valid Values: `true, false`

Dependencies: None

This Option determines how an `EventAgentLogout` will be processed: immediately (`false`) or pending (`true`) until the conclusion of all active interactions that the Agent is party to. When the logout is processed immediately, all reporting for ongoing calls is terminated at that point.

recoveryTimeout

Type: Optional

Default Value: `0 (no timeout set)`

Valid Values: `any timespan either denoted with units (1d, 5h, 5m, 3600s, etc.) or without units, in which case the entry is interpreted as milliseconds.`

Dependencies: None

If the recovery log playback has not been completed by this timeout, processing ends and start up continues without completing the recovery log processing.

shortAbandonEnabled

Note: Contact Professional Services prior to enabling this option.

Type: Optional

Default Value: `false`

Valid Values: `true`, `false`

Dependencies: None

This option enables the removal of abandoned interactions from the *Abandoned* count in the *Contact Statistics* report if the interactions were abandoned prior to a configured time threshold. It should be noted that the interactions are just removed from that one statistic and will continue to be included in the abandoned counts used to calculate the service levels. See *serviceLevelFormula*.

As only abandoned interactions for inbound voice, chat or im interactions are reported, this option applies to those media. It is possible to set a `defaultShortAbandonThreshold` in the *voice*, *chat*, and *im* Option sections and that Option applies to all interactions of the media type. Further configuration can be carried out by adding the `shortAbandonThreshold` Option to the Annex data of a Virtual Queue. See *Adding Annex Data for Virtual Queues and Skills*.

siteBased

Note: Contact Professional Services prior to enabling this option.

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 Gplus 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 report.

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.

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 Gplus 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: See *Appendix: Time Zone ID List*.

Dependencies: None.

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

useLocalTime

Note: Contact Professional Services prior to configuring this option.

Type: Optional

Default Value: `auto`

Valid Values: `auto`, `true`, `false`

Dependencies: None

The `useLocalTime` option allows the Gplus 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`.

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.

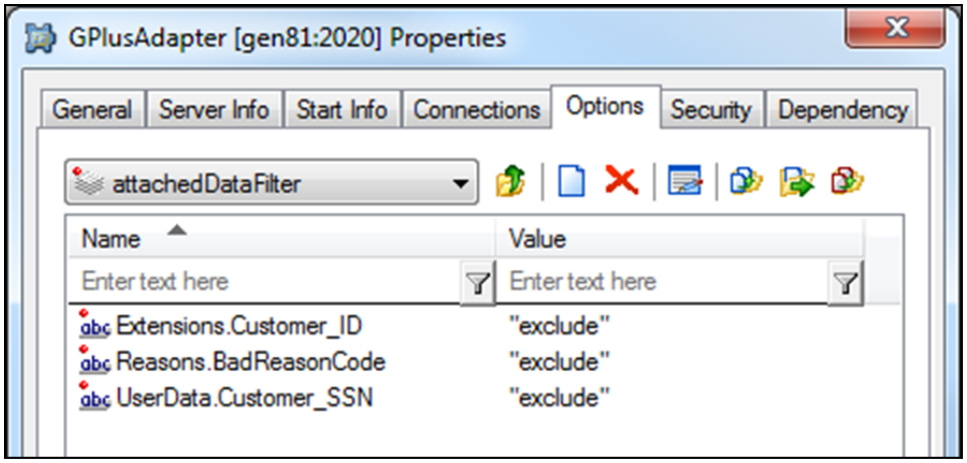
3.3 attachedDataFilter Section

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 be 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.



3.4 billing Section

This Option Section contains the configuration options for submitting usage reports to *Genesys Billing Database Server* (*BDS*). Most of the configuration options below deal with where the Adapter will store the generated files. Placing the files on the local host or uploading them to an *Amazon Web Services* (*AWS*) *Simple Storage Service* (*S3*) storage bucket are both supported.

backupDirectory

Type: Optional

Default Value: `bdsBackup`

Valid Values: Any valid folder on the Gplus Adapter server

Dependencies: `enabled, destination, retentionTime`

This Option specifies the folder where the *BDS* report files will be retained when those files have been successfully sent to the *AWS S3* storage bucket. The contents of the folder are managed by the Adapter. Files are deleted when they have reached the threshold specified in the *retentionTime* option.

destination

Type: Mandatory

Default Value: `local`

Valid Values: `local, S3`

Dependencies: `enabled`

This Option specifies whether *BDS* reports are retained on the local server or transferred to an *AWS S3* bucket. If *S3* is specified as the destination, the server or the application must be configured properly with credentials to access the bucket. The AWS credentials webpage lists the details for several different authentication methods. See Working with AWS Credentials.

directory

Type: Optional

Default Value: `bds`

Valid Values: Any valid folder on the Gplus Adapter server

Dependencies: `enabled, destination, retentionTime`

This Option specifies the local folder where the *BDS* reports will be kept for the *BDS* application to access them. The contents of the folder are managed by the Adapter. The files are deleted when they have reached the threshold specified in the *retentionTime* option.

enabled

Type: Mandatory

Default Value: `false`

Valid Values: `true, false`

Dependencies: None

This option enables the Adapter to generate the daily *BDS* usage reports used by *Genesys* for their subscription billing model. The reports are generated at midnight UTC and contain the activity for the previous 24 hours.

fileNamePrefix

Type: Optional

Default Value: `BDS_Report.`

Valid Values: any file name prefix followed by a period

Dependencies: None

This option allows the user to change the prefix for the billing report file name. For example, `GplusWFM_TenantID.`.

region

Type: Optional

Default Value: *Not set*

Valid Values: An *AWS* S3 region identifier

Dependencies: `destination`

This Option allows users to specify the region where their AWS S3 storage bucket resides. See AWS Service Endpoints.

retentionTime

Type: Optional

Default Value: `7d`

Valid Values: A positive integer with a '`d`' suffix.

Dependencies: `directory`, `backupDirectory`

This Option specifies the duration in days that reports are kept inside the local *BDS* report directory or the backup directory that retains copies of the files transferred to the *S3* storage bucket.

s3Bucket

Type: Optional

Default Value: *Not set*

Valid Values: A globally unique *S3* bucket name

Dependencies: `destination, region`

This Option specifies the AWS S3 storage bucket in the configured *region* that will serve as the destination for the generated BDS reports. *Appendix: AWS Credentials – S3 Storage Bucket* discusses AWS credentials used for accessing the storage bucket.

s3Subdirectory

Type: Optional

Default Value: *Not set* - root of bucket

Valid Values: Path to an existing folder in the *S3* bucket

Dependencies: `destination, s3Bucket`

This Option specifies the path to the folder in the *AWS S3* storage bucket that will contain the forwarded *BDS* reports.

testing

Type: Testing only

Default Value: `false`

Valid Values: `true, false`

Dependencies: None

This Option enables the generation of test reports on the interval specified in the *testingInterval* option.

testInterval

Type: Testing only

Default Value: `5m`

Valid Values: A positive integer with an `'m'` suffix

Dependencies: None

This Option specifies the interval duration in minutes for the generated test reports.

3.5 callback Section

This Option section provides the configuration options for support of the Genesys *Callback 8.5* functionality. The typical configuration should only entail enabling the Option section as the defaults will be correct for the majority of installations. When this Option Section is enabled, the original inbound call and the subsequent successful callback will be counted as a single call.

abandonDelay

Type: Optional

Default Value: `0`

Valid Values: See *Threshold Time Format*

Dependencies: `callback/enabled, callback/callbackType`

Tracking unsuccessful callbacks in "agent first" mode of the *Premise* version of *Genesys Callback* requires the Adapter to delay processing those callbacks when determining when the retry limit has been reached. That delay is configurable with this Option. It should be greater than the combination of the configured delay between callback retries and the time threshold that an agent has to refuse a callback retry.

Note: Setting this Option requires an in-depth level knowledge of the Callback implementation details and should be undertaken with the collaboration of qualified personnel.

agentAddedDeterminer

Type: Optional

Default Value: `UserData._CB_N_AGENT_ADDED_TO_IXN`

Valid Values: See *Event Attribute Determiners*

Dependencies: callback/enabled

This Option identifies the event attribute that identifies whether an agent has been added to the callback. This option is optional and is only used when using an environment that requires it. Any returned non-zero value, including null will be treated as true. If not set, this will also default to true.

callbackType

Type: Optional

Default Value: `cloud`

Valid Values: `cloud` or `premise`

Dependencies: `callback/enabled`

Genesys Callback comes in two versions: *Cloud* and *Premise* which have been implemented differently. This Option allows the *Callback* version to be set for the implementation being used in the Genesys environment.

cancelledDisposition

Type: Optional

Default Value: `COMPLETED.FAILED`

Valid Values: See *Event Attribute Determiners*

Dependencies: `callback/enabled, callback/connectedDeterminer`

This Option identifies the value used to indicate an abandoned contact in the disposition. If the value returned from the `dispositionDeterminer` equals this value, the interaction is considered to be abandoned.

connectedDeterminer

Type: Optional

Default Value: `eventAttribute: UserData._CB_T_CUSTOMER_CONNECTED`

Valid Values: See *Event Attribute Determiners*

Dependencies: `callback/enabled, callback/connectedValue`

This Option identifies the `EventDiverted` attribute that contains the callback connected status or connection timestamp.

dispositionDeterminer

Type: Optional

Default Value: `eventAttribute: UserData._CB_DISPOSITION`

Valid Values: See *Event Attribute Determiners*

Dependencies: `callback/enabled, callback/scheduledDisposition`

This Option identifies the event attribute that contains the disposition type for the associated call. Presently, the Adapter only checks this attribute when the original inbound call is *Abandoned* after the caller has elected to be called back rather than waiting for an available agent.

enabled

Type: Mandatory

Default Value: `false`

Valid Values: `true` or `false`

Dependencies:

This Option enables the Adapter to track *Genesys Callback* interactions where the callbacks are directly associated with the original inbound call and the queue delay is calculated as the period between the inbound call and the successful *callback*.

idDeterminer

Type: Optional

Default Value: `eventAttribute: UserData._CB_SERVICE_ID`

Valid Values: See *Event Attribute Determiners*

Dependencies: `callback/enabled`

This Option identifies the event attribute that contains the unique identifier that links the original inbound customer call with the outbound *callback*(s).

maxDuration

Type: Optional

Default Value: `1d`

Valid Values: *Threshold Time Format*

Dependencies: `callback/enabled`

Examples: `1d2h35m30s, 12h25m`

This Option specifies a time threshold after which an inbound call is counted as Abandoned if there has not been a successful callback.

scheduledDisposition

Type: Optional

Default Value: `SCHEDULED`

Valid Values: Any text string

Dependencies: `callback/enabled, callback/dispositionDeterminer`

This Option specifies value of the event attribute identified by the `dispositionDeterminer` Option that identifies the disposition type as a scheduled callback. This attribute will be set on the *Abandoned* event generated on the initial inbound call when the caller elects to be called back.

3.6 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 Gplus 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 must match the exact value generated by the Gplus Adapter when applying the `callType.determiner` to the 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 Time Format*) and/or a `shortName` can be set in a manner similar to that of the Virtual Queues. See *Adding Annex Data for Virtual Queues and Skills*.

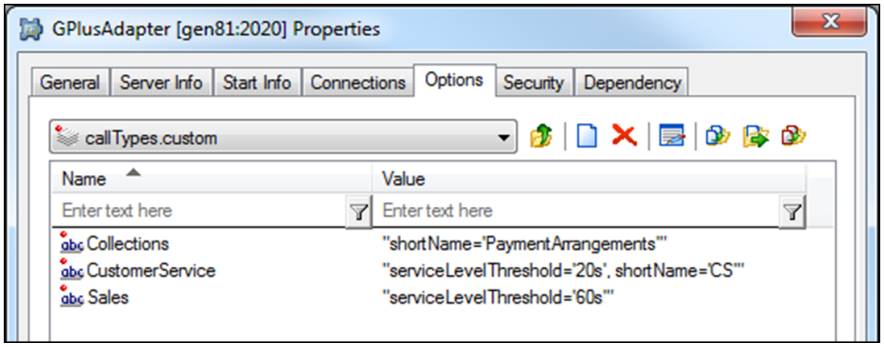
If no `serviceLevelThreshold` is specified, the Gplus Adapter will use the `defaultServiceThreshold` Option from the voice Section of the Adapter Options. If no `shortName` is specified, the Gplus Adapter will report the `Option Name`.

Type: Optional

Default Value: Section is not used

Valid Values: `serviceLevelThreshold` and `shortName`

Dependencies: `application/callType`, `event.properties/callType.determiner`



3.7 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 `AfterChatWork(User_9)`

The text string must be contained in the attribute of a chat `NotReadyForMedia` event specified by the *reason.notReady.determiner*.

defaultAbandonedThreshold

Type: Optional

Default Value: `0s`

Valid Values: *Threshold Time 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 *serviceLevelFormula* 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 *Adding Annex Data for Virtual Queues and Skills* 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: *Threshold Time 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 *Adding Annex Data for Virtual Queues and Skills* 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.

defaultShortAbandonThreshold

Type: Optional

Default Value: *Not set*

Valid Values: *Threshold Time Format*

Dependencies: `application/shortAbandonEnabled`

This option specifies the default time threshold that an abandoned chat must exceed in order to be included in the *Actual Abandons (ABD)* column of the *Forecast* report. This option does not affect the service level calculations.

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 Gplus 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.

It should be noted that the default for this Option is the same as that for voice and email (`UnknownTarget`). If the adapter is monitoring multiple media and the defaults are retained, the interactions that cannot be allocated to a proper target will be combined under `UnknownTarget` in the *Forecast Report*.

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

Note: Contact Professional Services before using this option.

Type: Optional

Default Value: *Not set*

Valid Values: See *Event Attribute Determiners*

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*.

maxInteractionTime

Type: Optional

Default Value: *Not Set*

Valid Values: *Threshold Time Format*

Dependencies: None

While unlikely, it is possible for some interactions to become *stuck* in certain states if the terminating events are not received due to a mishap with an Interaction Server connection. Normally, the stuck interaction records in the Adapter would be terminated as soon as the connection was lost but this response relies on the addp protocol being configured properly for the connection(s) to the Interaction Server(s). See *Additional Configuration*.

This option sets a timeout that defines the maximum amount of time that a chat can remain in a handled or wrap state before the Adapter automatically terminates its associated record in the data tables. If the Option is not set, the stuck records are not terminated until the agent logs out or any agent logs in to the Place associated with the interaction.

If used, this timeout must be set to a value significantly higher than the expected longest chat duration to ensure that an ongoing interaction is not terminated prematurely in the Adapter tables. It should be noted that the timeout also applies to wrap up sessions so the expected duration of the wrap time must be taken into account.

outOfFocus.determiner

Note: The customer is responsible for setting the out-of-focus custom event attribute(s), e.g., using a desktop plugin.

Type: Optional

Default Value: *Not set*

Valid Values: See *Event Attribute Determiners*

Dependencies: `chat/outOfFocus.values`

This option specifies the attribute(s) of a chat Interaction Server Event that are checked to determine if the chat is out-of-focus. Out-of-focus chats do not contribute to historical handle time metrics or RTA states. If an agent enters an ACW state with chats that are all out-of-focus, work time counts towards the last released chat. If no chat was released beforehand, this is reported as unattached ACW.

outOfFocus.values

Type: Optional

Default Value: *Not set*

Valid Values: Comma-separated string

Dependencies: `chat/outOfFocus.determiner`

This option specifies a comma separated list of values that indicate the chat is out-of-focus. These values are checked using the `outOfFocus.determiner`.

reason.notReady.determiner

Note: Contact Professional Services before using this option.

Type: Optional

Default Value: `eventAttribute: ReasonDescription`

Valid Values: See *Event Attribute Determiners*

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*.

reason.ready.determiner

Note: Contact Professional Services before using this option.

Type: Optional

Default Value: *Not set*

Valid Values: See *Event Attribute Determiners*

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 an `EventReadyForMedia` event.

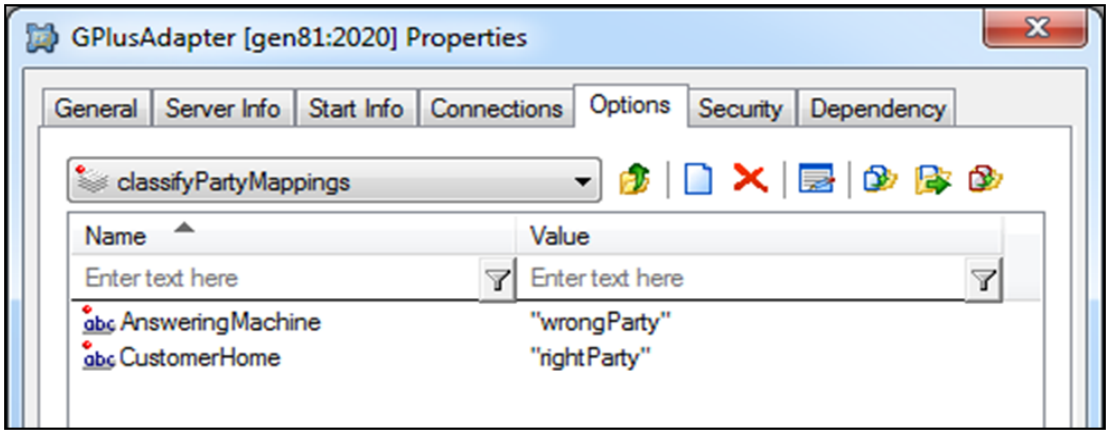
Certain implementations may require that the combination of an `Available` RTA state and the reason code map to a different RTA state and reason text string. This mapping can be configured in the *readyCodeMappings Section*.

3.8 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 *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:

- 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*.



3.9 customReporting 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 `customReporting` options provide support for reporting on Cloud iWD interactions.

Note: For these custom events to propagate to the adapter, the `enableCustomWfmEvents` flag must be enabled in Genesys Cloud iWD. Contact Professional Services before using this option.

customAttributeDeterminer

Type: Optional

Default Value: *Not set*

Valid Values: See *Event Attribute Determiners*

Dependencies: *customAttributeValues*

This option specifies the attribute of an `EventCustomReporting` event and future interaction events, indicating if these events are part of a Cloud iWD interaction. If this option is not set, this section is disabled and any `EventCustomReporting` events received will be ignored.

customAttributeValues

Type: Optional

Default Value: *Not set*

Valid Values: Comma separated strings

Dependencies: *customAttributeDeterminer*

This option specifies a comma separated list of values that indicate that this is part of a Cloud iWD interaction. These values are checked using the above determiner. If this option is not set, this section is disabled and any `EventCustomReporting` events received will be ignored.

customIdDeterminer

Type: Optional

Default Value: *Not set*

Valid Values: See *Event Attribute Determiners*

Dependencies: None

This option specifies the ID of a Cloud iWD interaction as this may differ from the InteractionId(s) in use. If this option is not set, this section is disabled and any `EventCustomReporting` events received will be ignored.

customReportingReclassifyId

Type: Optional

Default Value: 9001

Valid Values: Any positive integer

Dependencies: *customAttributeDeterminer*, *customAttributeValues*, *customIdDeterminer*

This option specifies the value found in the `CustomEventId` attribute of the `EventCustomReporting` event, indicating to end the existing callflow of the Cloud iWD interaction and remove it from the callFlowManager.

customReportingStartId

Type: Optional

Default Value: 9000

Valid Values: Any positive integer

Dependencies: *customAttributeDeterminer*, *customAttributeValues*, *customIdDeterminer*

This option specifies the value found in the `CustomEventId` attribute of the `EventCustomReporting` event, indicating the start of the Cloud iWD interaction.

customReportingStopId

Type: Optional

Default Value: 9002

Valid Values: Any positive integer

Dependencies: *customAttributeDeterminer*, *customAttributeValues*, *customIdDeterminer*

This option specifies the value found in the `CustomEventId` attribute of the `EventCustomReporting` event, indicating the end of the Cloud iWD interaction.

3.10 email Section

defaultServiceThreshold

Type: Optional

Default Value: `0s`

Valid Values: *Threshold Time 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 *Adding Annex Data for Virtual Queues and Skills* 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 Gplus 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.

It should be noted that the default for this Option is the same as that for voice and chat (`UnknownTarget`). If the adapter is monitoring multiple media and the defaults are retained, the interactions that cannot be allocated to a proper target will be combined under `UnknownTarget` in the *Forecast Report*.

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 `EmailACW (User_5)` RTA state.

The text string must be contained in the attribute of an email `NotReadyForMedia` event specified by the *reason.notReady.determiner*.

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

Note: Contact Professional Services before using this option.

Type: Optional

Default Value: *Not set*

Valid Values: See *Event Attribute Determiners*

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*.

maxInteractionTime

Type: Optional

Default Value: *Not Set*

Valid Values: *Threshold Time Format*

Dependencies: None

While unlikely, it is possible for some interactions to become *stuck* in certain states if the terminating events are not received due to a mishap with an Interaction Server connection. Normally, the *stuck* interaction records in the Adapter would be terminated as soon as the connection was lost but this response relies on the addp protocol being configured properly for the connection(s) to the Interaction Server(s). See *Additional Configuration*.

This option sets a timeout that defines the maximum amount of time that an email can remain in a *handled* or *wrap* state before the Adapter automatically terminates its associated record in the data tables. If the Option is not set, the *stuck* records are not terminated until the agent logs out or any agent logs in to the Place associated with the interaction.

If used, this timeout must be set to a value significantly higher than the expected longest email duration to ensure that an ongoing interaction is not terminated prematurely in the Adapter tables. It should be noted that the timeout also applies to wrap sessions so the expected duration of any wrap time must be taken into account.

reason.notReady.determiner

Note: Contact Professional Services before using this option.

Type: Optional

Default Value: `eventAttribute: ReasonDescription`

Valid Values: See *Event Attribute Determiners*

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*.

reason.ready.determiner

Note: Contact Professional Services before using this option.

Type: Optional

Default Value: *Not set*

Valid Values: See *Event Attribute Determiners*

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 an `EventReadyForMedia` event.

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*.

3.11 event.properties Section

callType.determiner

Type: Optional

Default Value: `eventAttribute: ThisQueue`

Valid Values: See *Event Attribute Determiners*

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

Note: Contact Professional Services prior to configuring this option.

Type: Optional

Default Value: `eventAttribute: UserData.GSW_CAMPAIGN_NAME`

Valid Values: See *Event Attribute Determiners*

Dependencies: `outbound/enabled`

This option specifies what part (attribute) of an event 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 Voice Campaigns* for more information.

chat.target.determiner

Type: Optional

Default Value: `eventAttribute: interactionProperties.interactionQueue`

Valid Values: See *Event Attribute Determiners*

Dependencies: `chat/enabled`

This Option identifies the attribute in the queued event to be used as the chat target or queue. It replaces the discontinued `interactionTarget.determiner` that was formerly used for email and chat interactions.

email.target.determiner

Type: Optional

Default Value: `eventAttribute: interactionProperties.interactionQueue`

Valid Values: See *Event Attribute Determiners*

Dependencies: `email/enabled`

This Option identifies the attribute in the queued event to be used as the email target or queue. It replaces `the interactionTarget.determiner` that was formerly used for email and chat interactions.

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.

ignoreRONA_EventAttribute

Note: Contact Professional Services prior to configuring this option.

Type: Optional

Default Value: *Not set*

Valid Values: `UserData.`{RONA identifier field name}

Dependencies: Nortel Switch

This option specifies the attached data field that the Gplus 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.

im.target.determiner

Type: Optional

Default Value: `eventAttribute: interactionProperties.interactionQueue`

Valid Values: See *Event Attribute Determiners*

Dependencies: `im/enabled`

This Option identifies the attribute in the queued event to be used as the im target or queue.

ixn.logout.determiner

Type: Optional

Default Value: `eventAttribute: ReasonDescription`

Valid Values: See *Event Attribute Determiners*

Dependencies: None

This Option identifies the attribute in the Interaction Server logout event to be included as the Reason field in the RTA state message.

reason.logout.determiner

Type: Optional

Default Value: `eventAttribute: Extensions.ReasonCode`

Valid Values: See *Event Attribute Determiners*

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

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

Type: Optional

Default Value: `eventAttribute: Reasons.ReasonCode`

Valid Values: See *Event Attribute Determiners*

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*.

ronaValue

Type: Optional

Default Value: None

Valid Values: `any string`

Dependencies: `ignoreRONA_EventAttribute`

Setting this value determines the valid values for the attribute defined in the *ignoreRONA_EventAttribute* option in order to determine if a call is a RONA scenario. If the `ronaValue` matches the `ignoreRONA_EventAttribute` value, then the call is counted as answered.

serviceDate.determiner

Type: Optional

Default Value: *Not set*

Valid Values: See *Event Attribute Determiners*

Dependencies: None

This option specifies the attribute in a 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"

usePlacedInQueueTime

Type: Optional

Default Value: true

Valid Values: true or false

Dependencies: None

This option specifies whether the queue times used for interaction server interactions comes from the UserData.PlacedInQueueAt attribute (true), or the event time on the queue event (false) for determining queue delay. The offered is always counted when it occurs, regardless of this setting.

3.12 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 Section*, *genesys.placeGroups Section* and *genesys.queueGroups Section*. Filters combined with the Stream configurations (*historical:<streamName> Section* and/or *rta:<streamName> Section*) allow the Gplus Adapter to support multiple WFM 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 options allows two alternatives 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 in *Adding A PlaceGroup to a Site*. The syntax for the group options is as follows:

- agentGroup.<AgentGroupName> : include or exclude
- campaign.<Campaign Name> : include or exclude
- dataGroup.<Data Group Name> : include or exclude
- loginQueue.<ACDQueue DN> : include or exclude
- media.<mediaType> : include or exclude
- placeGroup.<PlaceGroupName> : 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 campaign.<Campaign Name> filter is similar to the targetGroup.<DNGroupName> filter in terms of its effect. An excluded outbound campaign is not reported in the *Outbound Contact Statistics report* but the agents handling those calls will still have their call activity reported in the *Agent Scorecard Metric report*. However, the campaign.<Campaign Name> filter is quite different from the other filters with the exception of the media.<mediaType> filter in that it references a single campaign rather than a defined group of Agents, Places or VQs.

The media.<mediaType> filter allows for interactions with a specific media type to be included or excluded.

3.13 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 Gplus 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 Gplus 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 report*; 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.

3.14 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 Gplus 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 Gplus 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 Gplus 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, *Adding A PlaceGroup to a Site*.

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.

3.15 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 Gplus 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 Gplus 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.

3.16 historical.ftp Section

This section lists the configuration options for transferring historical report files from the server hosting the Gplus 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 *Streams and Filters*, this Section should be set up to transfer the report files associated with the Stream defined in the *historical.reports Section*.

agentDailyDirectory

Type: Optional

Default Value: *Not set*

Valid Values: Any folder path on the remote server relative to the ftp user's default directory

Dependencies: `historical.reports/dailyReportEnabled`

If this option is set, this directory will be used for the *Agent Scoreboard Metric* report. If this is not set, the *remoteDestinationDirectory* Option will be used for this report instead.

backupRetentionDays

Type: Mandatory

Default Value: `30`

Valid Values: Any positive integer

Dependencies: `historical.reports/localBackupDirectory`

This Option specifies the number of days that historical report files created by the Gplus 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.

campaignDirectory

Type: Optional

Default Value: *Not set*

Valid Values: Any folder path on the remote server relative to the ftp user's default directory

Dependencies: None

If this option is set, this directory will be used for the *Campaign Contact Statistics* report. If this is not set, the *remoteDestinationDirectory* Option will be used for this report instead.

chatDirectory

Type: Optional

Default Value: Not set

Valid Values: Any directory

Dependencies: None

If this option is set, this directory will be used for the *Chat Contact Statistics* report. If this is not set, the *remoteDestinationDirectory* Option will be used for this report instead.

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 Gplus Adapter applications that do not need to send report files on a continuous basis but only in the event of a mishap.

enableHostVerification

Type: Optional

Default Value: `true`

Valid Values: `true, false`

Dependencies: knownHostEntry

This option enables SFTP host validation against `known_hosts` file or the *knownHostEntry* Option.

For more information, see *Setting Up Secure File Transfer* and *Appendix: Secure File Transfer Problems*.

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.

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.

knownHostEntry

Type: Optional

Default Value: *Not set*

Valid Values: Any string

Dependencies: `enableHostVerification`

This option holds a value that represents an entry in a `known_hosts` file. When the `enableHostVerification` option is enabled, the SFTP transfer manager will verify host connections against the entry. This option can be used in situations where the `known_hosts` file is inaccessible or cannot be updated. For more information, see *Setting Up Secure File Transfer* and *Appendix: Secure File Transfer Problems*.

localBackupDirectory

Type: Mandatory

Default Value: `ftp/reportsBackup`

Valid Values: Any existing folder on the Gplus Adapter server.

Dependencies: None

This option identifies the directory on the Gplus 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 report location specified in the `historical.reports` Option section and setting the two options to the same folder establishes the association between the ftp client and the historical Stream.

mediaAgentReportDirectory

Type: Optional

Default Value: Not set

Valid Values: Any directory

Dependencies: *verint.mediaAgentReportEnabled*

If this option is set, this directory will be used for the *Media Agent Daily* report. If this is not set, the *remoteDestinationDirectory* Option will be used for this report instead.

password

Type: Optional

Default Value: *Not set*

Valid Values: Any alphanumeric password

Dependencies: `User identified by userName`

This option should be used to replace the *userPassword* Option if there is a requirement that the password text not be displayed as plain text in either *Genesys Administrator* or the *Configuration Manager*. Both applications automatically hide the Option value if the Option name is "password".

This Option will take precedence when both password Options have been set.

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 Gplus 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 WFM 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.

retry

Type: Optional

Default Value: 0

Valid Values: Any non-negative integer

Dependencies: None

This option allows users to define the number of retries used for those environments with ftp server issues that require retries sooner than the next interval. This option is not normally required to be set, as the first failure is most often quite relevant and should not be ignored.

sessionTimeout

Type: Mandatory

Default Value: `120s`

Valid Values: A time that can include minutes (m) or seconds (s)

Dependencies: None

This option has been added to set the maximum duration of the file transfer session associated with the transfer of the report files to the remote server for both regular FTP and SSH transfers. It ensures that the session will be closed in the event of a mishap on the remote server side during the transfer.

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 Gplus Adapter and the public key must be registered on the server hosting the WFM reporting application. Configuring an SSH connection is discussed in *Setting Up Secure File Transfer*.

Appendix: Secure File Transfer Problems 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: Deprecated

Default Value: *Not set*

Valid Values: Any alphanumeric password

Dependencies: User identified by `userName`, `password`

This option specifies the password for the user identified by the `userName` Option and is required to establish the connection to the ftp server but both *Genesys Administrator* and *Configuration Manager* display the password as plain text. The `password` Option should be used instead.

3.17 historical.reports Section

countIrregularIncoming

Type: Optional

Default Value: `false`

Valid Values: `true, false`

Dependencies: None

This option specifies whether non-routed calls received by an agent will be reported in the Agent Scorecard Metric report as internal calls and their handle time allocated toward Aux In time (if the agent was also in NotReady).

The default adapter behavior is to report only incoming calls with the Genesys CallType of Internal as internal calls. If this option is set to `true`, all non-routed incoming calls received by the agent are counted as internal no matter what their Genesys CallType (Inbound, Consult, etc.).

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 Gplus Adapter. Daily reports are not required for all WFM integrations since some are replaced by RTA applications or require that optional WFM modules be purchased.

The reports are contained in the `Agent_YYYYMMdd.txt` files.

directory

Type: Mandatory

Default Value: `ftp/reports`

Valid Values: Any valid folder on the Gplus 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 Gplus 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 Gplus 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

prefix

Type: Optional

Default Value: *Not set*

Valid Values: Any string compatible with a file name

Dependencies: None

This option specifies the prefix to apply to all historical report file names in the historical stream. If individual reports require unique prefixes, separate streams must be used. The remaining file names are unchanged after the prefix.

reportRetentionDays

Type: Mandatory

Default Value: `30`

Valid Values: Any positive integer

Dependencies: `historical.ftp/localBackupDirectory`

This option specifies the number of days that historical reports will be held in the report directory when not configured for FTP/SFTP transfers. The report files are removed permanently from the report directory as they expire.

serviceLevelFormula

Type: Mandatory

Default Value: `3`

Valid Values: `1 - 4` (formulae index listed below)

Dependencies: serviceLevelThreshold, abandonedThreshold

This option specifies which formula the Gplus 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` 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.

timeZone

Type: Optional

Default Value: None

Valid Values: Listed in *Appendix: Time Zone ID List*

Dependencies: None

This option can be used to set the time zone for this historical stream. If not set, the value in *timeZone* will be used instead.

Note: The Adapter must be restarted for changes to take effect; restarting the individual stream is not sufficient. This is a known issue.

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 and im 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.mediaAgentReportEnabled

Type: Optional

Default Value: `false`

Valid Values: `true, false`

Dependencies: historical.reports/mediaAgentReportEnabled

This option can be used to daily multimedia agent report (chat only at present). These reports are contained in the `Media_MMddhhmm.txt` files.

verint.version

Type: Optional

Default Value: `legacy`

Valid Values: `legacy, 11.x`

Dependencies: Verint Impact 360 version

This Option configures how the statistics are generated for the historical reports. There are two possible settings: `legacy` that creates the reports in the same way as earlier versions of the Adapter and `11.x` that meets the new report specifications for the Impact 360 version 11.0 and later. The differences are described generically for all of the media types below:

Legacy

- The Average Handle Time (AHT) statistic in the reports includes only the handle time that took place in the interval.
- The interaction counts (offered, accepted etc.) are incremented in the interval that the interaction was queued, answered etc.

11.x

- The interaction statistics are reported at the conclusion of the interaction's handle time. Identifying the point when the handle time has "concluded" is dependent upon whether the wrap time immediately follows the time that the agent is an active party to the interaction. If the wrap time is contiguous with that time, then the conclusion of the interaction's handle time occurs at the end of the wrap time.
- One example is that the Volume Handled (VH) statistic in the voice Contact Statistics report is the count of the calls where the handle time concludes in the reported interval.
- The Average Handle Time (AHT) statistic includes all of the handle time (including any time from prior intervals) of the interactions that "concluded" in the reported interval.

The Option must be set in the `historical.reports` Option Section as well as all of the configured historical Stream sections. The historical Stream sections cannot have a different Option value than the setting in the `historical.reports` Section.

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.

3.18 historical:<streamName> Section

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

Report Options

- *countIrregularIncoming*
- *dailyReportEnabled*
- *enabled*
- *filter*
- *includeOutboundInternalInNOC*
- *intervalReportEnabled*
- *intervalReport.interval*
- *prefix*
- *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.
- *timeZone*
- *verint.campaignReportEnabled*
- *verint.chatReportEnabled*
- *verint.emailReportEnabled*
- *verint.mediaAgentReportEnabled*
- *verint.version*
- *verint.voiceReportEnabled*

There is no `directory` option for the Stream configuration as the adapter automatically creates sub-directories in the directory configured by the *directory* Option. The created stream directory uses this Section's `streamName` as its name.

FTP Options

- *agentDailyDirectory*
- *backupRetentionDays*
- *campaignDirectory*
- *chatDirectory*
- *connectionMode*
- *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 WFM server.
- *keyPath*
- *keyPassphrase*
- *remoteDestinationDirectory*
- *remoteHost*
- *remotePort*
- *secureTransfer*
- *sessionTimeout*
- *userName*
- *userPassword*
- *password*

3.19 im Section

This Section is used to configure the Gplus Adapter instance if internal *Interaction Workspace Instant Messaging* interactions are being used by the Agents and those interactions are to be reported to the WFM application.

defaultAbandonedThreshold

Type: Optional

Default Value: `0s`

Valid Values: *Threshold Time 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 *serviceLevelFormula* actually use the abandoned threshold so this option has no effect unless one of those formulae (1 or 4) has been selected. An im interaction is considered abandoned if the first agent leaves the im session prior to the second agent replying.

Individual abandoned thresholds can be set for each im target/queue by adding `Annex Data` to the im `Virtual Queues` as shown in *Adding Annex Data for Virtual Queues and Skills* 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: *Threshold Time Format*

Dependencies: None

This option specifies the default service threshold for the service level calculations that form part of the interval reports. An im interaction is considered *serviced* at the time that the second agent first joins the im session with the first agent.

This option should be set when a single service threshold is used as a default for all im targets. Individual service thresholds can be set for each im target/queue by adding `Annex Data` to the im `Virtual Queues` as shown in *Adding Annex Data for Virtual Queues and Skills* 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.

defaultShortAbandonThreshold

Type: Optional

Default Value: *Not set*

Valid Values: *Threshold Time Format*

Dependencies: `application/shortAbandonEnabled`

This option specifies the default time threshold that an abandoned im interaction must exceed in order to be included in the *Actual Abandons (ABD)* column of the *Chat Contact Statistics* report. This option does not affect the service level calculations.

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 *Instant Messaging* (im) historical and RTA activity in the contact center.

interaction.reason.determiner

Note: Contact Professional Services before using this option.

Type: Optional

Default Value: *Not set*

Valid Values: See *Event Attribute Determiners*

Dependencies: `chat/enabled, rta/enabled`

This option specifies the attribute of an *Instant Messaging* (im) 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 im interaction.

Certain implementations may require that the combination of a `RoutedIM` 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*.

maxInteractionTime

Type: Optional

Default Value: *Not Set*

Valid Values: *Threshold Time Format*

Dependencies: None

While unlikely, it is possible for some interactions to become *stuck* in certain states if the terminating events are not received due to a mishap with an Interaction Server connection. Normally, the *stuck* interaction records in the Adapter would be terminated as soon as the connection was lost but this response relies on the *Setting addp Protocol* being configured properly for the connection(s) to the Interaction Server(s).

This option sets a timeout that defines the maximum amount of time that an *Instant Messaging* (im) interaction can remain in a *handled* or *wrap* state before the Adapter automatically terminates its associated record in the data tables. If the Option is not set, the *stuck* records are not terminated until the agent logs out or any agent logs in to the Place associated with the interaction.

If used, this timeout must be set to a value significantly higher than the expected longest IM duration to ensure that an ongoing interaction is not terminated prematurely in the Adapter tables. It should be noted that the timeout also applies to wrap up sessions so the expected duration of the wrap time must be taken into account.

3.20 individualDNs Section

This section is optional and must be added manually as it is not included in the Application Template. The intent of this Option section to allow for monitoring a small number of DNs that cannot presently be identified with the existing Options. The DN number is the Option key and the value field can be left blank.

3.21 interactionCodeMappings Section

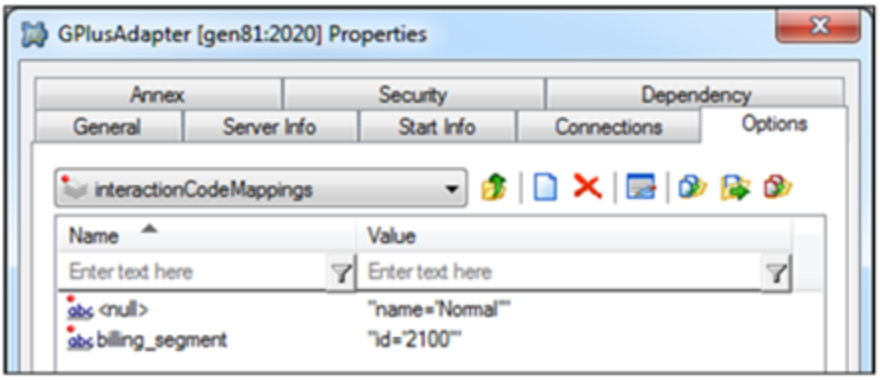
This section is optional and must be added manually as it is not included in the Application Template. When the section is included, the reasons parsed by the `interaction.reason.determiner(s)` will be compared against the Option key names in this section. If there is a match, the parameters listed in 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 *Real Time Adherence (RTA)*. The `name` field is the new Reason that will be included with the RTA state message sent to the WFM 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.



3.22 interactionTargets Section

This Section is optional but it may be 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.

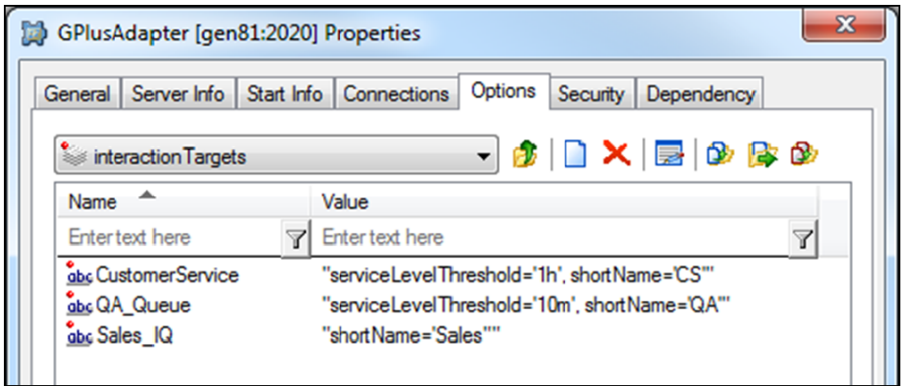
Each Option name represents the specific name of an actual `Interaction Queue` defined in the Configuration Server. For each `Interaction Queue`, the Option value can be used to specify the `serviceLevelThreshold`, `abandonedThreshold`, `shortAbandonThreshold` (see *Threshold Time Format*) and the `shortName`. These settings will then 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`



3.23 ixnProcessObjects Section

This Section is optional, but it may be required in certain configurations.

In typical configurations, interaction server interactions arrive on an interaction queue and are then associated to a virtual queue (VQ) providing the VQ reporting. The optional `ixnProcessObjects` section should be used when interaction queue reporting is desired instead, thereby using the reporting “queue” object on the interval report.

The section contains a list of Interaction Queues and their purpose according to the Gplus Adapter to determine how to report on digital interaction activity at various stages of handling. These `Interaction Queues` are monitored by the Adapter for queueing and terminating events.

All `Interaction Queues` involved with the following actions must be added and categorized in this section:

- completion of an interaction
- transferring an interaction
- holding an interaction for quality assurance handling

The option `Name` represents the specific name of an `Interaction Queue` configured in the Genesys environment. The option `Value` categorizes this queue with one of the interaction handling roles defined by the Gplus Adapter.

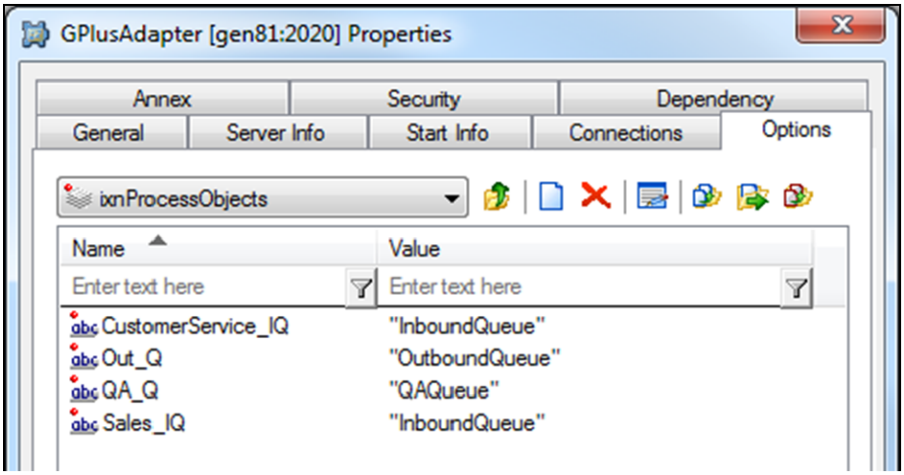
Valid Value	Description
InboundQueue	Use for reporting of queue events for interaction queues
OutboundQueue	Use for reporting of interactions completed or cancelled
QAQueue	Use for reporting queue events of the outbound queue of the original interaction queue and inbound queue of the QA queue
ObservationQueue	Use for monitoring the queue for abandons (no queue reporting will occur)

Type: Mandatory for QAQueues

Default Value: *Not Set*

Valid Values: `InboundQueue, OutboundQueue, QAQueue, ObservationQueue`

Dependencies: `email/enabled, chat/enabled, im/enabled, media:<name>/enabled`



3.24 media:<name> Section

The present version of the Gplus Adapter has defined configuration Options for each of the following media:

- voice
- email
- outboundpreview
- chat
- im

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

Note: Contact Professional Services prior to configuring this option. It requires a detailed understanding of the two related media.

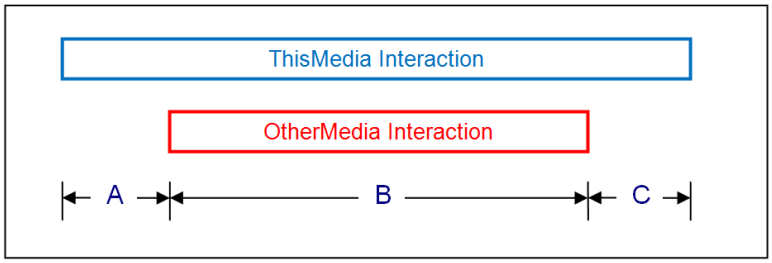
Type: Optional

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 (AfterOpenMediaWork) 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

defaultAbandonedThreshold

Type: Optional

Default Value: `0s`

Valid Values: *Threshold Time 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 *Adding Annex Data for Virtual Queues and Skills* 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: *Threshold Time 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 *Adding Annex Data for Virtual Queues and Skills* 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.

defaultShortAbandonThreshold

Type: Optional

Default Value: `0`

Valid Values: *Threshold Time Format*

Dependencies: `application/shortAbandonEanabled`

This option specifies the default short abandon threshold for the service level calculations that form part of the interval reports. If `application/shortAbandonEnabled` is set to false, this option has no effect.

This option should be set when a single short abandon threshold is used as a default for all `ThisMedia` targets. Individual short abandon thresholds can be set for each `ThisMedia` target/queue by adding `Annex Data` to the `ThisMedia Virtual Queues` as shown in *Adding Annex Data for Virtual Queues and Skills*. 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 Gplus Adapter to report on unattached wrap up (AfterInteractionWork) time. This option is only applicable when the `application/countUnattachedACW` Option is set to true.

Unattached AfterInteractionWork 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

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

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.

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

Note: Contact Professional Services before using this option.

Type: Optional

Default Value: *Not set*

Valid Values: See *Event Attribute Determiners*

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*.

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.

maxInteractionTime

Type: Optional

Default Value: *Not Set*

Valid Values: *Threshold Time Format*

Dependencies: None

While unlikely, it is possible for some interactions to become *stuck* in certain states if the terminating events are not received due to a mishap with an Interaction Server connection. Normally, the stuck interaction records in the Adapter would be terminated as soon as the connection was lost but this response relies on the *Setting addp Protocol* being configured properly for the connection(s) to the Interaction Server(s).

This option sets a timeout that defines the maximum amount of time that an interaction of the `ThisMedia` type can remain in a handled or wrap state before the Adapter automatically terminates its

associated record in the data tables. If the Option is not set, the stuck records are not terminated until the agent logs out or any agent logs in to the Place associated with the interaction.

If used, this timeout must be set to a value significantly higher than the expected longest duration for the interactions of the `ThisMedia` type to ensure that an ongoing interaction is not terminated prematurely in the Adapter tables. It should be noted that the timeout applies to wrap sessions so the expected duration of the wrap time must be taken into account.

otherIdDeterminer

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

Type: Optional

Default Value: *Not Set*

Valid Values: See *Event Attribute Determiners*

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.

otherMediaTypes

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

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`.

reason.notReady.determiner

Note: Contact Professional Services before using this option.

Type: Optional

Default Value: `eventAttribute: ReasonDescription`

Valid Values: See *Event Attribute Determiners*

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*.

reason.ready.determiner

Note: Contact Professional Services before using this option.

Type: Optional

Default Value: *Not set*

Valid Values: See *Event Attribute Determiners*

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 an `EventReadyForMedia` event.

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*.

reportingType

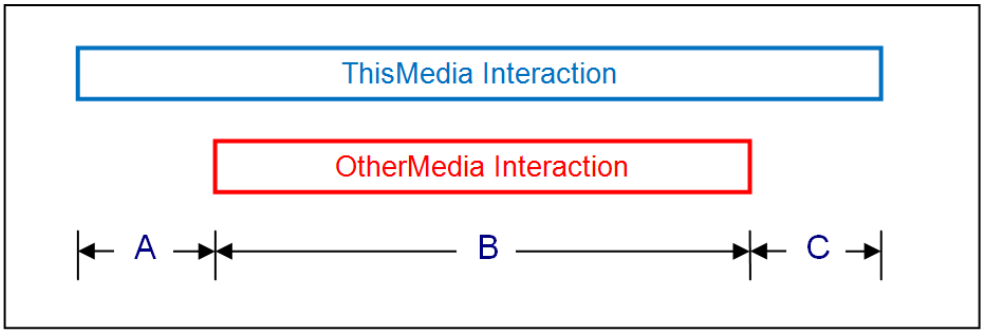
Note: Contact Professional Services prior to configuring this option as it will require a detailed understanding of the association between the two related media.

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.

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 WFM's 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 *Real Time Adherence (RTA)*.

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 WFM's 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 *Real Time Adherence (RTA)*.

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 WFM's 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 *Real Time Adherence (RTA)*.

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

Note: Contact Professional Services prior to configuring this option as it will require a detailed understanding of the medias' association.

Type: Optional

Default Value: *Not Set*

Valid Values: See *Event Attribute Determiners*

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.

3.25 outbound Section

classifyPartyDeterminer

Type: Optional

Default Value: `eventAttribute:UserData.GSW_DISPOSITION_CODE`

Valid Values: See *Event Attribute Determiners*

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 Section` 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 Gplus 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 RP will be included in the Dials column of the Outbound Statistics Report.

enabled

Type: Optional

Default Value: false

Valid Values: true, false

Dependencies: *campaignTarget.determiner*

This option enables reporting of outbound campaign calls in the *Outbound Statistics* report.

See *Outbound Voice Campaigns* for more information.

ignoreCampaignConsults

Type: Optional

Default Value: `false`

Valid Values: `true, false`

Dependencies: *outbound/enabled*

This Option configures how the Adapter will allocate a routed consult call originating from an outbound Campaign call. Setting this Option to `true` will prevent the `campaignTarget.determiner` from being applied to the queued consult call and the determiner specified by the `event.-properties/callType.determiner` will be used instead.

interaction.reason.determiner

Type: Optional

Default Value: none

Valid Values: See *Event Attribute Determiners*

Dependencies: `outbound/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*.

maxInteractionTime

Type: Optional

Default Value: *Not Set*

Valid Values: *Threshold Time Format*

Dependencies: None

While unlikely, it is possible for some calls to become *stuck* in certain states if the terminating events are not received due to a mishap with a TServer connection. Normally, the *stuck* interaction records in the Adapter would be terminated as soon as the connection was lost but this response does rely on the *Setting addp Protocol* being configured properly for the connection(s) to the TServer(s).

This option sets a timeout that defines the maximum amount of time that an outbound campaign call can remain in a *handled* or *wrap* state before the Adapter automatically terminates its associated record in the data tables. If the Option is not set, the records are not terminated until the agent logs out or any agent logs in to the Place associated with the call.

If used, this timeout must be set to a value significantly higher than the expected longest campaign call duration to ensure that an ongoing call is not terminated prematurely in the Adapter tables. It should be noted that the timeout also applies to `ACW` sessions so the expected duration of the wrap time must be taken into account.

pushPreviewMediaType

Type: Optional

Default Value: `outboundpreview`

Valid Values: Any text string

Dependencies: Push Preview outbound record media type

This option allows the expected media type name for *Push Preview* outbound records to be changed from the default, `outboundpreview`. A mismatch will result in the dialed outbound calls not being properly associated with an Outbound campaign.

Note: Applicable exclusively to routed push preview implementations and does not pertain to direct push preview.

recordProcessedOnRelease

Type: Optional

Default Value: `true`

Valid Values: `true, false`

Dependencies: Predictive or Progressive Campaign Type, OCS `OCServer/record_processed` Option

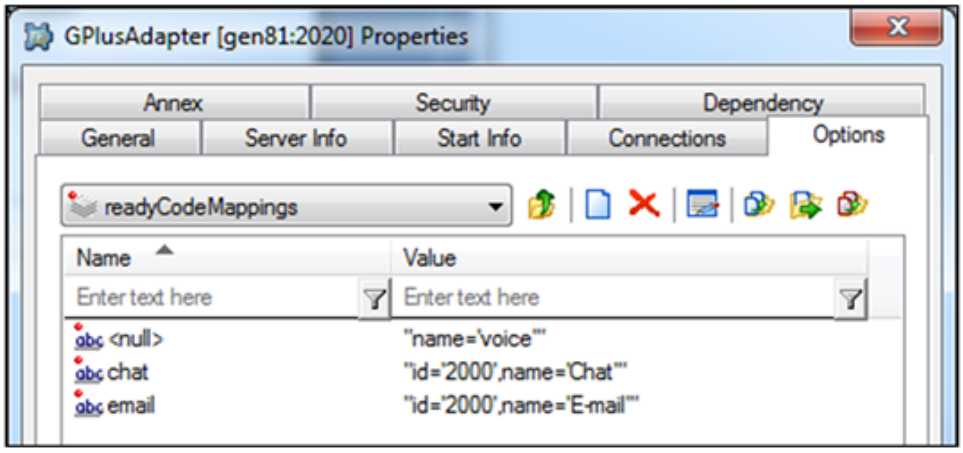
This option sets when the Gplus Adapter considers the Outbound Contact Server (OCS) outbound call record to be finished. If this option is set to `true`, then the record is marked as processed when the outbound call is released. If the option is set to `false`, the record is considered processed when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

3.26 readyCodeMappings Section

Note: Contact Professional Services before configuring this 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 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 *Real Time Adherence (RTA)*. The `name` field is the new Reason that will be included with the RTA state message sent to the WFM server.

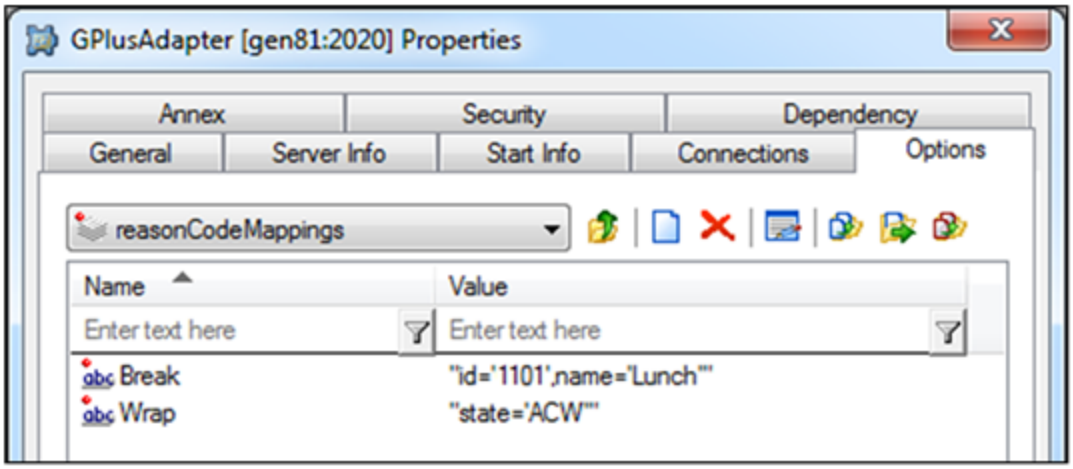
The `<null>` Option key signifies an event with no reason attached and it is used to configure a default behavior for the *Ready* events.

3.27 reasonCodeMappings Section

Note: Contact Professional Services before configuring this 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(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 three possible parameters:

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



The `id` field is the new RTA State Name as defined in *Real Time Adherence (RTA)*. The `name` field is the new Reason that will be included with the RTA state message sent to the WFM server. 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.

3.28 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 Gplus Adapter installations where RTA data is not required.

extendedPhoneStates

Type: Optional

Default Value: `false`

Valid Values: `true, false`

Dependencies: None

This Option adds OffHook (1307), Dialing (1308), and Ringing (1309) phone states to the reported RTA states if set to `true`.

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.

prioritizeStates

Type: Optional

Default Value: `true`

Valid Values: `true, false`

Dependencies: None

This Option specifies whether the RTA states with the same priority are collapsed into a single state or if each will be reported separately. RTA states with the same priority would be things like Ready on voice, but not chat collapses to Ready, and when going Ready on chat is still Ready and would not be reported when this option is set to true, as the state has not changed, but would be reported if set to false.

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.

verint.listenPort

Type: Optional

Default Value: 12000

Valid Values: Any open port on the Gplus Adapter host

Dependencies: None

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

verint.longAgentID

Type: Optional

Default Value: false

Valid Values: true, false

Dependencies: Must align with Verint WFM configuration

This option specifies the Agent ID length to expect from the Verint RTA application. The default value, `false`, corresponds to the Agent ID length of 24-characters. The `true` value corresponds to an Agent ID length of 128-characters. This selection must align with the Verint WFM configuration - see Verint's Advanced configuration parameter `KEEP_CCM_PLUGIN_ORIGINAL_AGENTID_LENGTH`.

3.29 rta:<streamName> Section

This Section is optional and is used when the Gplus 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 WFM Server. The Options are listed below but are not discussed here as they directly correspond to those listed in the *rta Section*.

Options

- *enabled*
- *extendedPhoneStates*
- *filter*
- *prioritizeStates*
- *reportDndAsNotReady*
- *verint.listenPort*
- *verint.longAgentID*

3.30 vht Section

This Section is optional and is used to configure how the Gplus Adapter instance tracks VirtualHold callbacks. The Adapter presently only supports *Virtual Hold – Concierge* mode and tracking other VirtualHold modes could result in unexpected behavior. The other requirement is that the call must be redirected from the IVR to a monitored `VQ` after the caller has selected the Virtual Hold callback

option. This VQ is used to initialize the information maintained by the Adapter to track the callback and the queued event starts the timer for the queue delay calculation.

The default behavior is to treat the initial inbound customer call and the delayed callback as two calls. The options listed configure the association between the two calls and treat the callback delay as "queue time" for a single call that is included in the `Offered` statistic of the *Contact Statistics* report.

Note: Contact Professional Services before configuring this section.

attribute.determiner

Type: Optional

Default Value: *Not set*

Valid Values: See *Event Attribute Determiners*

Dependencies: `voice/enabled`

This Option identifies the call attribute that provides the link between the initial customer call and the eventual Virtual Hold callback. Setting this option enables the functionality where the Adapter tracks the linked calls as a single call with the extended queue time.

The specified linking attribute must be contained in the TServer/SIPServer `EventDiverted` generated when a caller chooses a Virtual Hold callback rather than waiting for the next available agent. As mentioned above, the call must be queued in and diverted from a monitored VQ.

The Adapter first parses the call target from the attributes of the `EventQueued` using the `event.-properties/callType.determiner`. It then parses the linking attribute from the `EventDiverted` generated as the call leaves the monitored VQ. The parsed attribute value must be the unique call ID that will also be attached to the outbound callback.

The presence of the linking attribute is then checked in the `EventDialing` generated from the monitored `Trunk Group` or `Routing Point` at the start of the callback. It will also be checked in the `EventEstablished` generated when the Agent picks up the outbound callback. The same `attrib-ute.determiner` is used for the initial customer call and the subsequent callback(s) so the linking attribute must have the same key and value for both calls.

callbackAccepted.determiner

Type: Optional

Default Value: Not set

Valid Values: See *Event Attribute Determiners*

Dependencies: *voice/enabled*

This Option identifies the call attribute that indicates whether this callback was accepted by the customer. Leave this option blank unless needed, which will depend on the implementation. This option cannot be used without setting the `callbackAcceptedValue` option.

callbackAcceptedValue

Type: Optional

Default Value: Not set

Valid Values: any valid string

Dependencies: *voice/enabled*

This Option identifies the value that indicates that a callback was accepted by the customer. This option is used in conjunction with `callbackAccepted.determiner`. Any value that does not match this value (null does not match anything) is considered to be not accepted.

defaultRetries

Type: Optional

Default Value: `1`

Valid Values: `1 to 1000`

Dependencies: None

The option specifies the maximum number of callback attempts that will be made to contact the customer. It should be exactly the same value as the Virtual Hold setting that specifies the number of callbacks that will be attempted. Once this limit has been reached without an answer from the customer, the initial customer call is reported as *Abandoned*.

It should be noted that the retry count maintained by the Adapter for the callbacks will only be incremented if the linking attribute described above is contained in the `EventDialing` generated from the monitored `Trunk Group` or `Routing Point`.

This option can also be set for an individual VQ using the `vhtRetries` Option as described in the *vhtExpiry, vhtFlushTime, vhtRetries* section.

defaultTimeout

Type: Optional

Default Value: `24h`

Valid Values: *Threshold Time Format*

Dependencies: None

This option specifies the amount of time (starting from the initial queued event) that the Adapter will wait before reporting the initial call as *Abandoned* if none of the callbacks are successful.

This option can also be set for an individual VQ using the `vhtExpiry` Option as described in the *vhtExpiry, vhtFlushTime, vhtRetries* section.

timeToFlushAt

Type: Optional

Default Value: *Not set*

Valid Values: `00:00 to 23:59` (24 hour clock - GMT)

Dependencies: None

The Virtual Hold application can be configured to clear or flush all of the pending callbacks daily. This Option should be set to the time of day when that "flush" occurs. Inbound calls associated with the flushed pending callbacks will be reported as *Abandoned*.

This option can also be set for an individual VQ using the `vhtFlushTime` Option as described in the *vhtExpiry, vhtFlushTime, vhtRetries* section.

3.31 voice Section

defaultAbandonedThreshold

Type: Optional

Default Value: `0s`

Valid Values: *Threshold Time Format*

Dependencies: *serviceLevelFormula*

This option specifies the default abandoned threshold for the service level calculations that form part of the interval reports. Only two of the *serviceLevelFormula* 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 *Adding Annex Data for Virtual Queues and Skills*. These individual thresholds will override the default specified with this option.

defaultServiceThreshold

Type: Optional

Default Value: `0s`

Valid Values: *Threshold Time 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 the *interactionTargets Section*. These individual thresholds will override the default specified by this option.

defaultShortAbandonThreshold

Type: Optional

Default Value: *Not set*

Valid Values: *Threshold Time Format*

Dependencies: *application/shortAbandonEnabled*

This option specifies the default time threshold that an abandoned call must exceed in order to be included in the *Actual Abandons (ABD)* column of the *Forecast* report. This option does not affect the service level calculations.

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 Gplus 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 Gplus Adapter was installed to only track the agents' other media interactions.

interaction.reason.determiner

Note: Contact Professional Services before using this option.

Type: Optional

Default Value: *Not set*

Valid Values: See *Event Attribute Determiners*

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*.

isAfterCallWork.class

Type: Optional

Default Value: `DefaultAfterCallWorkProperty`

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 AfterCallWork.

maxInteractionTime

Type: Optional

Default Value: *Not Set*

Valid Values: *Threshold Time Format*

Dependencies: None

While unlikely, it is possible for some calls to become *stuck* in certain states if the terminating events are not received due to a mishap with a TServer connection. Normally, the *stuck* interaction records in the Adapter would be terminated as soon as the connection was lost but this response relies on the *Setting addp Protocol* being configured properly for the connection(s) to the TServer(s).

This option sets a timeout that defines the maximum amount of time that an inbound routed call can remain in a *handled* or *wrap* state before the Adapter automatically terminates its associated record in the data tables. If the Option is not set, the records are not terminated until the agent logs out or any agent logs in to the Place associated with the call.

If used, this timeout must be set to a value significantly higher than the expected longest routed inbound call duration to ensure that an ongoing call is not terminated prematurely in the Adapter tables. It should be noted that the timeout also applies to `ACW` sessions so the expected duration of the wrap time must be taken into account.

parallelQueuesEnabled

Note: Contact Professional Services prior to changing this option.

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.

reason.notReady.determiner

Note: Contact Professional Services before using this option.

Type: Optional

Default Value: `eventAttribute: Reasons.ReasonCode`

Valid Values: See *Event Attribute Determiners*

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 an `EventAgentNotReady` event.

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*.

reason.ready.determiner

Note: Contact Professional Services before using this option.

Type: Optional

Default Value: *Not set*

Valid Values: See *Event Attribute Determiners*

Dependencies: *voice/enabled*, *rta/enabled*

This option specifies the attribute a voice TServer/SIPServer event that will be included as the *Reason* field of an RTA state message generated by an `EventAgentReady` event.

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*.

4 Additional Configuration

4.1 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.

4.2 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
```

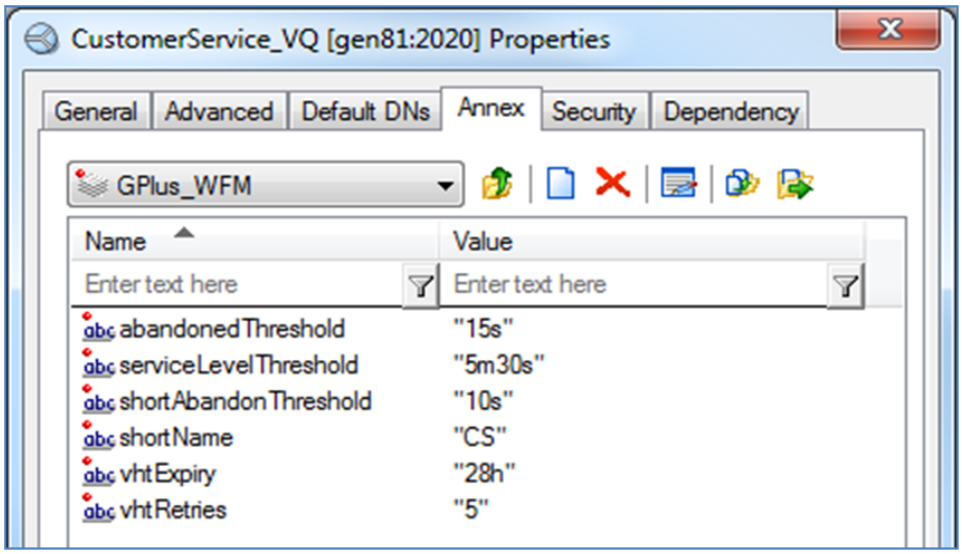
Examples:

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

4.3 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. The following parameters can be added to these Configuration Objects to modify the reported results:

- `abandonedThreshold` - Skill or VQ
- `serviceLevelThreshold` - Skill or VQ
- `shortAbandonThreshold` - Skill or VQ
- `shortName` - Skill or VQ
- `vhtExpiry` - VQ only
- `vhtFlushTime` - VQ only
- `vhtRetries` - VQ only



abandonedThreshold, serviceLevelThreshold, shortAbandonThreshold

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

Configuring abandoned thresholds for an email Virtual Queue while allowed would have little effect as emails are not abandoned.

The 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 *Contact Statistics* reports.

Conversely, the `shortAbandonThreshold` option does remove interactions from the *Actual Abandons* count if those interactions are abandoned prior to the threshold time but the option does not affect the service level calculations.

The figure above 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.

The following steps should be followed to add a `serviceLevelThreshold`, `abandonedThreshold` or `shortAbandonThreshold` 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, abandonedThreshold or shortAbandonThreshold with value equal to the new threshold as described in *Threshold Time Format*.

shortName

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.
4. The short name must be unique with respect to all other short names.

vhtExpiry, vhtFlushTime, vhtRetries

These three *Virtual Hold* options can only be added to the Annex tab of a Virtual Queue that queues an initial customer inbound call. They provide a VQ level of configuration versus the three *vht* options, *defaultRetries*, *timeToFlushAt* and *defaultTimeout*, that set application level defaults for the *Virtual Hold* configuration.

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 vhtExpiry with the Value set to the amount of time that the Adapter will wait for a successful callback using the *Threshold Time Format*.
3. Add a new option to this section named vhtFlushTime with the Value set to the time during the day that the Adapter will flush the calls that have not resulted in a successful callback and count those calls as *Abandoned*. This time setting will be in the GMT in the range from 00:00 to 23:59.
4. Add a new option to this section named vhtRetries with the Value set to an integer value equal to the number of callback attempts that could be made before the call is reported as Abandoned.

4.4 Adding A PlaceGroup to a Site

All of the reports with the exception of the daily Agent Productivity report can include a Site field if required. The Gplus 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.

4.5 Event Attribute Determiners

There are a number of *determiner* Options in the Adapter configuration. These Options are used to identify an event attribute that can be used for any number of purposes; some of which are listed below:

- Contact queue or target
- Not Ready reason
- Outbound Campaign name
- Link attribute for Virtual Hold™ and Genesys callbacks

All of the determiners share common functionality and work in exactly the same way.

Event Attributes

The Adapter works with the low level events received from the Genesys Servers and the configurable determiners are the main tools used to access information from those events. An event is essentially a collection of key/value pairs (attributes) that describe the event and the determiners retrieve information from one of those key/value pairs.

An example event that will be used to demonstrate how the reason.notReady.determiner can be configured is shown below. One thing to note is that the value part of the key/value pair can also be a collection of key/value pairs; the attributes can be nested in other attributes. Both the *Extensions* and *Reasons* attributes are key/value collections. However, the *GPlusWFM* prefix does not indicate a collection as it just identifies the extra attributes added to the event by the Adapter.

```
EventAgentNotReady
  AgentID: 12345
  AgentWorkMode: AuxWork
  EventSequenceNumber: 123456789
  Extensions.ReasonCode: 5
  Extensions.AgentSessionID: ABCDEFGHIJ1234567890KLMNPQR
  GPlusWFM.LocalTime: 1234567890000
  GPlusWFM.RemoteTime: 1234567890000
  GPlusWFM.SourceId: 101
  GPlusWFM.SourceName: TServer_A
  GPlusWFM.Time: 1234567890000
  Reasons.Break: 5
  ThisDN: 60000
  ThisQueue: 3000
  TimeinSecs: 1234567890
  TimeinuSecs: 000000
```

Determiner Types

The event attributes can be accessed by one of the three determiner types:

eventAttribute

This determiner type specifies the actual attribute key name when identifying the attribute and returns that attribute's value.

```
eventAttribute: Extensions.ReasonCode  -> 5
```

eventAttributeKey

This determiner type specifies the attribute name for the collection attribute that contains the actual attribute of interest. The key name for the first attribute in the collection is returned.

```
eventAttributeKey: Reasons  -> Break
```

eventAttributeValue

This determiner type specifies the attribute name for the collection attribute that contains the actual attribute of interest. The value for the first attribute in the collection is returned.

```
eventAttributeValue: Reasons  -> 5
```

The `eventAttributeKey` and `eventAttributeValue` determiners address a problem that usually arises with NotReady reasons where the attribute key name is descriptive (e.g., Break, Meeting, Lunch etc.) and as a result, there are multiple possible key names that cannot be addressed with a single `eventAttribute` determiner.

Multiple Determiners

Determiners can also be combined to parse more than one attribute in an event. The Adapter iterates though the list of determiners until a value is returned. The alternatives are formatted as a comma separated list:

```
eventAttribute: NonExistent, eventAttributeKey: Reasons  -> Break
```

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 Gplus 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.

The Gplus adapter can be configured to utilize ftp over SSH to deliver reports securely. This can be done with either a username/password combination, or with public/private key encryption. The adapter will also perform host verification against the SSH server using the `known_hosts` file or the `knownHostEntry` option.

A description of some of the possible errors due to a misconfigured client or server has been included as *Appendix: Secure File Transfer Problems*.

5.1 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 Gplus Adapter server and the public key is stored on the remote WFM 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:

- `aes{128,192,256}-{cbc,ctr}`
- `aes{128,256}-gcm@openssh.com`
- `blowfish-{cbc,ctr}`
- `chacha20-poly1305@openssh.com`
- `3des-{cbc,ctr}`
- `twofish{128,192,256}-{cbc,ctr}`
- `twofish-cbc`
- `serpent{128,192,256}-{cbc,ctr}`
- `idea-{cbc,ctr}`
- `cast128-{cbc,ctr}`
- `arcfour`
- `arcfour{128,256}`

As mentioned, the private key is stored with the Gplus 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 WFM 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.

5.2 Host Verification

The `known_hosts` file is located in `~/.ssh/known_hosts` for linux or `%USERPROFILE%\.ssh\known_hosts` for Windows and contains a list of entries of trusted public keys that will be used to verify the identity of a host when we try to connect to it. Each host entry is broken into 3 parts: the hostname, the algorithm, and the key, separated by spaces. Entries of this type can be generated using the `ssh-keyscan` command. For example:

```
ssh-keyscan -H <yourHostName>
```

Might display a line of output similar to this:

```
<yourHostName> ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIISnLYall/CYVBcGYW/rqGPKt6
```

The output of the `ssh-keyscan` can also be piped directly to your `known_hosts` file:

```
ssh-keyscan -H <yourHostName> >> ~/.ssh/known_hosts
```

For situations where the `known_hosts` file is inaccessible or can't be updated, host verification can also be performed by using the `knownHostEntry` option. This option accepts a value that represents a known host entry like the ones described above.

Host verification can optionally be disabled by setting the `enableHostVerification` option to false, but this is not recommended as it makes the adapter vulnerable to man-in-the-middle attacks.

5.3 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

enableHostVerification: `true` to enable SFTP host validation

keyPath: path and name of the private key file

keyPassphrase: pass phrase that allows access to the private key

knownHostEntry: represents an entry in a `known_hosts` file

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`

5.4 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. The public key authentication is more secure and more difficult to implement.

SSH Credentials

To connect to the SSH server, we need a username and account set up on the SFTP server with any applicable subdirectories as configured to be used by the adapter. Configuring the SFTP server is beyond the scope of this document as this is dependent on the particular SFTP server being used.

If using public/private keys is the desired configuration, we will need the private key associated with the user's login. If not using keys, skip this step. The adapter expects this file to be in PEM format. This will be the private key that will be used in authentication to the FTP server. Contact your SFTP server administrator for a key file.

6 Outbound Voice Campaigns

The Gplus 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 and how the Adapter classifies the calls associated with the campaigns.

There are six types of Genesys outbound campaigns that can be monitored by the Gplus Adapter:

1. Predictive
2. Progressive
3. Predictive with ASM
4. Progressive with ASM
5. Preview
6. Push Preview
7. Direct Push Preview

The first four, Predictive (ASM) and Progressive (ASM), have an outbound call routed to the agent after being dialed by a dialer while the Preview, Push Preview, and Direct Push Preview have the outbound dialed from the agent's DN.

The one common requirement is that there must be a campaign name in the event attributes to be parsed by the `event.properties/campaignTarget.determiner`.

6.1 Predictive and Progressive Campaigns

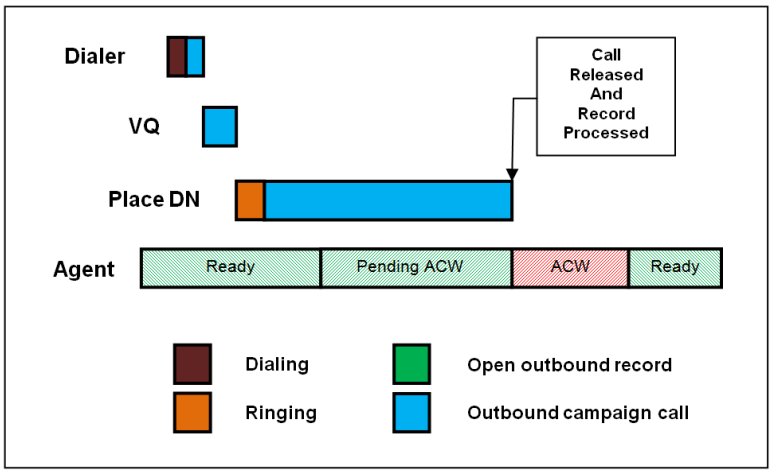
Both the Predictive and Progressive campaign types have call flows that start as a dialing event on a `Routing Point`, queued on a `Virtual Queue`, diverted to and then answered by an available agent with an optional ACW interval after the call is released. The adapter does not differentiate between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

The figures below illustrate the reason for including the *recordProcessedOnRelease* 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

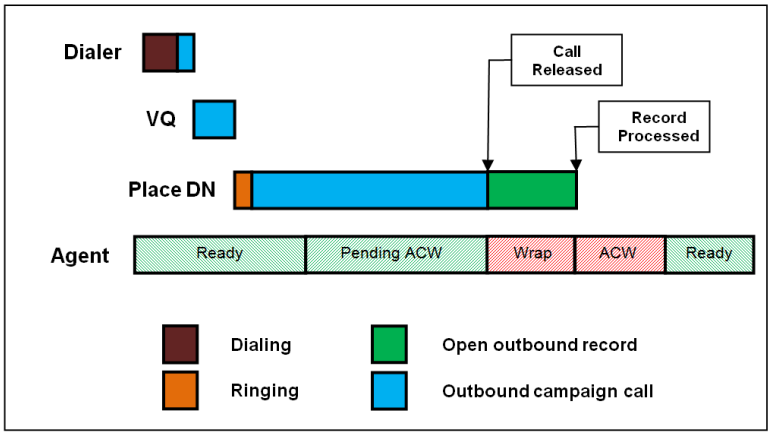
The first Figure shows the record being processed at the conclusion of the call. There is no wrap time associated with an open OCS record and any disposition codes must be attached to the call prior to the call being released.



recordProcessedOnReleased = false

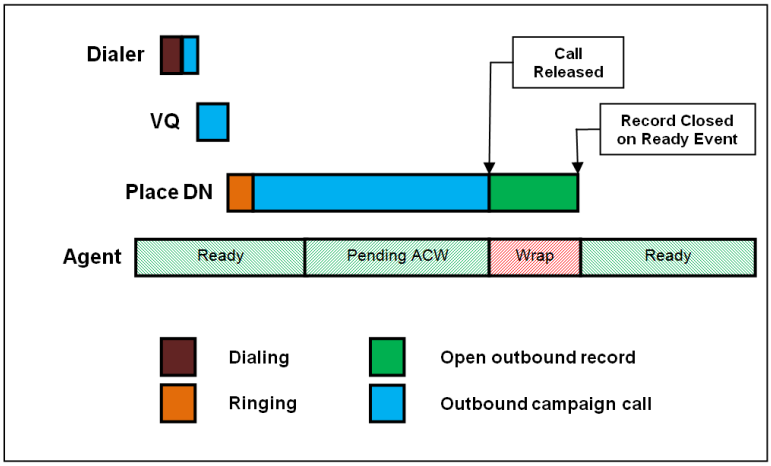
The second Figure shows the record being processed after the conclusion of the call. There is wrap time associated with an open OCS record and also ACW time associated with the call that is counted after the record is processed. Both will be counted as `AfterContactWorkTime` in the *Outbound* report.

Any disposition codes must be added to the call prior to the record being processed.



recordProcessedOnReleased = false – Record Closed Prematurely

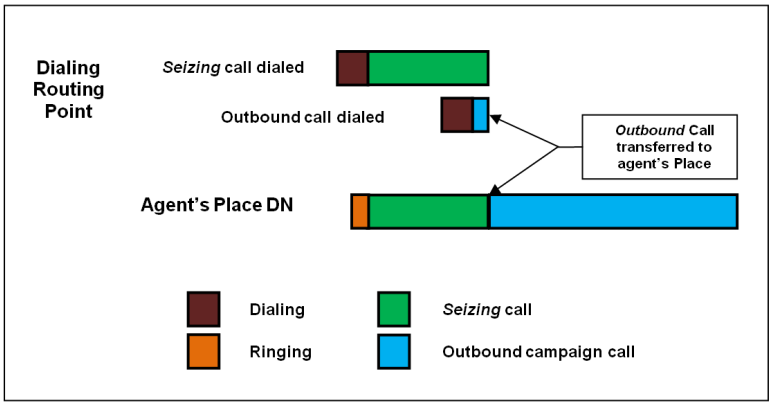
The third Figure shows the record being closed on an error generated by the OCS when the agent goes into a Ready state rather than marking the OCS record as done. Any errors caused by state changes or problems with the OCS will result in the Adapter considering the record closed. There is wrap time associated 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.



6.2 Predictive and Progressive with ASM

Agent Seizing Mode (ASM) adds an extra *seizing* call from the dialer to the start of the Predictive and Progressive call flows. This extra call ensures that an agent is engaged to join the successfully completed outbound call immediately after the campaign call is answered.

The figure below shows the extra activity at the start of the call for the *seizing* modes. After the outbound campaign call is transferred to the agent's Place, the call is tracked in exactly the same manner as a normal Predictive or Progressive call.



The Adapter ignores the contribution of the *seizing* call as long as there are no queued events associated with it. It must be noted that if the *seizing* call is routed to the agent through a monitored VQ, that call and the subsequent outbound campaign call will be counted as inbound customer calls. The simple solution to this situation is to exclude any VQs that might queue a *seizing* call.

6.3 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.

The Preview and Push Preview campaign types are restricted to using the `campaignTarget.determiner` to identify the campaign name. This is the default behavior for the Adapter and the campaign calls will be reported in the and reports.

6.4 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 only be counted in the *Agent Scorecard Metric* report as non-customer outbound activity.

7 Streams and Filters

7.1 Multiple Adapters

The initial versions of the Adapter were limited to providing a set of historical reports and one RTA data feed to a single WFM Server. *Stream* functionality was added to the Adapter with Version 5.1 to allow a single Adapter instance to provide historical reports and RTA data feeds to multiple WFM Servers. *Filters* provide a means to tailor the information provided through each *Stream* by including only the VQs, Places, outbound Campaigns and/or Agents that are to be reported upon through the associated *Stream(s).* One Adapter instance can essentially be configured as multiple Adapters.

There is nothing in the *Stream* and *Filter* functionality that could not be duplicated with multiple Adapter instances. What *Streams* and *Filters* provide is more effective resource utilization:

- Single connection to Genesys environment (Configuration Server, TServers, SIPServers, and Interaction Servers).
- Single `Application` configuration object in the Configuration Server.
- Runs in a single Java Virtual Machine (JVM).
- Extra memory required per *Stream* is minimal.

7.2 Media and Reports

There are two common *Stream* configurations:

- Consolidating multiple Adapters into a single instance
- Segregating media to their own reports

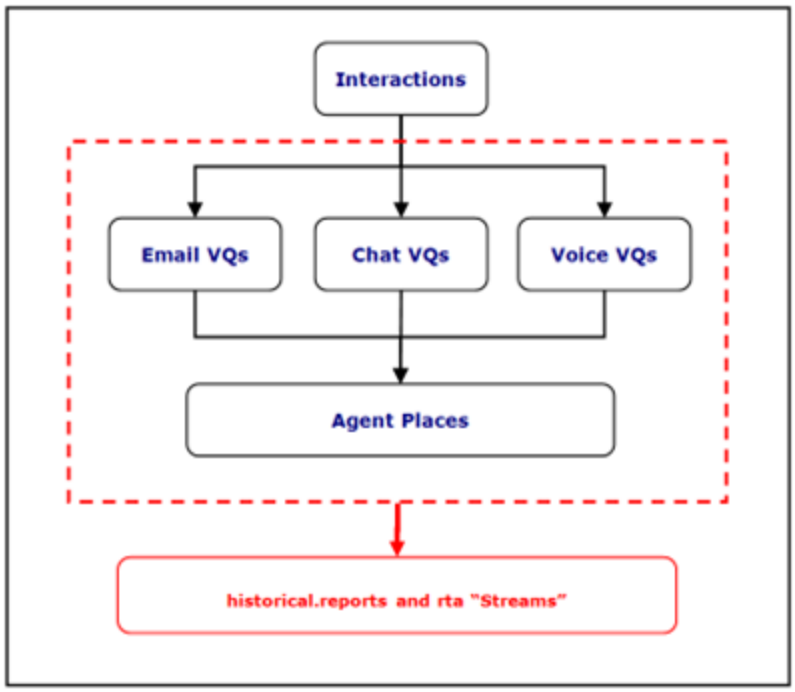
Individual *Filters* are configured as separate Option as described in the *filter:<filterName> Section*. A *Filter* defines a set of monitored `Agents`, `Places`, and `Virtual Queues` (`VQs`) that any associated Stream will report upon. Only `Agents`, `Places`, and `VQs` that are part of the groups 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.

No Filters and Streams

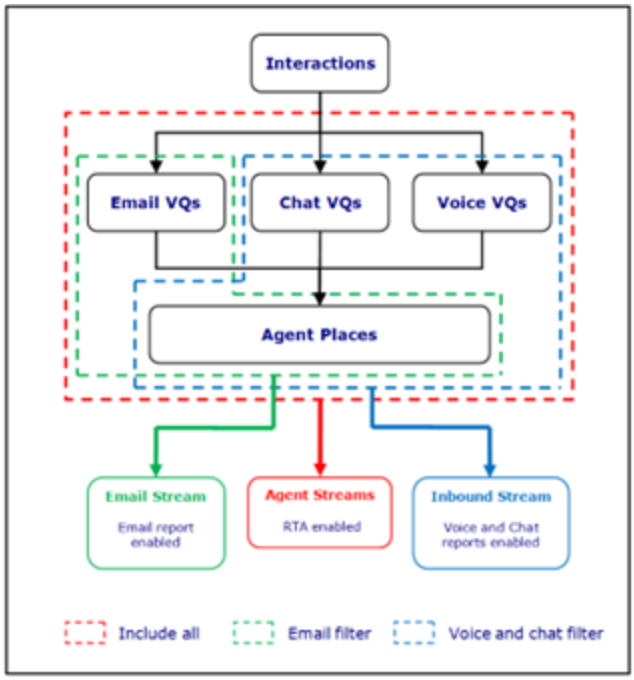
The figure below shows the Adapter configuration without configured *Filters* and *Streams*.

There is a mixing of media in the Queue report.



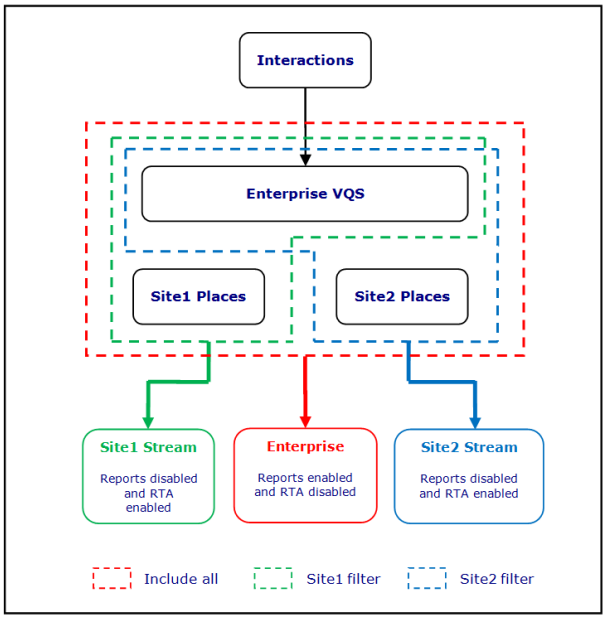
Separated By Media

The figure below shows one example of a solution to the media mix described above. Two *Filters* are configured to separate the email VQs from the chat and voice VQs while still including all of the agents.



Separated By Site

The figure below shows another example where the RTA feeds are separated by Site with the report still including all of the Agent activity.



8 Recovery, Restarts, and Reconnects

The Gplus Adapter was designed to run for extended periods with little or no maintenance. However, all of the tracked activity and configuration information is stored in in-memory tables and that data can be lost or left incomplete due to upsets or restarts. The following is a description of the mechanisms used by the Adapter to mitigate the effect of the upsets and restarts.

8.1 Recovery

Recovery refers to the processes in place for the Adapter to *recover* after a shutdown and restart. While running, the Gplus Adapter is continuously updating a log file with the events received from the Genesys servers. One of the first steps after a restart is to replay the current day's recovery log and rebuild the internal tables. That data is then used at the end of that day to generate the daily *Agent Productivity* report. While the report will be incomplete, it will include the agent activity prior to the Adapter stopping.

The interval *Forecast* reports will also be generated after the restart but the retained information from the recovery log is not used for those reports. Only the events received after the restart will be processed for these reports. For the most part, the statistics contained in the reports will accurately reflect the activity within an interval or two but there are some cases such as the *Backlog* statistic in the *BackOfficeEmail* report that might take days or longer to approach the actual number. This occurs because the Adapter is relying on received events to build the statistics. If long lived interactions are not generating events, those interactions are essentially invisible to the Adapter.

8.2 Restarts

A restart refers to a shutdown and restart that was not the result of an incident but rather a scheduled downtime for maintenance, an upgrade or some other planned activity. The Adapter still steps through the recovery log and rebuilds the internal tables but there is an additional step if a graceful shutdown was used to stop the Adapter.

Note: To gracefully shut down the Adapter, right click the adapter in Solution Control Interface or Administrator, select "Graceful Stop" and wait for the adapter to stop.

If the Adapter was stopped using *Graceful Stop* through the Solution Control server, a state dump file is written that contains a list of the queued interactions at the time of the shutdown. The adapter uses this list on restart to restore the backlogged interactions and query the Interaction Server to determine if those interactions are still active. Only those active interactions that are still either queued or retained in a workbin will be recovered. Interactions that are no longer queued are truncated. In some cases, those interactions may still be present but no longer in the queue. There is no method to retrieve the activity that occurred as the interaction transitioned from the queue so all of that activity is discarded. Recovered interactions will retain their initial queue times and associated target. Interactions in that list that could not be recovered are truncated.

The graceful shutdown list of queued interactions is restricted to Interaction Server interactions only and does not include queued voice calls. This is intended to mitigate the data loss that occurs for

long running interactions when there is an adapter restart. There will be a loss of data for any short term active interactions. This would suggest that any planned restarts should be scheduled for periods of little or no activity.

8.3 Reconnects

Disconnects and the subsequent reconnects can occur as the result of two different scenarios:

- the disconnected Genesys server had an upset that caused a shutdown and restart, or
- there was a network problem that resulted in the connection being dropped and eventually re-established.

The Adapter cannot differentiate these two based on the received events so the Adapter response is to treat it as if the Genesys server had undergone an upset and had to be restarted. It turns out that the SIPServers/TServers and the Interaction Server employ different strategies to deal with a restart which, in turn, results in the Adapter treating the disconnects from the two server types differently.

SIPServer/TServer

If a SIPServer or TServer undergoes an upset and must be restarted, all the former voice interactions are removed. The Adapter response is to truncate the records for all voice calls at the start of the disconnect phase whether those calls are queued or being handled by an agent. Again, it must be emphasized that the Adapter response is the same no matter what caused the disconnect and ongoing voice interactions are truncated even if the fault was a loss of the connection for a few seconds due to a network problem.

Interaction Server

The Interaction Server retains the prior active interactions on a restart but returns all of them to the original Interaction Queue. Any active interaction that was being handled by an Agent is no longer being handled.

The Adapter response is to query the Interaction Server on reconnect as to the status of the formerly active interactions. Interactions that are still queued either in an Interaction Queue or workbin are maintained in the Adapter tables and processed accordingly when any future activity occurs. Any other prior interactions are truncated in the Adapter tables and reported as completed.

A Appendix: Time Zone ID List

The list of acceptable values to specify the Gplus 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

ACT	AET	AGT
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

ACT	AET	AGT
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

ACT	AET	AGT
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

ACT	AET	AGT
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

ACT	AET	AGT
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

B Appendix: Secure File Transfer Problems

Below are examples of what the logging should look like, and what logging indicates a problem. Each line of logging is prefaced with something similar to the following:

```
09 Mar 2022 17:47:03,448 85580 [FtpTaskQueue]
INFO  com.ariasolutions.iconnect.historical.transfer.TransferManager [] –
```

B.1 Successful connection

Connecting successfully using public/private key should look something like:

```
Transferring files to WFM: [ftp\reports\030922.1745]
<yourDirectory>\ftp\reports\030922.1745 successfully transferred 1 of 1 files sent to WFM FTP server
Backing up file '<yourDirectory>\ftp\reports\Historical-030922.1746 to <yourDir-
ectory>\ftp\reportsBackup\030922.1745
FTP task completed
```

B.2 Misnamed or missing key file

A misnamed or missing key file will look like:

```
Transferring files to WFM: [ftp\reports\Historical-031022.1118, ftp\reports\rta.txt, ftp\re-
ports\rta.vendor.txt]
Error occured while uploading reports to WFM FTP server
com.ariasolutions.iconnect.historical.transfer.FtpException: Error occured while attempting to send files
C:\Users\<yourUser>\.ssh\testKey2 (The system cannot find the file specified)
    at com.ariasolutions.iconnect.historical.transfer.SftpTransferAgent.transferFiles(Sft-
pTransferAgent.java:309) ~[main/:?]
    at com.ariasolutions.iconnect.historical.transfer.TransferManager.transferFiles(Trans-
ferManager.java:354) [main/:?]
    at com.ariasolutions.iconnect.historical.transfer.TransferManager$1.run(TransferManager.java:207)
[main/:?]
    at com.ariasolutions.utils.concurrent.TaskFuture.run(TaskFuture.java:80) [main/:?]
    at com.ariasolutions.utils.concurrent.TaskQueue.run(TaskQueue.java:41) [main/:?]
    at java.lang.Thread.run(Thread.java:834) [?:?]
Caused by: java.io.FileNotFoundException: C:\Users\<yourUser>\.ssh\missingTestKey (The system cannot find
the file specified)
    at java.io.FileInputStream.open0(Native Method) ~[?:?]
    at java.io.FileInputStream.open(FileInputStream.java:219) ~[?:?]
    at java.io.FileInputStream.<init>(FileInputStream.java:157) ~[?:?]
    at java.io.FileReader.<init>(FileReader.java:75) ~[?:?]
    at net.schmizz.sshj.userauth.keyprovider.KeyProviderUtil.detectKeyFileFormat(KeyProviderUtil.java:36)
~[sshj-0.32.0.jar:?]
    at net.schmizz.sshj.SSHClient.loadKeys(SSHClient.java:559) ~[sshj-0.32.0.jar:?]
    at net.schmizz.sshj.SSHClient.loadKeys(SSHClient.java:514) ~[sshj-0.32.0.jar:?]
    at com.ariasolutions.iconnect.historical.transfer.SftpTransferAgent.transferFiles(Sft-
pTransferAgent.java:268) ~[main/:?]
    ... 5 more
Of 3 files prepared, 3 files were not sent to the WFM server
0 of 3 files sent to WFM FTP server
FTP task completed
```

B.3 Ftp server's host key not in known_hosts file

If the ftp server's host key is not in our `known_hosts` file (located in `~/.ssh/` or `<C:\Users\<yourUser>\.ssh\` by default) and is also not in the `knownHostEntry` option:

```
Transferring files to WFM: [ftp\reports\Historical-031022.1118]
Error occured while uploading reports to WFM FTP server
com.ariasolutions.iconnect.historical.transfer.FtpException: Error occured while attempting to send files
Could not verify `ssh-ed25519` host key with fingerprint `<yourServerFingerprintHere>` for `w2k12genlab`
on port 22
    at com.ariasolutions.iconnect.historical.transfer.SftpTransferAgent.transferFiles(Sft-
pTransferAgent.java:309) ~[main/:?]
    at com.ariasolutions.iconnect.historical.transfer.TransferManager.transferFiles(Trans-
ferManager.java:354) [main/:?]
    at com.ariasolutions.iconnect.historical.transfer.TransferManager$1.run(TransferManager.java:207)
[main/:?]
    at com.ariasolutions.utils.concurrent.TaskFuture.run(TaskFuture.java:80) [main/:?]
    at com.ariasolutions.utils.concurrent.TaskQueue.run(TaskQueue.java:41) [main/:?]
    at java.lang.Thread.run(Thread.java:834) [?:?]
Caused by: net.schmizz.sshj.transport.TransportException: Could not verify `ssh-ed25519` host key with fin-
gerprint `<yourServerFingerprintHere>` for `<yourHost>` on port 22
    at net.schmizz.sshj.transport.KeyExchanger.verifyHost(KeyExchanger.java:220) ~[sshj-0.32.0.jar:?]
    at net.schmizz.sshj.transport.KeyExchanger.handle(KeyExchanger.java:390) ~[sshj-0.32.0.jar:?]
    at net.schmizz.sshj.transport.TransportImpl.handle(TransportImpl.java:474) ~[sshj-0.32.0.jar:?]
    at net.schmizz.sshj.transport.Decoder.decode(Decoder.java:113) ~[sshj-0.32.0.jar:?]
    at net.schmizz.sshj.transport.Decoder.received(Decoder.java:200) ~[sshj-0.32.0.jar:?]
    at net.schmizz.sshj.transport.Reader.run(Reader.java:60) ~[sshj-0.32.0.jar:?]
Of 1 files prepared, 1 files were not sent to the WFM server
0 of 1 files sent to WFM FTP server
FTP task completed
```

B.4 Local source directory misconfigured

If the local source directory is misconfigured, all you will see is:

```
FTP source directory does not exist: C:\Users\rmccarthy.CORP\git\g-wfm5\GPlusWFM\ftp\reports_wrong
```

B.5 Unreachable / misspelled host

If you have an unreachable (or misspelled) host, you will see:

```
Transferring files to WFM: [ftp\reports\031022.1115, ftp\reports\031022.1130]
Error occured while uploading reports to WFM FTP server
com.ariasolutions.iconnect.historical.transfer.FtpException: Error occured while attempting to send files
<badHostNameHere>
    at com.ariasolutions.iconnect.historical.transfer.SftpTransferAgent.transferFiles(Sft-
pTransferAgent.java:309) ~[main/:?]
    at com.ariasolutions.iconnect.historical.transfer.TransferManager.transferFiles(Trans-
ferManager.java:354) [main/:?]
    at com.ariasolutions.iconnect.historical.transfer.TransferManager$1.run(TransferManager.java:207)
[main/:?]
```

```
    at com.ariasolutions.utils.concurrent.TaskFuture.run(TaskFuture.java:80) [main/:?]
    at com.ariasolutions.utils.concurrent.TaskQueue.run(TaskQueue.java:41) [main/:?]
    at java.lang.Thread.run(Thread.java:834) [?:?]
Caused by: java.net.UnknownHostException: <badHostNameHere>
    at java.net.AbstractPlainSocketImpl.connect(AbstractPlainSocketImpl.java:220) ~[?:?]
    at java.net.SocksSocketImpl.connect(SocksSocketImpl.java:403) ~[?:?]
    at java.net.Socket.connect(Socket.java:608) ~[?:?]
    at net.schmizz.sshj.SocketClient.connect(SocketClient.java:138) ~[sshj-0.32.0.jar:?]
    at net.schmizz.sshj.SocketClient.connect(SocketClient.java:128) ~[sshj-0.32.0.jar:?]
    at com.ariasolutions.iconnect.historical.transfer.SftpTransferAgent.transferFiles(Sft-
pTransferAgent.java:263) ~[main/:?]
    ... 5 more
Of 2 files prepared, 2 files were not sent to the WFM server
0 of 2 files sent to WFM FTP server
FTP task completed
```

B.6 Missing remote directory

A missing remote directory (**Note:** directories are not created, they must already exist) will appear to be successful, but the reports will not be in the destination folder as expected. In this case, the destination directory 'wfmReports_not' does not exist. Instead of finding the reports in that sub-directory, the parent directory now has a file named 'wfmReports_not', which contains the contents of the last report:

```
Transferring files to WFM: [ftp\reports\031022.1130, ftp\reports\031022.1145]
<installDirectory>\ftp\reports\031022.1130 successfully transferred
<installDirectory>\ftp\reports\031022.1145 successfully transferred
2 of 2 files sent to WFM FTP server
Backing up file '<installDirectory>\ftp\reports\031022.1130 to <installDir-
ectory>\ftp\reportsBackup\031022.1130
Backing up file '<installDirectory>\ftp\reports\031022.1145 to <installDir-
ectory>\ftp\reportsBackup\031022.1145
FTP task completed
```

B.7 Username misspelled / incorrect password

In the following example, the username was misspelled in the options, but the same results would be seen for an incorrect password:

```
Transferring files to WFM: [ftp\reports\031022.1215]
Error occured while uploading reports to WFM FTP server
com.ariasolutions.iconnect.historical.transfer.FtpException: Error occured while attempting to send files
Exhausted available authentication methods
    at com.ariasolutions.iconnect.historical.transfer.SftpTransferAgent.transferFiles(Sft-
pTransferAgent.java:309) ~[main/:?]
    at com.ariasolutions.iconnect.historical.transfer.TransferManager.transferFiles(Trans-
ferManager.java:354) [main/:?]
    at com.ariasolutions.iconnect.historical.transfer.TransferManager$1.run(TransferManager.java:207)
[main/:?]
    at com.ariasolutions.utils.concurrent.TaskFuture.run(TaskFuture.java:80) [main/:?]
```

```
    at com.ariasolutions.utils.concurrent.TaskQueue.run(TaskQueue.java:41) [main/:?]
    at java.lang.Thread.run(Thread.java:834) [?:?]
Caused by: net.schmizz.sshj.userauth.UserAuthException: Exhausted available authentication methods
    at net.schmizz.sshj.SSHClient.auth(SSHClient.java:227) ~[sshj-0.32.0.jar:?]
    at net.schmizz.sshj.SSHClient.authPublickey(SSHClient.java:342) ~[sshj-0.32.0.jar:?]
    at net.schmizz.sshj.SSHClient.authPublickey(SSHClient.java:360) ~[sshj-0.32.0.jar:?]
    at com.ariasolutions.iconnect.historical.transfer.SftpTransferAgent.transferFiles(Sft-
pTransferAgent.java:269) ~[main/:?]
    ... 5 more
Of 1 files prepared, 1 files were not sent to the WFM server
0 of 1 files sent to WFM FTP server
FTP task completed
```

C Appendix: Managing Windows Service

C.1 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 Gplus 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.

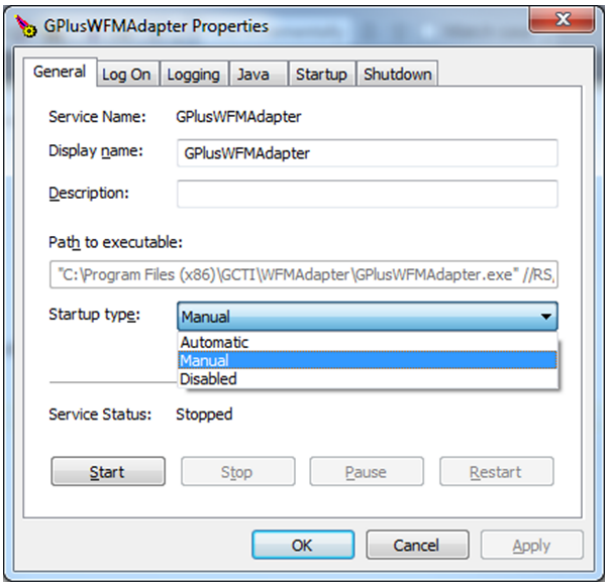
General Tab

Startup type:

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

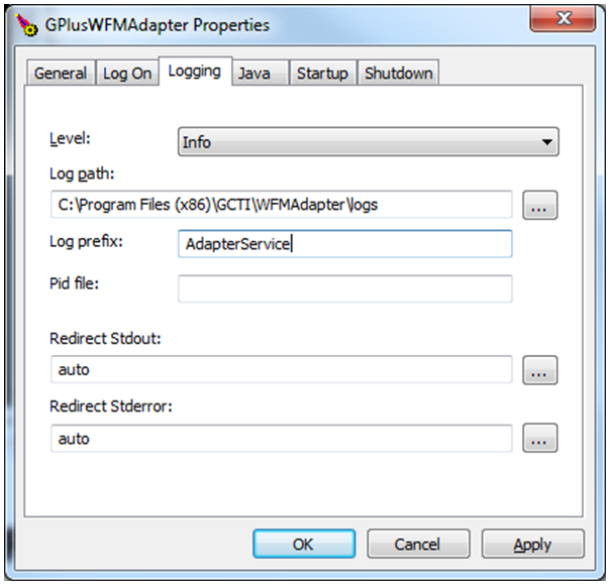
Start, Stop and Restart Buttons

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



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:

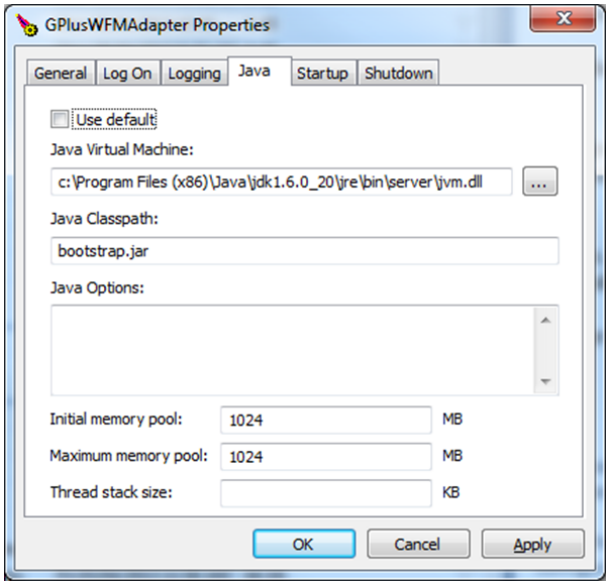
The default is the Adapter's `logs` folder but this parameter can be set to another folder if required.

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.

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: *startup.bat and svcinstall.bat*.

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 = `4096 MB`

Maximum memory pool:

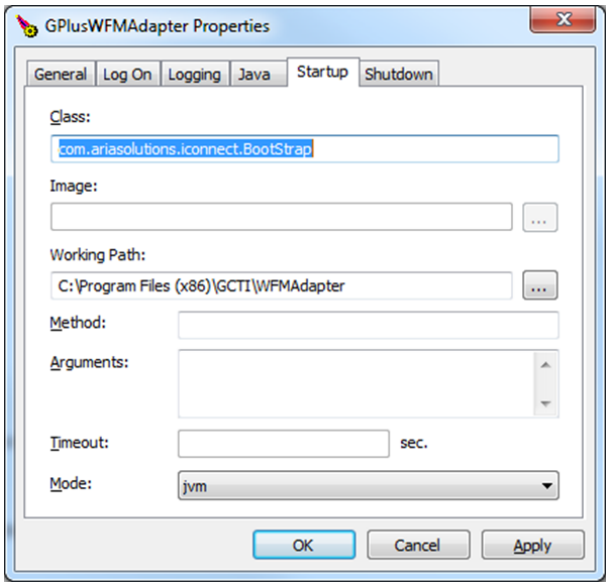
This parameter specifies the maximum amount of system memory that will be requested by the JVM.

Default = `4096 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.

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`.

D Appendix: Log Files

Files

Depending upon the logging configuration discussed in *Event and Error Logs*, 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 `log4j2.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.

D.1 Files

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
- `log4j2.xml.example` – logging configuration
- `license.txt` – license file

D.2 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.

D.3 Logging Threshold and Messages

The *logThreshold* 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 estab-
lished 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.

D.4 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 *useLocalTime* configuration option.

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

D.5 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.

D.6 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': QueuedIn-
teraction [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': QueuedIn-
teraction [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`.

E Appendix: Windows Installation Script

E.1 Files

The `windows_install.bat` script file can carry out all of the tasks required to install the Gplus Adapter instance as a Windows Service. This file uses the script files contained in the `scripts\windows` folder and those script files should not be altered.

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 specific WFM Adapter vendor, 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 script also displays the relevant contents of the license file.

E.2 Invoking the Installation Script

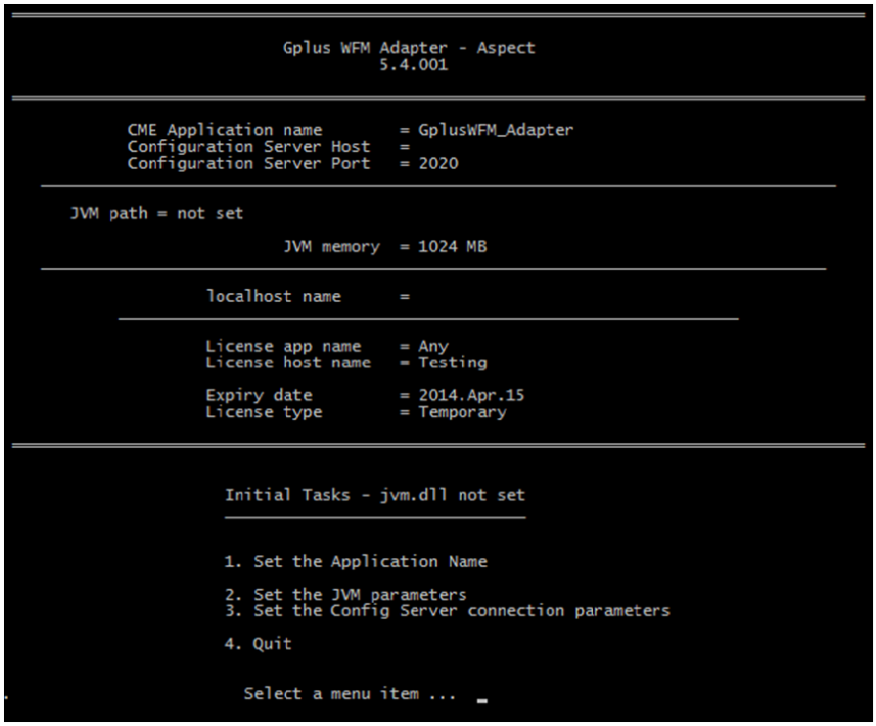
It is important that the script execute with Administrator privileges so that the Windows service can be installed and the properties files modified. This can be accomplished by right clicking on the `windows_install.bat` file and selecting *Run as administrator*.

E.3 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.

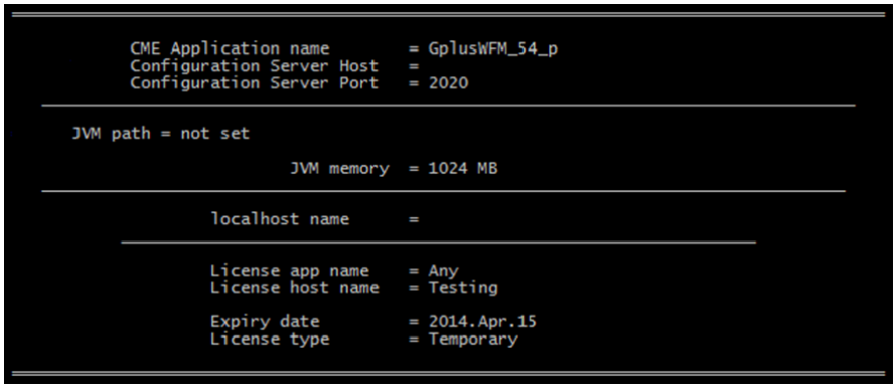


The screenshot above shows the default settings from the properties files as well as the parameters for the temporary license provided with the Adapter release.

Set the Application name

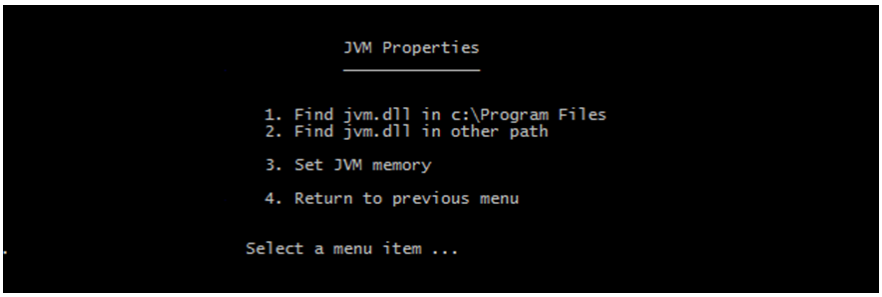
The Application name should be set first. This name must match the name of the Gplus Adapter's Application object in the Configuration Server. The name will be used for the name of the installed service and the renamed `prunsrv.exe` (Java service wrapper) that will be copied to the Adapter's main folder.

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



Set the JVM parameters

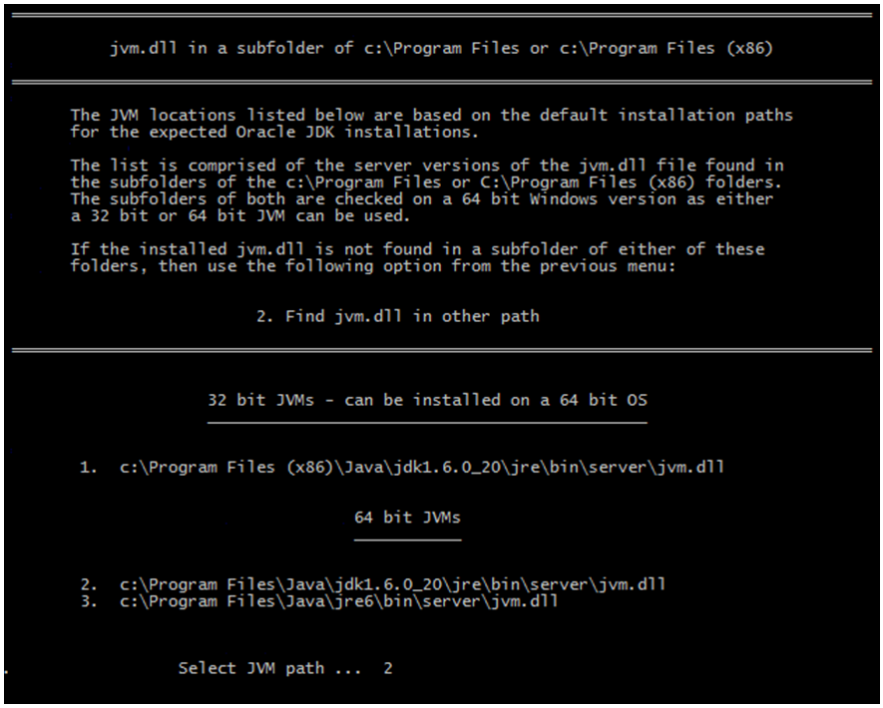
This menu item brings up a second menu that allows the Java Runtime Environment to be configured. Once this has been completed, the Adapter can be installed as a service.



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.



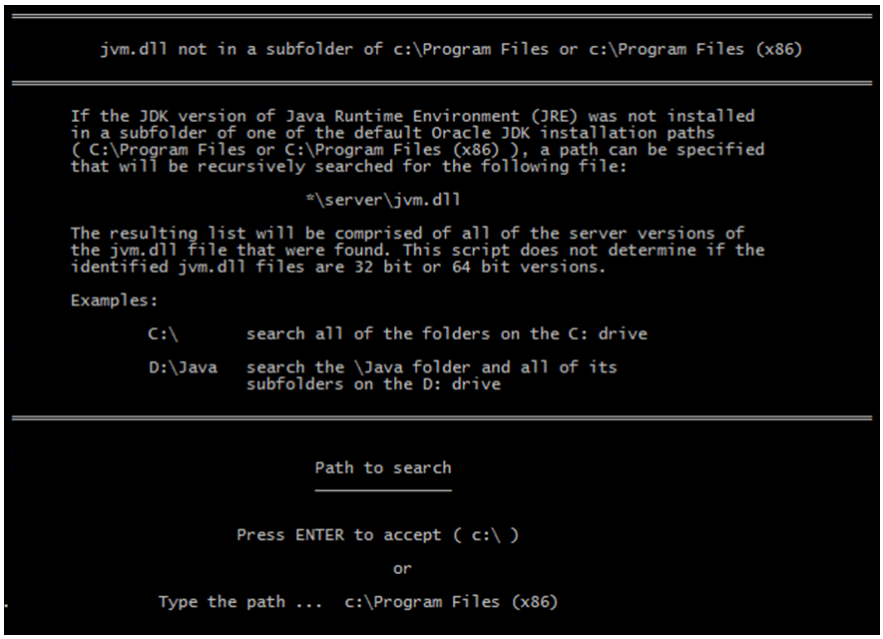
The change is immediately written to the `config\jvm.properties` file and the status section reflects the specification of the 64 bit JVM as shown below.

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.

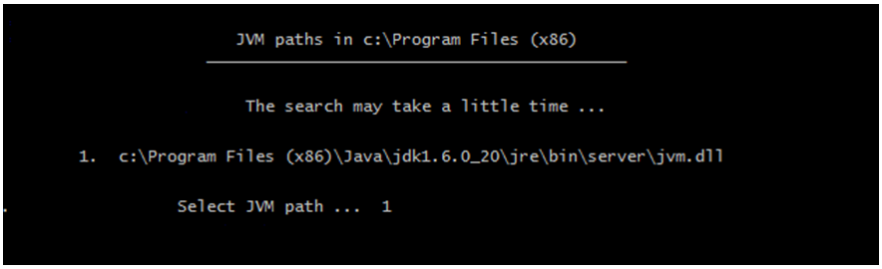
First, a new parameter, `JVM type`, has been added. This indicates which of the prunsrv.exe service wrappers will be used. In the example above, the 64 bit non-Itanium™ version will be used.

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

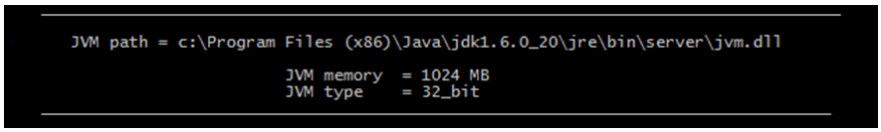


If the Java instance has not been installed in a subfolder of either `c:\Program Files` or `c:\Program Files (x86)`, the script will search for the `jvm.dll` file in other locations.

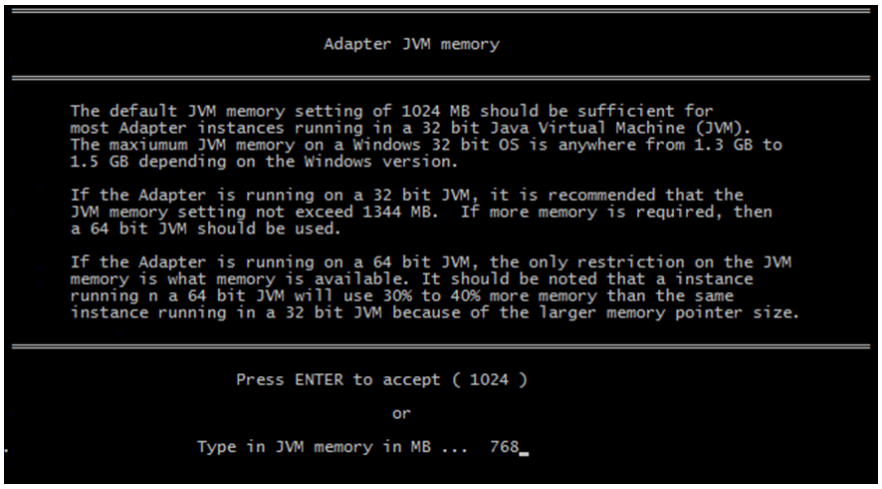


The result will be a list of the paths to any found `jvm.dll` files but there will no indication as to whether the files are 32 bit or 64 bit versions.

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`.

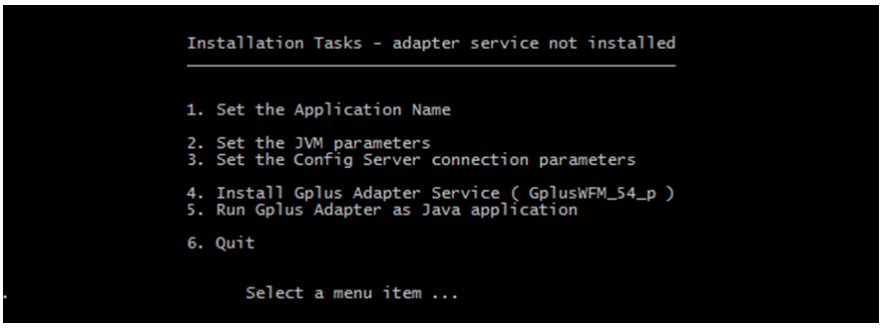


Set JVM memory



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.

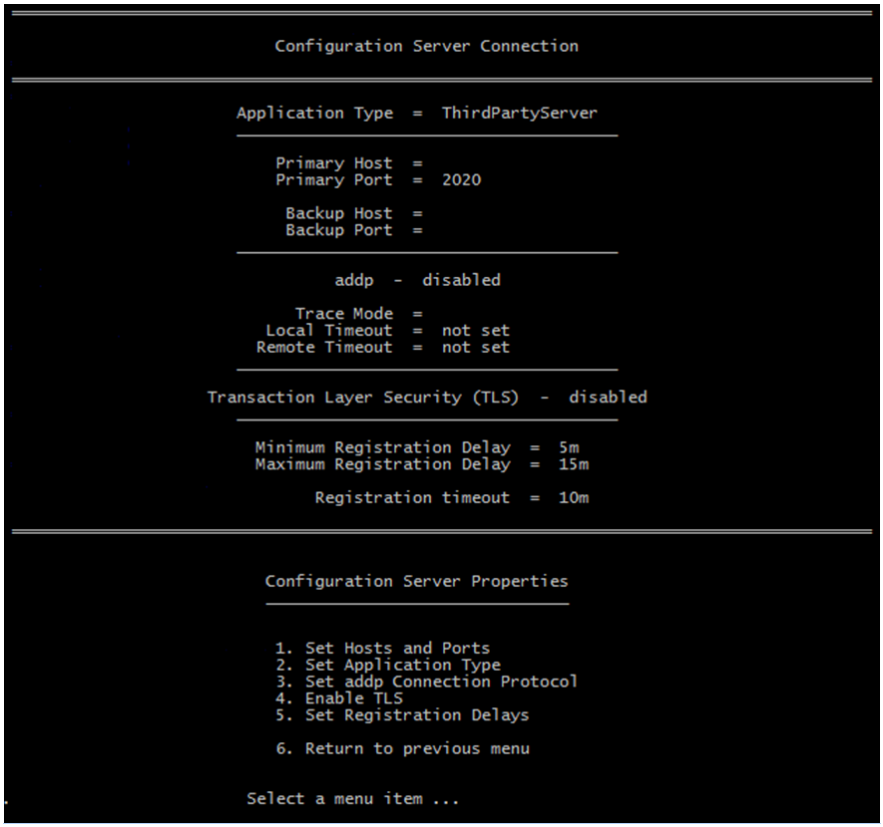
E.4 Installation Tasks



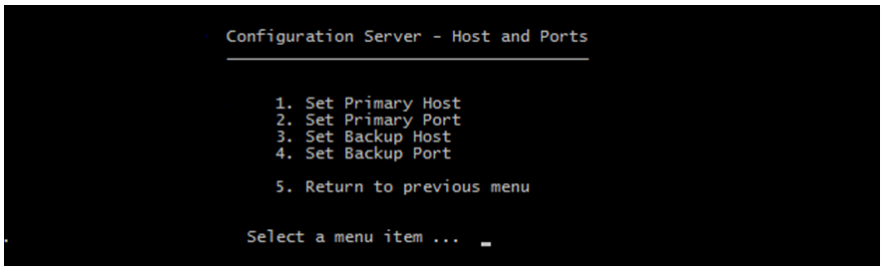
Once the path to the jvm.dll file is set, the Gplus Adapter can be installed as a Windows Service. The main menu reflects this change in status with the addition of two more menu options. It is possible to install the service with no issues but starting that service or running the Adapter as a console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

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.

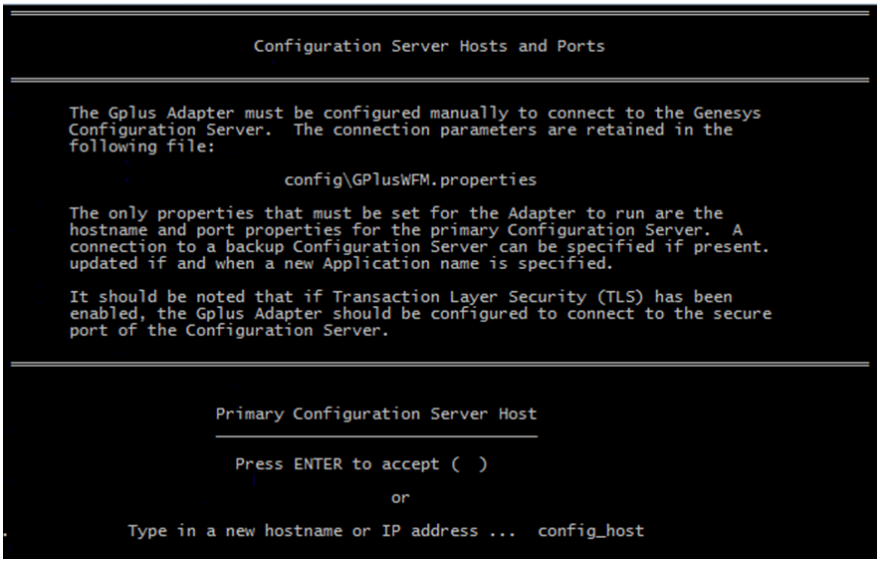


Hosts and ports



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

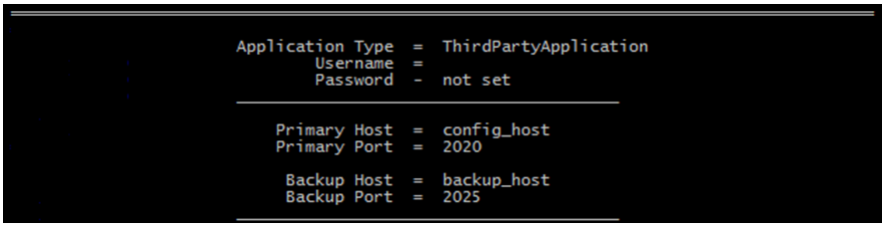
hostname or the IP address. If TLS has been enabled, the configured port should be the Configuration Server's secure port rather than the default.

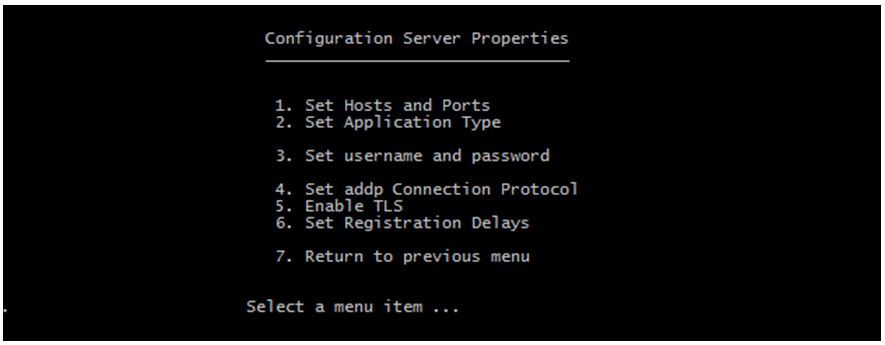


Set Application type

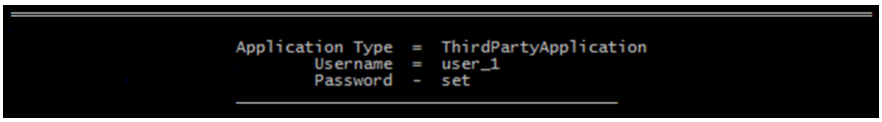
The Adapter's Application type affects the parameters that will be sent to the Configuration Server as part of the Adapter's registration request. A `ThirdPartyServer` type does not provide a `username` or `password` but is only allowed one connection. The alternative, `ThirdPartyApplication`, requires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

ThirdPartyApplication





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.



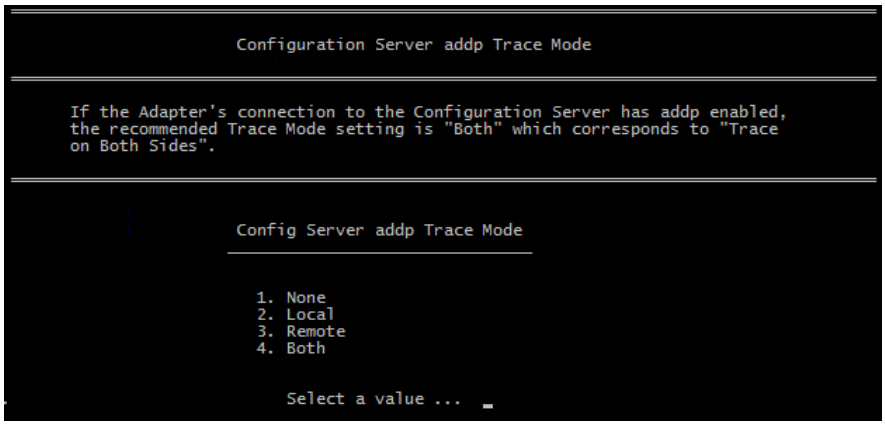
When the password is set, that string is encrypted and stored in the `config/enc` file. If that file is present, the script indicates that the password has been *set* but does not display it.

Set addp Connection Protocol

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

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

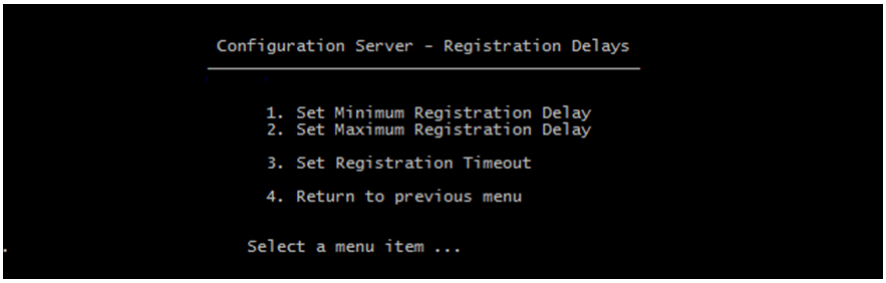


The script lists the allowed trace mode settings. It is recommended that the trace mode be set *Both*

Enable/Disable TLS

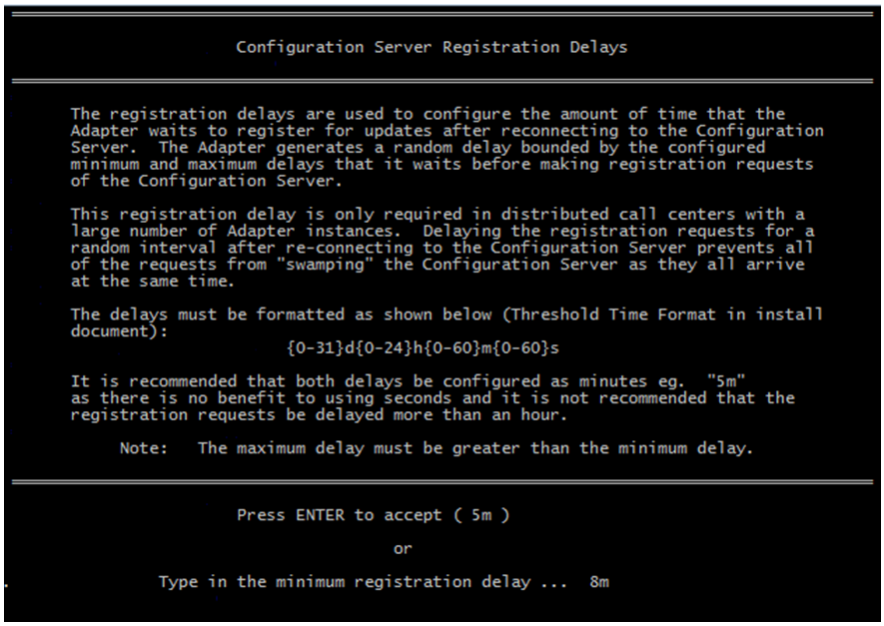
TLS (Transaction Layer Security) can be set on the Configuration Server connection but the majority of the configuration work will be on the server side. The extent of the changes on the Adapter side is to enable TLS and set the connection to point to the Configuration Server's secure port.

Set Registration delays

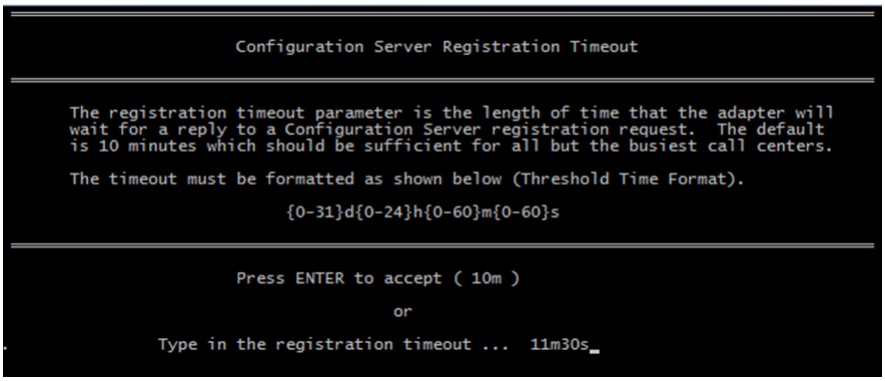


This menu section may be necessary in very busy call centers where there are a number of applications connected to the Configuration Server. In that case, the response to requests may be delayed and it may be necessary to delay those requests when the Configuration Server is restarting after a mishap.

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.



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 before generating a new request.



Installing the Adapter as a Service

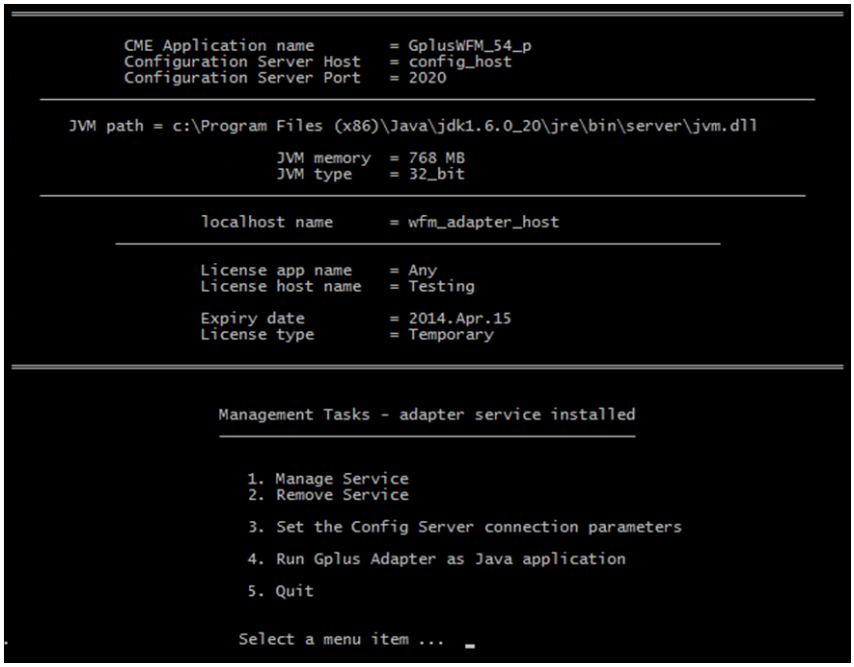
When the option to install the Adapter as a Service is selected, two actions occur:

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.

E.5 Management Tasks

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



Manage Service

Selecting the *Manage Service* option invokes the `prunmgr.exe` application that is detailed in *prunmgr Application*. 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 Gplus 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.

E.6 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.

F Appendix: AWS Credentials – S3 Storage Bucket

The following discussion is based on the information provided on the Amazon Web Services (AWS) web page, Working with AWS Credentials. This discussion assumes that Genesys has provided the credentials for the AWS Identity and Access Management (IAM) User that will be used to access the Simple Storage Service (S3) storage bucket.

F.1 IAM User versus *root* User

The root user associated with the AWS account should not be used to access the S3 storage bucket. AWS provides Identity and Access Management (IAM) User accounts associated with that root user. These accounts can be created with limited permissions that allow transferring files to the bucket and little else.

F.2 IAM User Credentials

An IAM user created with Programmatic access is identified with an Access Key ID and a Secret Access Key rather than a password. That user does not have access to the AWS Management Console.

While there are a few different methods for supplying the AWS credentials when accessing the storage bucket, the Gplus Adapter code incorporates the Default Credential Provider Chain class that steps through the following locations to find credentials used to access the S3 storage bucket:

1. Environment variables
2. Java system properties
3. The *default credential profiles file* – recommended

This Appendix will discuss the third option of using a credentials file. The other methods are also detailed on the web page mentioned above but will not be discussed here.

F.3 Location of Credentials File

The default path and name for the credentials file is as follows with `'~'` indicating the active user's home directory:

```
~/.aws/credentials
```

This can be changed by modifying or adding the `AWS_CREDENTIAL_PROFILES_FILE` environment variable as below:

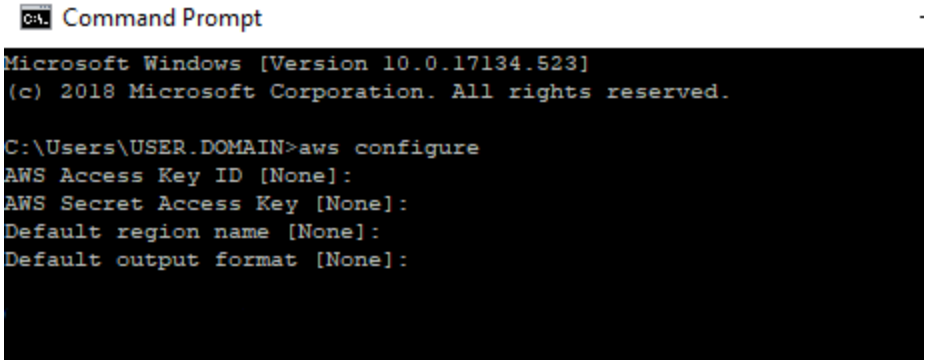
- Linux – `export AWS_CREDENTIAL_PROFILES_FILE=/path/credentials`
- Windows – `set AWS_CREDENTIAL_PROFILES_FILE=/path/credentials`

F.4 Creating the Credentials File

The credentials file is a text file that can be created with a text editor. As mentioned above, the default name of the file is `credentials` and the default location is the user's `.aws` folder.

```
[default]
aws_access_key_id={ACCESS_KEY_ID}
aws_secret_access_key={SECRET_ACCESS_KEY}
```

The credentials file can also be created using the AWS Command Line Interface (AWS CLI) as part of the configuration process for that application. The following shows a screenshot when the CLI is invoked with the `aws` configure command.



The result is two files in the `~/.aws` folder: config and credentials. The config file contains the default region and output format while the credentials file is as described above.