



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

# Digital Messaging Server Guide

Messaging Applications 9.1.0

11/14/2024

# Table of Contents

<b>Digital Messaging Server</b>	<b>3</b>
<b>Deploy Digital Messaging Server</b>	<b>5</b>
Connection Table	15
<b>Deploy DMS with a Facebook Channel</b>	<b>16</b>
Facebook Interaction Attributes	18
<b>Deploy DMS with a Twitter Channel</b>	<b>27</b>
Twitter Interaction Attributes	28
<b>Custom Media Channel Driver</b>	<b>34</b>
Security	35
<b>Other DMS Configuration</b>	<b>38</b>
<b>Outbound interaction from Workflow</b>	<b>42</b>
DMS Scalability	48

# Digital Messaging Server

Digital Messaging Server (DMS) is the eServices component that interfaces with social media sites to bring interactions into the Genesys system.

This document describes what you must do to deploy and use DMS.

You can also use DMS with a [Custom Media Channel Driver](#).

<p><b>Deployment</b></p> <p>This section describes how to deploy DMS and its channels.</p> <hr/> <p><a href="#">Deploy DMS</a></p> <p><a href="#">Deploy with a Facebook channel</a></p> <p><a href="#">Deploy with a Twitter channel</a></p> <p><a href="#">Deploy with a custom media channel</a></p>	<p><b>Configuration Options</b></p> <p>This chapter documents the configuration options used by DMS.</p> <hr/> <p><a href="#">DMS Configuration Options</a></p>
---	---

## Support for InviteToChat ESP request

Starting from the DMS version 9.1.003.10, the InviteToChat method is supported. This method allows initiation of outbound chat sessions from a strategy and sending WhatsApp notification templates to customers. It is an invitation for a customer who is using voice IVR interaction to switch to Chat or Messaging and continue the conversation. For more information, see [Outbound interaction from Workflow](#).

## Support for High Availability by default

### Important

- Starting from the DMS version 9.1.008.08, Chat Server basic protocol is not supported. When upgrading to DMS 9.1.008.08, follow the upgrade procedure given in [Upgrade Notes](#).

Starting from the DMS version 9.1.006.09, Chat Server flex protocol is enabled in DMS configuration by default. This provides High Availability support with multiple Chat Servers. It necessarily requires

High Availability configuration for DMS and Chat Server(s). For information on configuring DMS High Availability, see [DMS High Availability configuration](#).

# Deploy Digital Messaging Server

The following steps describe how to deploy Digital Messaging Server (DMS).

## Prerequisites

- Genesys components
  - Message Server 8.5.x
  - Chat Server 8.5.309.17 or higher
  - Universal Contact Server 8.5.300.39 or higher
  - Interaction Server 8.5.300.07 or higher
- Third-party components
  - One of the following Java components:
    - Oracle Java 8 Developer's Kit (JDK)
    - OpenJDK 8
    - OpenJDK 11
    - OpenJDK 17

### Important

**vcruntime140\_1.dll** file is required to ensure that DMS starts correctly with OpenJDK 17 on Windows.

- High loaded DMS
  - 16 GB RAM
- For Unix systems: Increase the kernel parameter `/proc/sys/vm/max_map_count` to 131060 to accommodate a higher number of parallel chat sessions,

## Creating a DMS Application object

Create an Application object for DMS if it does not already exist.

1. Import the DMS Application Template from the CD.

2. Create a new Application object based on the template.
  - a. Open the **Properties** dialog box of the Application object.
  - b. On the **Server Info** tab:
    - i. In the **Host** box, enter the name of the desired host.
    - ii. In the **Port** box, enter the port DMS will use.
3. On the **Start Info** tab, enter some characters in the **Working Directory**, **Command Line**, and **Command Line Arguments** fields. These characters will be overwritten with the correct values during the installation, but they cannot be left blank at this point.
4. On the **Connections** tab, add a connection to [all components mentioned](#).
5. If this is for a multi-tenant environment, add the tenant(s) on the **Tenants** tab.
6. Open your Interaction Server Application and add a connection to DMS. Specify the Connection Protocol as either **simple** or **addp**.

## Installing DMS

See the [Supported Operating Environment Reference](#) for prerequisites before you install DMS. For information on compatibility between DMS and various Genesys drivers, see [Digital Messaging Server and Driver Compatibility](#).

### Windows

1. Locate and run the **Setup.exe** file for DMS.
2. Enter the login information for your Configuration Server:
  - Host
  - Port
  - User
  - Password
3. Select the checkbox if you want to use the Client Side Port, and then click **Next**.
4. Select the appropriate DMS Application object from the list.
5. Choose a destination for the installation.
6. Select the Java instance to use with DMS, then click **Next**.
7. Specify a log folder, then click **Next**.
8. Click **Install**.

### Linux

1. Locate the **install.sh** file for DMS.

2. Use the command `>install.sh` to start the installation script.
3. On request, enter the absolute path for the directory of the Java executable.
4. Press Enter to confirm the host name for the installation.
5. Enter the login information for your Configuration Server:
  - Host
  - Port
  - User
  - Password
6. From the list of applications, select one and enter its number in the list.
7. Press Enter to confirm the suggested destination directory, or choose another one.
8. Answer other questions, if required.

## DMS High Availability configuration

### Important

- Starting from the DMS version 9.1.003.10, High Availability is supported for WhatsApp and Apple Business Chat channels.
- Starting from the DMS version 9.1.006.09, DMS by default supports High Availability for WhatsApp, and Apple Business Chat channels, and it necessarily requires High Availability configuration.
- Starting from the DMS version 9.1.008.08 and Genesys Driver for SMS and MMS 9.0.003.03, High Availability is supported for SMS messaging with media type SMS Session. This feature requires installation of Workspace Desktop Edition 8.5.153.05.

Digital Messaging Server supports High Availability in *Warm Standby* mode. This section describes how to configure DMS to achieve High Availability (HA) on both Windows and Linux operating systems.

### Configuring DMS Application objects

1. On the **Server Info** tab of the Primary DMS application, specify the name of the Backup DMS Server.
2. On the **Server Info** tab of the Primary DMS application, specify *Warm Standby* as the Redundancy Type.
3. On the Interaction Server's **Connections** tab, configure a connection to the Primary DMS.

## Setting JVM properties

### Important

Starting from DMS version 9.1.006.09, this property is true by default after DMS installation.

Set the following property in the **JavaServerStarter.ini** file (for Windows) or **dmserver.sh** file (for Unix): `-Dgenesys.mcr.stdserverex.flexchatprotocol=true`

## Configuring interactions

### SQL Scripts

You must run the provided SQL script to interact with Interaction Server's database. You can find the following SQL scripts in the **ha\_sql\_scripts** folder within the DMS installation folder:

- **ha\_support\_mssql\_nvarchar.sql**
- **ha\_support\_mssql\_varchar.sql**

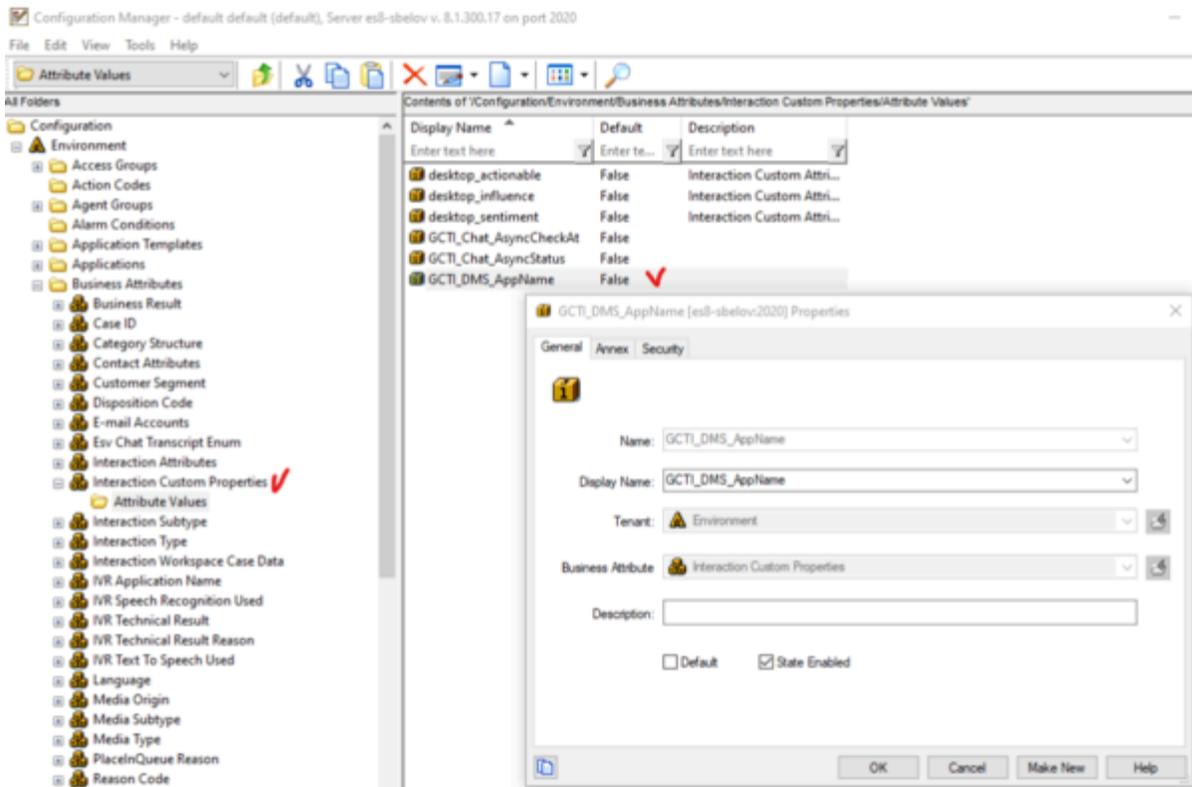
These scripts are prepared for Microsoft SQL DBMS. You can run one of these scripts based on your environment to create nvarchar or varchar string field types.

### Interaction Custom Properties

Create and configure **GCTI\_DMS\_AppName** custom property in the Configuration Server.

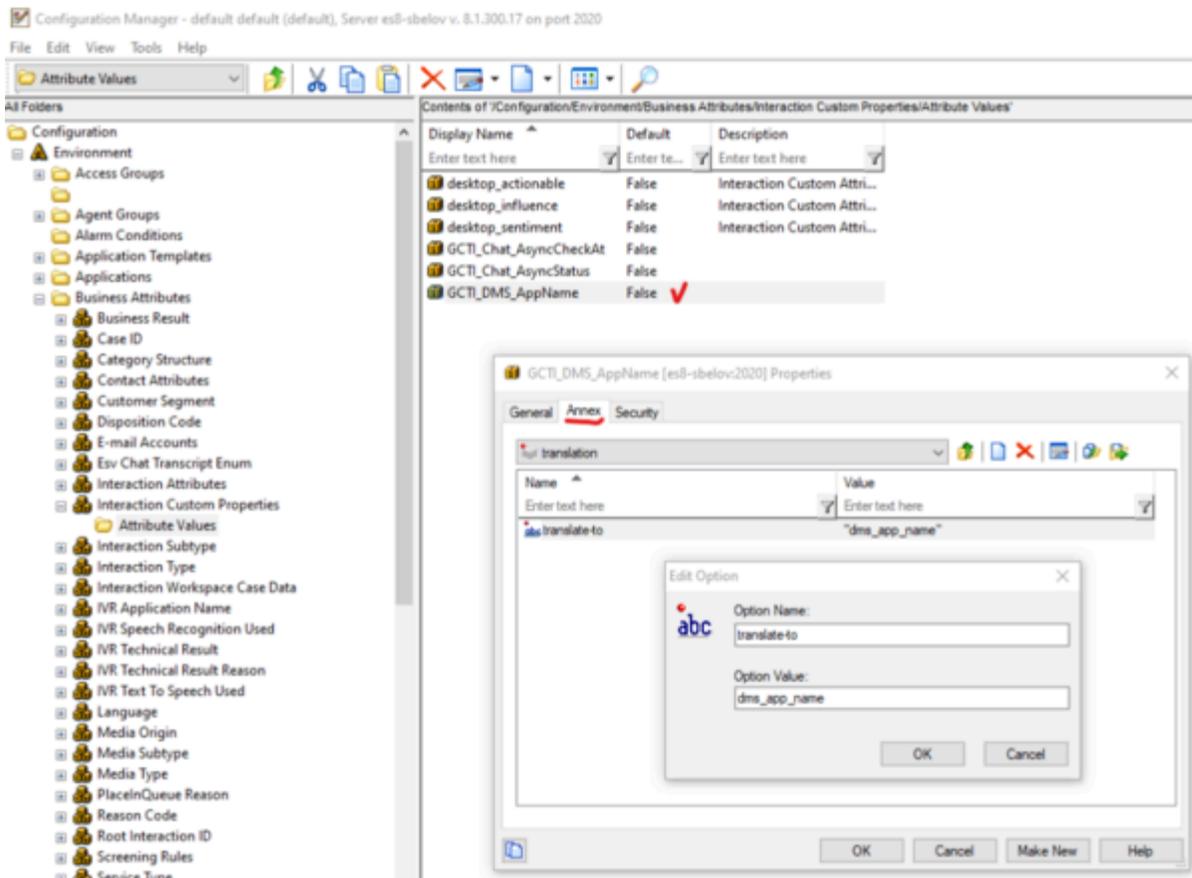
1. Create a custom property, **GCTI\_DMS\_AppName** as shown in the following image:

## Deploy Digital Messaging Server



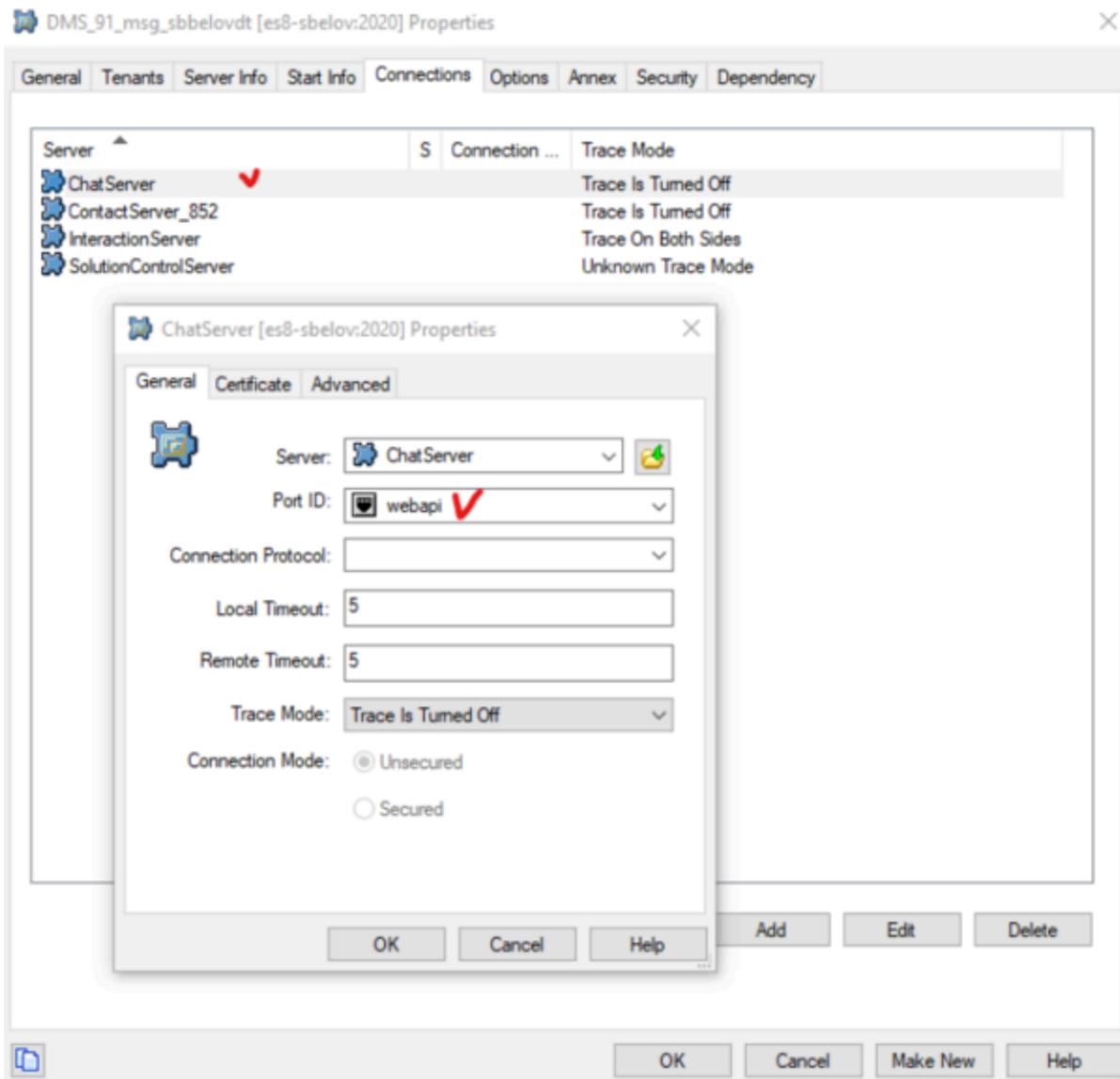
2. For the created custom property in the **translation** section of Annex, specify the value as `dms_app_name` for the **translate-to** option.

## Deploy Digital Messaging Server



## Chat Server Connections

You can establish connections to multiple primary Chat Servers using the **Connections** tab in DMS. The Chat Servers listed in the **Connections** tab must not use backup support.



### Tip

You must configure Chat Server HA using the instructions provided in the [Deploying High-Availability Chat Server](#) guide.

## DMS Scalability

For information on configurations related to DMS scalability, see [DMS scalability](#).

### Next Steps

- Configure the options as necessary. Refer to the [Digital Messaging Server](#) page in the [eServices Options Reference](#).
- Continue deployment by:
  - Adding a channel, such as [Facebook](#), a [Twitter](#), or a [Custom Media Channel](#).
  - Adding a messaging channel, such as [Apple Business Chat](#), [SMS/MMS](#), or [WhatsApp](#).

#### Important

If desired, you can upload the DMS installation package to Genesys Administrator Extension for storage purposes or to share the package with other machines in your environment. Refer to the [GAX documentation](#) for additional information.

### Uninstalling or upgrading DMS

#### Uninstalling DMS

#### Important

- If you have also installed Genesys drivers for DMS (for example, Apple Business Chat), you must uninstall these drivers separately. These drivers are not uninstalled automatically if DMS is uninstalled.
- You cannot uninstall DMS and then install a new instance of DMS at the same location and use the old driver instances. You must also reinstall drivers that you want to use with the new DMS instance, even if the driver versions have not changed.
- You must delete the DMS Application object separately.

#### Windows

To uninstall using the wizard:

1. Run the **setup.exe** file for DMS that is located on the product CD or in the installation package.
2. Select **Maintenance of existing installation**.
3. Select the DMS instance that you want to remove.
4. Select **Next**. Wait for several seconds until the Genesys Installation Wizard window appears on your desktop.

5. Select **Remove**.
6. Select **Next**.
7. Select **Yes** to confirm the operation.
8. Choose when to restart your computer.
9. Select **Finish**.

To uninstall using the Control Panel:

1. Open the Control Panel.
2. Select the DMS installation that you want to remove.
3. Select **Uninstall** and then **Next**. Wait for several seconds until the Genesys Installation Wizard window appears on your desktop
4. Select **Remove**.
5. Select **Next**.
6. Select **Yes** to confirm the operation.
7. Choose when to restart your computer.
8. Select **Finish**.

### Linux

Delete the DMS files directly by performing the following command: `rm -rf <DMS_FOLDER>`

## Upgrading DMS

### Important

- If you have also installed Genesys drivers for DMS (for example, Apple Business Chat), you must update these drivers separately. These drivers are not updated automatically when DMS is updated.
- You cannot upgrade DMS and use the old driver instances. You must also reinstall drivers that you want to use with the upgraded DMS instance, even if the driver versions have not changed.
- You must update the DMS Application separately, if needed.
- If upgrading to the DMS version 9.1.006.09 or later, DMS High Availability configuration is required. For more information, see [DMS High Availability configuration](#).

### Windows: Upgrading via the wizard

1. Run the **setup.exe** file for DMS that is located on the product CD or in the installation package.

2. Select **Install new instance of the application** and then select **Next**.
3. Enter your Configuration Server parameters.
4. Select **Next**.
5. Select the checkbox if you want to use the Client Side Port, and then select **Next**.
6. Select the DMS application that you want to upgrade and then select **Next**.
7. Select **Browse** and select the folder in which the existing DMS instance is installed, then select **Next**.
8. Select the Java instance to use with DMS, then select **Next**.
9. Specify a log folder, then select **Next**.
10. Select **Install**.

### Linux: Upgrading via the install.sh script

1. Run the **./install.sh** file for DMS that is located on the product CD or in the installation package.
2. Specify the path to the Java executable folder.
3. Specify the hostname.
4. Specify the hostname of the Primary Configuration Server.
5. Specify the Primary Configuration Server port, username, and password.
6. Select the DMS application that you want to upgrade.
7. Specify the existing DMS installation path.
8. Select how to handle existing data in the DMS folder. For example, 2 only overwrites those files that are part of the installation package.

# Connection Table

The following table lists Digital Messaging Server connections that are set in the Configuration Layer for different drivers. The connection to Message Server is optional and is only required if you are planning to use it for the application log output.

## Important

- It is possible to have **multiple connections** to some components. In the table, these components are marked with a \*.
- For information on connections related to DMS scalability, see **DMS scalability**.

Driver installed with DMS	Connects to
Apple Business Chat Driver	Message Server Chat Server* Interaction Server Solution Control Server Universal Contact Server
Facebook Driver	Message Server Chat Server* Interaction Server Solution Control Server
Twitter Driver	Message Server Chat Server* Interaction Server Solution Control Server
Genesys Driver for use with Genesys Hub	Message Server Chat Server* Interaction Server Solution Control Server Universal Contact Server
Genesys Driver for SMS and MMS	Message Server Chat Server* Interaction Server Solution Control Server Universal Contact Server

---

# Deploy DMS with a Facebook Channel

For a Facebook channel, you need two installation packages:

- Digital Messaging Server
- Genesys Cloud API Driver for Facebook.

The driver adds Facebook-specific features to DMS and does not require its own Application object in the Configuration Server database.

You can also create a [Custom Media Channel Driver](#).

## Create the Facebook Channel

1. [Deploy DMS](#).
2. Run the installation for Genesys Driver for Use with Facebook, selecting the desired DMS object.
3. Locate the **driver-for-facebook-options.cfg** configuration file in the \<**Digital Messaging Server application**>\media-channel-drivers\channel-facebook directory.
4. In Configuration Manager, open your Digital Messaging Server Application, go to the **Options** tab, and import **driver-for-facebook-options.cfg**.

## Configure the Options

Refer to the [Genesys Cloud API Driver for Facebook](#) page in the [eServices Options Reference](#).

## Interaction Attributes

The driver provides a number of interaction attributes. A reference listing is available on a [separate page](#).

## Sample Business Processes and Database Scripts

You must:

- Install the [sample Business Processes for Social Media](#).
- Run the database upgrade scripts installed with the sample Business Process. These scripts add required fields to the Interaction Server database.

## Requirement for Posting and Commenting

In order for the Genesys Desktop Plugin to display the **Delete**, **Delete Post**, **Comment**, and **Share** buttons, it checks the **\_facebookCanComment** and **\_facebookCanDeleteComments** keys in the interaction's attached data.

This means that, if the Facebook account holder did not set permissions to let everybody post/comment on the account holder's wall, the agent must *like* the poster's page or add him as a friend to be able to post on the wall.

The same holds for an autoresponse if it is included into the strategy: if the agent does not *like* the poster, or if he or she did not set permissions to let everybody post/comment on the wall, the post/comment will fail and a **Facebook POST failed** exception is written in the log.

# Facebook Interaction Attributes

## Basic Attributes

Genesys Driver for use with Facebook adds the following attributes to an interaction's user data.

Key Name	Value Type	Description
_facebookDriverVersion	UTF string	Driver version
_facebookItxType	Int	<ul style="list-style-type: none"> <li>1—Facebook Post and single Comment (if the interaction is a Reply, the parent Comment is also included, as well as all elements in <a href="#">Comment Attributes</a> below).</li> <li>2—Facebook Post and multiple Comments</li> </ul>
_umsChannel	UTF String	Name of the channel that includes the monitor that submitted the interaction
_umsChannelMonitor	UTF String	Name of the monitor that submitted the interaction
_umsChannelInfo	Key-value list	Present if the driver was able to get Publisher information. See <a href="#">_umsChannelInfo Attributes</a> below.
_facebookPostId	UTF String	Post ID
_facebookContentCreatedOnlyByAdmin		0 or 1
_facebookCanComment	Int	Whether the Comments are (0) or are not (1) allowed on the object.
_facebookNComments	UTF string	Number of Comments for this Post
_facebookShareCount	UTF string	Number of times the Post was shared
_facebookPostCanBeDeleted	UTF string	true or false
_facebookMediaAttachmentsCount	Int	Number of files attached to the Post
Subject	UTF string	Subject of the Post, usually taken from Post text itself. If the message in the Post is empty, Subject is taken from the name or description of the first

Key Name	Value Type	Description
		attached file. If that information is missing, Subject is equal to <code>_facebookPostId</code> .
<code>_facebookMessageText</code>	UTF string	Message in the Post
<code>_facebookNLikes</code>	UTF string	Number of Likes for the Post
<code>_facebookUpdatedTime</code>	UTF string	The time, expressed as a UNIX timestamp, that the Post was last updated, which occurs when a User comments on the Post.
<code>_facebookCreatedTime</code>	UTF string	The time the Post was published, expressed as UNIX timestamp
<code>_facebookSourceId</code>	UTF string	Facebook ID of the object where this Post was published
<code>_facebookSourceObjectType</code>	Int	Type of the Facebook object that published the Post: <ul style="list-style-type: none"> <li>• 0—User</li> <li>• 1—Page</li> <li>• 2—Event</li> <li>• 3—Group</li> <li>• 4—Application</li> </ul>
<code>_facebookActorId</code>	UTF string	Facebook ID of the object that published this Post
<code>_facebookActorObjectType</code>	Int	Type of the Facebook object that published the Post: <ul style="list-style-type: none"> <li>• 0—User</li> <li>• 1—Page</li> <li>• 2—Event</li> <li>• 3—Group</li> <li>• 4—Application</li> </ul>
<code>_facebookActorName</code>	UTF string	Name of the Page or User that published the Post. If it is a User ( <code>_facebookActorIdType = 0</code> ), then in addition to the name, the attributes listed in <a href="#">User Attributes</a> below are also included.
<code>_facebookComments</code>	Key-value list	The list of key-value pairs provided in <a href="#">facebookComments Attributes</a> below. The key for each list equals the value of <code>_facebookCommentId</code> .

Key Name	Value Type	Description
_facebookXML	UTF string	An XML string which includes information about all objects that participated in or were mentioned in the Post/Comments/Replies

## Private Message Attributes

Key Name	Value Type	Description
_facebookDriverVersion	UTF String	Driver version
_facebookItxType	int	10 (Private Message)
_umsChannel	UTF String	Configured Facebook channel name
_umsChannelMonitor	UTF String	Configured Facebook monitor name
_umsChatRequired	UTF String	Always true
_umsMediaType	String	Value of inbound-media option for the monitor. To work with Workspace Desktop Edition the value must be facebooksession.
_umsMediaTypeChat	UTF String	
_umsFromAddr	UTF String	Either full User name of the user who published the message, or Facebook User ID
_umsToAddr	UTF String	Page name that the message was sent to
_umsMsgContext	int	Facebook thread ID, used to identify Chat session.
_umsMsgPlainText	UTF String	Text of the message.
_facebookPMThreadId	UTF String	Facebook thread ID.
_facebookPMThreadNMessages	int	Total number of messages in the thread
_facebookPMMessageId	UTF String	Facebook message ID.
_facebookMessageText	UTF String	Text of the message.
_facebookCreatedTime	UTF String	Last update time of the thread (UNIX timestamp)
_facebookActorId	UTF String	Facebook ID of the User who sent the message
_facebookActorName	UTF String	Name of the user who published the private message All of the content of <b>User Attributes</b> (table farther down)

Key Name	Value Type	Description
		this page) is added to UserData as well.
Subject	UTF String	Subject of the Message. Truncated version of Message text.
_facebookXML	UTF String	XML string that includes some info about all objects that participated in this thread (Thread, Message, Page, User)
_facebookPMInboxUrl	String	Value is set to a link that points to a list of all conversations for this Page.

## \_facebookComments Attributes

The value of the **\_facebookComments** attribute: a list of key-value pairs with type Key-value list and keys equal to **\_facebookCommentId**.

Key Name	Value Type	Comments
_facebookActorId	UTF String	Facebook ID of the object that published this Comment
_facebookActorIdType	int	Type of the Facebook object that published the Comment: <ul style="list-style-type: none"> <li>• 0—User</li> <li>• 1—Page</li> <li>• 2—Event</li> <li>• 3—Group</li> <li>• 4—Application</li> </ul>
_facebookActorName	UTF String	Name of the Page or User that published the Post. If it is a User (_facebookActorIdType = 0), then in addition to the name, the attributes listed in <b>User Attributes</b> below are also included.

Plus the contents of **Comment Attributes** below

Here is an example of **\_facebookComments**:

```
'_facebookComments' [lst] = KVList:
  '744384022244153_8236433' [lst] = KVList:
    '_facebookCommentId' [str] = "'744384022244153_8236433'"
    '_facebookCommentParentId' [str] = "0"
    '_facebookText' [str] = "This is a Comment on the Post"
```

```
'_facebookTime' [str] = "1384811057"
'_facebookCommentNLikes' [str] = "0"
'_facebookActorId' [str] = "507441822604742"
'_facebookActorIdType' [int] = 1
'_facebookActorName' [str] = "Some Page Name"
'744384022244153_8250812' [lst] = KVList:
'_facebookCommentId' [str] = "'744384022244153_8250812"
'_facebookCommentParentId' [str] = "'744384022244153_8236433"
'_facebookText' [str] = "This is a Reply to the Comment"
'_facebookTime' [str] = "1385154821"
'_facebookCommentNLikes' [str] = "10"
'_facebookActorId' [str] = "100000523748952"
'_facebookActorIdType' [int] = 0
'_facebookActorName' [str] = "UserFirstName UserLastName"
Also all key-value pairs from User Attributes
'744384022244153_8250813' [lst] = KVList:
'_facebookCommentId' [str] = "'744384022244153_8250813"
...
```

## Comment Attributes

Key Name	Value Type	Description
_facebookCommentId	UTF String	Facebook Comment ID
_facebookCommentParentId	UTF String	Facebook parent Comment ID
_facebookText	UTF String	The text of the Comment
_facebookTime	UTF String	UNIX timestamp associated with the creation time of the Comment
_facebookCommentNLikes	UTF String	The number of likes for the Comment

## User Attributes

Key Name	Description
url	User's app scope ID
first_name	First Name of the user
last_name	Last name of the user
link	URL to the user's Facebook profile
name	Display name of the user
updated_time	Timestamp for the last updated time of the user's profile
picture	Key-Value list containing following values: <ul style="list-style-type: none"> <li>is_silhouette - Should be true is user if user does not have any profile picture, and false if profile picture is present.</li> </ul>

Key Name	Description
	<ul style="list-style-type: none"><li>• url - URL to the user's profile picture</li></ul>

Sample of user attributes:

```
{
  "id": "240958046340982",
  "first_name": "John",
  "last_name": "Smith",
  "link": "https://www.facebook.com/app_scoped_user_id/240958046340982/",
  "name": "John Smith",
  "updated_time": "2017-06-09T11:43:41+0000",
  "picture": {
    "data": {
      { "is_silhouette": false, "url": "https://scontent.xx.fbcdn.net/v/t1.0-1/p50x50/16195176_258650437905076_3924523920539810613_n.jpg?oh=da57bd2ece1b88e84206b908936b654f&oe=59DB5A2D" }
    }
  }
}
```

## \_umsChannelInfo Attributes

The value of the `_umsChannelInfo` attribute: a list of key-value pairs that represent information about the Facebook object on whose behalf Posts/Comment/Replies are published by this particular Monitor.

Key Name	Value Type	Comments
<code>_facebookSourceId</code>	UTF String	ID of the Facebook object on whose behalf the Post/Comment/Reply is published
<code>_facebookSourceObjectType</code>	UTF String	Type of the object on whose behalf the Post/Comment/Reply is published: <ul style="list-style-type: none"> <li>• <code>com.genesyslab.mcr.facebook.fql.Application</code></li> <li>• <code>com.genesyslab.mcr.facebook.fql.User</code></li> <li>• <code>com.genesyslab.mcr.facebook.fql.Page</code></li> </ul>
<code>_facebookSourceNickName</code>	UTF String	Name of the object
<code>_facebookSourceXML</code>	UTF String	Information about the object packed in an XML string

Sample:

```
'_umsChannelInfo' [lst] = KVList:  
'channel-facebook-2-monitor-G-Page-MBAero-Main' [lst] = KVList:  
  '_facebookSourceId' [str] = "178227039004532"  
  '_facebookSourceObjectType' [str] = "com.genesyslab.mcr.facebook.fql.Page"  
  '_facebookSourceNickName' [str] = "Some Page Name"  
  '_facebookSourceXML' [str] = "<com.genesyslab.mcr.facebook.fql.Page>"  
    <page__id>178227039004532</page__id>  
    <name>Some Page</name>  
    <pic__small>https://fbcdn-profile-a.akamaihd.net/hprofile-ak-ash1/373048_178227039004532_890644691_t.jpg</pic__small>  
    <pic__big>https://fbcdn-profile-a.akamaihd.net/hprofile-ak-ash1/373048_178227039004532_890644691_n.jpg</pic__big>  
    <page__url>https://www.facebook.com/SomePage</page__url>  
    <type>COMMUNITY</type>  
</com.genesyslab.mcr.facebook.fql.Page>"
```

# Deploy DMS with a Twitter Channel

For a Twitter channel, you need two installation packages:

- Digital Messaging Server
- Genesys Cloud API Driver for Twitter

The Driver adds Twitter-specific features to DMS and does not require its own Application object in the Configuration Server database.

You can also create a [Custom Media Channel Driver](#).

## Important

Tweets from a customer who has a protected twitter account will not enter DMS even if the business twitter account is authenticated to view the protected tweets from the customer's twitter account.

## Prepare the Twitter Channel

1. [Deploy DMS](#).
2. Run the installation for Genesys Driver for Use with Twitter, selecting the desired DMS object.
3. Locate the **driver-for-twitter-options.cfg** configuration file in the \<Digital Messaging Server application>\media-channel-drivers\channel-twitter directory.
4. In Genesys Administrator Extension, open your Digital Messaging Server Application, go to the **Options** tab, and import **driver-for-twitter-options.cfg**, selecting **No** in response to **Do you want to overwrite the existing data?**

## Configure the Options

Refer to the [Genesys Cloud API Driver for Twitter](#) page in the [eServices Options Reference](#).

# Twitter Interaction Attributes

## Basic Attributes

Genesys Cloud API Driver for Twitter adds the following attributes to the user data of a Twitter interaction.

Key name	Value type	Description
_twitterMsgType	UTF string	Message type Values: DirectMessage, Status.
_twitterQueryName	String	Query name of the channel monitor.
_twitterCreatedAt	UTC Time Zone	Message creation date and time.
_twitterMsgId	String	Message ID.
_twitterIsFavorited	String	Indicates whether the Tweet has likes (formerly known as favorites) or not.
_twitterFavoriteCount	Integer	Number of likes received by the Tweet.
_twitterUserName	String	Full name of the user who posted the Tweet.
_twitterFirstName	String	First name of the user.
_twitterLastName	String	Last name of the user.
_twitterUserId	String	Twitter user ID.
_twitterUserScreenName	String	Screen name of the user.
_twitterUserCreatedAt	UTC Time Zone	The UTC date and time when the user account was created on Twitter.
_twitterUserLanguage	String	Language code in the user's profile.
_twitterUserImageURL	String	URL of the user profile image (also known as avatar).
_twitterUserTimeZone	String	Time zone of the Twitter user.
_twitterUserUtcOffset	String	UTC offset for the user's time zone.
_twitterUserFollowersCount	String	The user's current number of followers.
_twitterUserFriendsCount	String	The number of other users this user is currently following.
_twitterUserStatusesCount	String	The number of Tweets (including

Key name	Value type	Description
		Retweets) posted by the user.
_twitterFromAddr	String	Screen name of the message author.
_twitterFromUserId	String	User ID of the message author.
_twitterUserIsFollower	String	Indicates whether the user is a follower of the channel's account or not.
_twitterUserIsFriend	String	Indicates whether the channel's account is following the user or not.
_twitterToAddr	String	Screen name of the message addressee.
_twitterToUserId	String	User ID of the message addressee.
_twitterMsgPlainText	String	Message content in plain text format.
Subject	String	Message subject.
_twitterMsgLanguage	String	Language code of the Tweet.
_twitterSource	String	Source of the message.
_twitterIsRetweet	String	Indicates whether the message is a Retweet of another message.
_twitterEntities	Key-value list	URL entities expressed as a key-value pairs list with URL and Media attributes.
_umsChannel	String	Channel name.
_umsMediaAccount	String	Twitter account ID of the channel.
_umsChannelInfo	Key-value list	The list of key-value pairs provided in <a href="#">_umsChannelInfo Attributes</a> below.
_twitterUtterances	Key-value list	The list of key-value pairs provided in <a href="#">_twitterUtterances Attributes</a> below.
_twitterTopics	String	Keywords in Tweets.
_twitterTimeFrame	String	Timeout for the group ID, in seconds. A new group ID is assigned after this timeout.
_twitterGroupId	String	Twitter group ID assigned to a group of Tweets from the same sender.

## \_umsChannelInfo Attributes

Key name	Value type	Description
_twitterUserId	String	Twitter account ID of the channel.
_twitterUserScreenName	String	Screen name of the channel's Twitter account.
_twitterUserName	String	The brand name which is the same as the channel's Twitter user name.
_twitterUserImageURL	String	URL of the brand's Twitter profile image (also known as avatar).
_twitterSourceNickName	String	Value retrieved from the <b>x-source-nick-name</b> configuration parameter.
_twitterSentiment	String	Sentiment polarity of the Tweet. It can be Positive, Neutral, or Negative.
_twitterSentimentScore	String	A numerical representation of the sentiment polarity of the Tweet.
_twitterSentimentConfidence	String	Confidence score of sentiment extractor.
_twitterActionability	String	A score determined by the number of Tweets that are accepted and rejected by agents. The default value is 50.

## \_twitterUtterances Attributes

Key name	Value type	Description
content	String	A speech act, usually a sentence from the received message/ Tweet. The initial message is divided into speech acts using punctuation patterns.
intention_type	String	Result of intention classifier.
intention_type_confidence	String	Confidence score of intention classifier.
sentiment	String	Sentiment polarity of the speech act. It can be Positive, Neutral, or Negative.
sentiment_score	String	A numerical representation of the sentiment polarity of the current speech act.
sentiment_confidence	String	Confidence score of sentiment extractor.

Key name	Value type	Description
topic_confidence	String	Confidence score of topic extractor.
topics	String	Noun phrases from the speech acts.

## Sample Interactions

### Public Tweet

```
'_twitterQueryName' [str] = "Public"
'_twitterCreatedAt' [str] = "2017-12-18T10:22:10Z"
'_twitterMsgType' [str] = "Status"
'_twitterMsgId' [str] = "942701514744672256"
'_twitterIsFavorited' [str] = "false"
'_twitterFavoriteCount' [int] = 0
'_twitterUserName' [str] = "Shruti Mishra"
'_twitterFirstName' [str] = "Shruti"
'_twitterLastName' [str] = "Mishra"
'_twitterUserId' [str] = "834289359406915584"
'_twitterUserScreenName' [str] = "shruti14111"
'_twitterUserCreatedAt' [str] = "2017-02-22T14:30:58Z"
'_twitterUserLanguage' [str] = "en"
'_twitterUserImageURL' [str] = "http://abs.twimg.com/sticky/default_profile_images/default_profile_normal.png"
'_twitterUserTimeZone' [str] = "New Delhi"
'_twitterUserUtcOffset' [str] = "19800.0"
'_twitterUserFollowersCount' [str] = "12"
'_twitterUserFriendsCount' [str] = "33"
'_twitterUserStatusesCount' [str] = "2071"
'_twitterFromAddr' [str] = "shruti14111"
'_twitterFromUserId' [str] = "834289359406915584"
'_twitterUserIsFollower' [str] = "true"
'_twitterUserIsFriend' [str] = "true"
'_twitterInReplyToUserId' [str] = "868003715545931776"
'_twitterToUserId' [str] = "868003715545931776"
'_twitterInReplyToScreenName' [str] = "Sparrow11071"
'_twitterToAddr' [str] = "Sparrow11071"
'_twitterMsgPlainText' [str] = "@Sparrow11071 Text Tweet to brand"
'Subject' [str] = "@Sparrow11071 Text Twe..."
'_twitterMsgLanguage' [str] = "en"
'_twitterSource' [str] = "<a href='\"http://twitter.com\"' rel='\"nofollow\"'>Twitter Web Client</a>"
'_twitterIsRetweet' [str] = "false"
'_twitterEntities' [lst] = KVList:
'Media' [lst] = KVList:
'Url' [lst] = KVList:
'_umsChannel' [str] = "channel-twitter-Test"
'_umsMediaAccount' [str] = "868003715545931776"
'_umsChannelInfo' [lst] = KVList:
  '_twitterUserId' [str] = "868003715545931776"
  '_twitterUserScreenName' [str] = "Sparrow11071"
  '_twitterUserName' [str] = "Sparrow Products"
  '_twitterUserImageURL' [str] = "http://abs.twimg.com/sticky/default_profile_images/default_profile_normal.png"
'_twitterSourceNickName' [str] = "*set value*"
'_twitterSentiment' [str] = "Neutral"
```

```
'_twitterSentimentScore' [str] = "0.0"
'_twitterSentimentConfidence' [str] = "5.0"
'_twitterActionability' [str] = "50.0"
'_twitterUtterances' [lst] = KVList:
  'utterance-0' [lst] = KVList:
    'content' [str] = "@Sparrow11071 Text Tweet to brand"
    'intention_type' [str] = "JUNK"
    'intention_type_confidence' [str] = "3.0"
    'sentiment' [str] = "Neutral"
    'sentiment_score' [str] = "0.0"
    'sentiment_confidence' [str] = "5.0"
    'topic_confidence' [str] = "74.0"
    'topics' [str] = "text tweet,sparrow11071"
'_twitterTopics' [str] = "text tweet,sparrow11071"
'_twitterTimeFrame' [str] = "60.0"
'_twitterGroupId' [str] = "834289359406915584:5977904549637487079"
```

## Direct Message

```
'_twitterQueryName' [str] = "User"
'_twitterCreatedAt' [str] = "2017-12-18T10:24:57Z"
'_umsMediaType' [str] = "twitterdirect"
'_twitterMsgType' [str] = "DirectMessage"
'_twitterMsgId' [str] = "942702217122824196"
'_twitterUserName' [str] = "Shruti Mishra"
'_twitterFirstName' [str] = "Shruti"
'_twitterLastName' [str] = "Mishra"
'_twitterUserId' [str] = "834289359406915584"
'_twitterUserScreenName' [str] = "shruti14111"
'_twitterUserCreatedAt' [str] = "2017-02-22T14:30:58Z"
'_twitterUserLanguage' [str] = "en"
'_twitterUserImageURL' [str] = "http://abs.twimg.com/sticky/default_profile_images/default_profile_normal.png"
'_twitterUserTimeZone' [str] = "New Delhi"
'_twitterUserUtcOffset' [str] = "19800.0"
'_twitterUserFollowersCount' [str] = "12"
'_twitterUserFriendsCount' [str] = "33"
'_twitterUserStatusesCount' [str] = "2071"
'_twitterFromAddr' [str] = "shruti14111"
'_twitterFromUserId' [str] = "834289359406915584"
'_twitterUserIsFollower' [str] = "true"
'_twitterUserIsFriend' [str] = "true"
'_twitterToAddr' [str] = "Sparrow11071"
'_twitterToUserId' [str] = "868003715545931776"
'_twitterMsgPlainText' [str] = "TEST DM"
'Subject' [str] = "TEST DM"
'_umsChannel' [str] = "channel-twitter-Test"
'_umsMediaAccount' [str] = "868003715545931776"
'_umsChannelInfo' [lst] = KVList:
  '_twitterUserId' [str] = "868003715545931776"
  '_twitterUserScreenName' [str] = "Sparrow11071"
  '_twitterUserName' [str] = "Sparrow Products"
  '_twitterUserImageURL' [str] = "http://abs.twimg.com/sticky/default_profile_images/default_profile_normal.png"
  '_twitterSourceNickName' [str] = "*set value*"
'_twitterEntities' [lst] = KVList:
'Media' [lst] = KVList:
'Url' [lst] = KVList:
'_twitterSentiment' [str] = "Neutral"
'_twitterSentimentScore' [str] = "0.0"
'_twitterSentimentConfidence' [str] = "5.0"
'_twitterActionability' [str] = "50.0"
```

```
'_twitterUtterances' [lst] = KVList:  
'utterance-0' [lst] = KVList:  
  'content' [str] = "TEST DM @Sparrow11071"  
  'intention_type' [str] = "RECOMMENDATION"  
  'intention_type_confidence' [str] = "24.0"  
  'sentiment' [str] = "Neutral"  
  'sentiment_score' [str] = "0.0"  
  'sentiment_confidence' [str] = "5.0"  
  'topic_confidence' [str] = "75.0"  
  'topics' [str] = "sparrow11071"  
'_twitterTopics' [str] = "sparrow11071"  
'_twitterTimeFrame' [str] = "60.0"  
'_twitterGroupId' [str] = "834289359406915584_868003715545931776:5977904549637654629"
```

# Custom Media Channel Driver

Genesys supplies drivers for media channels that connect DMS to Twitter and Facebook. You can also create a custom media channel driver, as described in [this API reference](#).

## Important

Installation of DMS is described on the [Deploy Digital Messaging Server](#) page.

---

# Security

This topic describes the security related configuration in DMS.

## Enabling a TLS connection as a Windows Service (Optional)

**Prerequisite:** TLS 1.2 or higher

When DMS has Transport Layer Security (TLS) configured, either as a server on its ESP port, or as a client in its connection to Configuration Server, Interaction Server, Message Server, Chat Server, and UCS, follow these steps to enable it as a Windows Service:

1. Select the Windows service related to DMS .
2. Select the **Log On** tab. The default setting is **Log on as local system account**.
3. Select **Log on as this account** and provide the login/password of a local host user.

## Masking sensitive data in log files

Although values for sensitive data such as passwords are masked in key-value lists, these values are not masked when users view or modify the related configuration options.

You can use the internal log-filtering mechanism in DMS to properly mask these values, based on the **logging-filter-default.json** configuration file that you put into the directory where your DMS jar file resides. Specify the configuration file to use in the value for **logging-filter-spec**. [Click here](#) to download a sample for **logging-filter-default.json**.

First, define a set of filters that are applied to the server's log messages before they are passed to a logging system. The filters intercept the original message's content and produce new content (possibly empty values) for specific messages in a log file (for example, a message that has specific identification information).

There are three types of filter procedures:

- Skip—Produces empty new content,
- Hide—Produces standard placeholder as a new content,
- Edit—Produces new content as a transformation of an original content.

The filter can modify content as part of a series of steps. For example, it can mask one category of information before masking a separate category.

Modification of content is based on a search-and-replace approach using regular expressions and replace expressions (“search” predicate and “replace” action). See the following links for more information:

- [Lesson: Regular Expressions \(Oracle\)](#)
- [Class Pattern \(Oracle\)](#)
- [Regular Expression Language - Quick Reference \(Microsoft\)](#)

You must extensively test regular expressions to ensure they perform as expected in all cases. The following tools might be useful for testing:

- [Regex Planet](#)
- [RegExr](#)

The following are examples and definitions of typical sensitive data:

- [Bank card number](#)
- [Social Security Number](#)
- [Phone number](#)

## Hiding Selected Data in Logs

This feature implements a Genesys standard detailed in the [Genesys Security Deployment Guide](#). It enables you to hide selected key/value pairs in the Parameters and UserData attributes of log messages generated by DMS. You can choose to hide just the value itself by replacing it with a series of asterisks (\*), or you can remove the whole key/value pair from the log output.

### Configuring [log-filter] and [log-filter-data] sections

This feature is implemented by defining the following configuration options in the DMS Application object:

- **default-filter-type** in the **[log-filter]** section defines the treatment for all KV pairs in the Parameters and User Data attributes.
  - This setting will be applied to the attributes of all KVList pairs in the attribute except those that are explicitly defined in the **[log-filter-data]** section.
- One or more **<key-name>** options in the **[log-filter-data]** section define the treatment for specific keys in the log, overriding the default treatment specified by **default-filter-type**.
  - If no value is specified for this option, no additional processing of this data element is performed.

#### Important

The default settings of the options enable all data to be visible in the log.

You can get additional implementation samples in the [Genesys Security Deployment Guide](#). For detailed descriptions of the configuration options used to configure this feature, refer to the [Framework Configuration Options Reference Manual](#).

---

## Supported Filters

### Important

custom-filter options are not supported.

Filter Name	Description
copy	The keys and values of the KVList pairs are copied to the log.
hide	The keys of the KVList pairs are copied to the log; the values are replaced with strings of asterisks.
skip	The KVList pairs are not copied to the log.

## Other DMS Configuration

See the [Digital Messaging Server Options Reference](#) for information on the configuration options available.

Options for a particular monitor override the general channel options.

You must set the following Java VM properties according to which DMS channels are used in your environment:

Channel(s)	Options
Chat mode only is used in all DMS channels	<ul style="list-style-type: none"> <li>• <b>-Dgenesys.mcr.stdserverex.itxrequired = false</b></li> <li>• <b>-Dgenesys.mcr.stdserverex.scsrequired = true</b></li> <li>• <b>-Dgenesys.mcr.stdserverex.chatrequired = true</b></li> </ul>
Paging mode only is used in all DMS channels	<ul style="list-style-type: none"> <li>• <b>-Dgenesys.mcr.stdserverex.itxrequired = true</b></li> <li>• <b>-Dgenesys.mcr.stdserverex.scsrequired = false</b></li> <li>• <b>-Dgenesys.mcr.stdserverex.chatrequired = false</b></li> </ul>
Both chat and paging modes are used in DMS channels	<ul style="list-style-type: none"> <li>• <b>-Dgenesys.mcr.stdserverex.itxrequired = true</b></li> <li>• <b>-Dgenesys.mcr.stdserverex.scsrequired = true</b></li> <li>• <b>-Dgenesys.mcr.stdserverex.chatrequired = true</b></li> </ul>
DMS channels do not post messages to DMS (for example, a channel for bots)	<ul style="list-style-type: none"> <li>• <b>-Dgenesys.mcr.stdserverex.itxrequired = false</b></li> <li>• <b>-Dgenesys.mcr.stdserverex.scsrequired = false</b></li> <li>• <b>-Dgenesys.mcr.stdserverex.chatrequired = false</b></li> </ul>

---

## Enabling proxy server support for DMS (Optional)

### Important

Starting with the release of Genesys Driver for use with Apple Business Chat 9.0.002.05, basic authentication over HTTP is supported between the driver and proxy server.

#### 1. Edit the appropriate file by adding these Java runtime startup options:

- If using Genesys Cloud API Driver for Facebook 8.5.300.09 or earlier, or Genesys Cloud API Driver for Twitter 8.5.300.07 or earlier:
  - -DproxySet=true
  - -DproxyHost=<host of proxy server>
  - -DproxyPort=<port of proxy server>
- If using Genesys Cloud API Driver for Facebook 8.5.300.10 or later, or Genesys Cloud API Driver for Twitter 8.5.300.08 or later:
  - -Dhttp.proxyHost=<host of proxy server>
  - -Dhttp.proxyPort=<port of proxy server>
  - -Dhttps.proxyHost=<host of proxy server>
  - -Dhttps.proxyPort=<port of proxy server>
- If using Genesys Driver for Apple Business Chat or Genesys Driver for Genesys Hub (WhatsApp) via HTTPS proxy server:
  - -Dhttps.proxyHost=<host of proxy server>
  - -Dhttps.proxyPort=<port of proxy server>
- If using Genesys Driver for Apple Business Chat via HTTP proxy server with basic authentication:
  - -Dhttp.proxyHost=<host of proxy server>
  - -Dhttp.proxyPort=<port of proxy server>
  - -Dhttp.proxyUser=<username>
  - -Dhttp.proxyPassword=<password>
  - -Djdk.http.auth.tunneling.disabledSchemes= (Note that the value of this JVM property must be empty)

#### 2. To enable authentication for the proxy server, add these options as well:

- If using Genesys Cloud API Driver for Facebook 8.5.300.09 or earlier, or Genesys Cloud API Driver for Twitter 8.5.300.07 or earlier:
  - -DproxyUser=<username>
  - -DproxyPassword=<password>

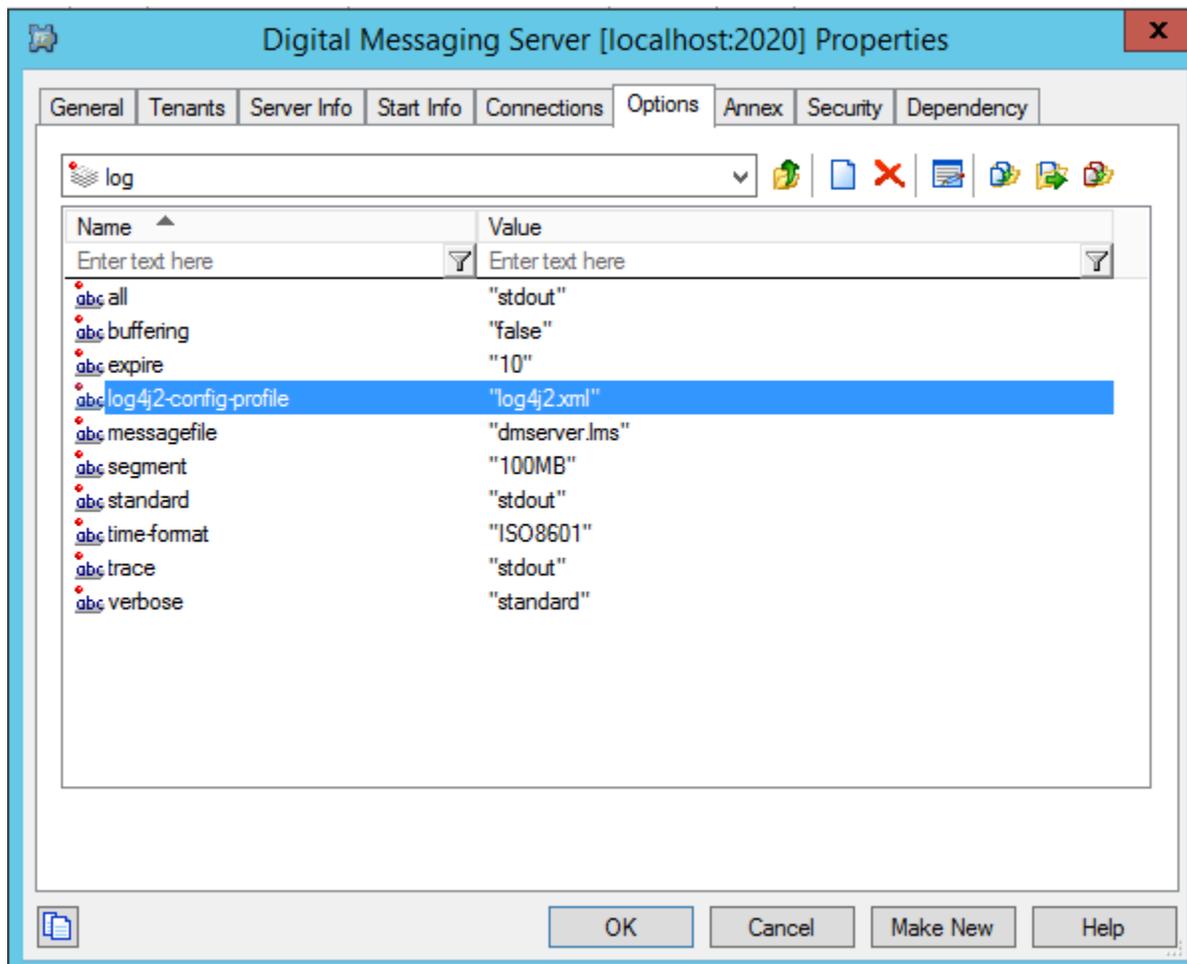
- If using Genesys Cloud API Driver for Facebook 8.5.300.10 or later, or Genesys Cloud API Driver for Twitter 8.5.300.08 or later:
  - -Dhttp.proxyUser=<username>
  - -Dhttp.proxyPassword=<password>

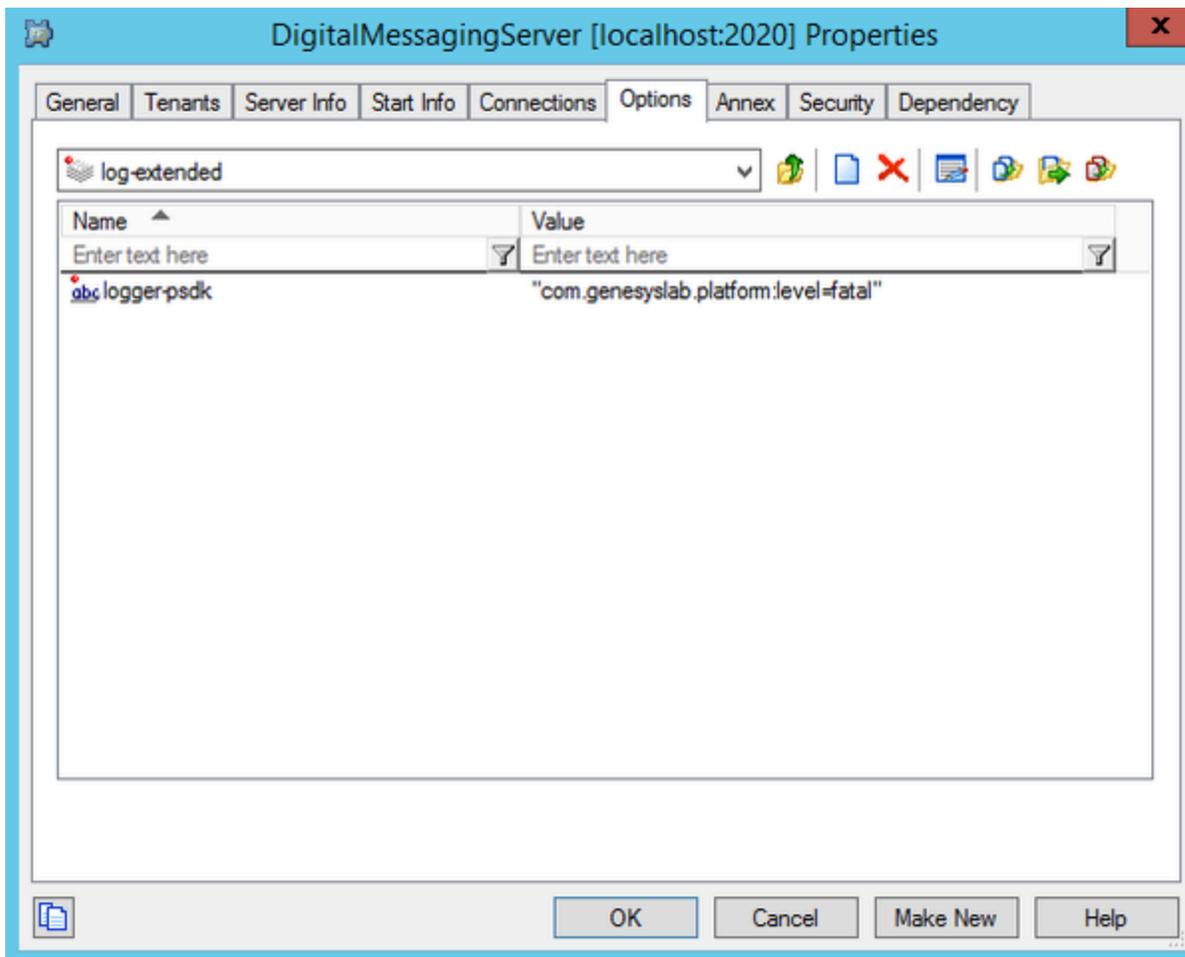
### 3. File to edit:

- Windows: In **JavaServerStarter.ini**, the **[JavaArgs]** section.
- Unix and related systems: In **dmserver.sh**, the last string, namely "**\$JAVA\_EXECUTABLE\_PATH/java -classpath \$CLASSPATH \$JVM\_PARAMS \$MAINCLASS \$\***"

## Setting up logging

After an upgrade, ensure that the **log4j2.xml** file is present in the DMS directory and set up the following logging options:





# Outbound interaction from Workflow

Starting from the DMS version 9.1.003.10, the InviteToChat method is supported. This method allows initiation of outbound chat sessions from a strategy and sending WhatsApp notification templates to customers. It is an invitation for a customer who is using voice IVR interaction to switch to Chat or Messaging and continue the conversation.

The InviteToChat method is supported for the following media types:

- Apple Business Chat
- WhatsApp

### Important

The InviteToChat method can only be used with DMS that is configured with the Flex Chat protocol.

## ESP request

**AppName:** Application name **AppType:** If application type is empty, **AppName** must be specified  
**Service:** ChatService **Method:** InviteToChat

Parameter	Value Type	Mandatory	Value Description
_umsChannel	String	Mandatory (only if more than one channel configured)	DMS channel name
_umsClientId	String	Mandatory	<ul style="list-style-type: none"> <li>• Apple Business Chat: Conversation ID</li> <li>• WhatsApp: Client phone number</li> <li>• SMS: Client phone number (Currently, not supported)</li> </ul>
_umsBusinessId	String	Mandatory	Business phone number which is set for a specific channel in the HUB configuration.
StandardResponseId	String	Optional	The default value is no SR is specified. The following content types are supported:

Parameter	Value Type	Mandatory	Value Description
			<ul style="list-style-type: none"> <li>• Apple Business Chat                             <ul style="list-style-type: none"> <li>• Plain text with and without attachments</li> <li>• All types of structured content ("interactive messages") supported by Apple Business Chat</li> <li>• Plain text with and without attachments</li> <li>• "contentType": "generic" (implements UI element "Carousel")</li> <li>• "contentType": "quick-replies"</li> <li>• "contentType": "single-selection-list"</li> </ul> </li> <li>• WhatsApp                             <ul style="list-style-type: none"> <li>• Plain text with and without attachments</li> <li>• "contentType": "notification"</li> </ul> </li> <li>• SMS                             <ul style="list-style-type: none"> <li>• Plain text</li> </ul> </li> </ul>
ChatQueueKey	String	Optional	The chat queue key pointing to the Chat Server endpoint that is used to place the newly created outbound interactions into the interaction queue. If this value is specified, the outbound interactions created from the strategy are placed in the specified interaction queue. By specifying a

Parameter	Value Type	Mandatory	Value Description
			<p>dedicated queue, users can differentiate between outbound interactions created from the strategy and inbound interactions. The queue can be used to route interactions to agents, park interactions, or terminate interactions using <b>TerminateSessionFlag</b>.</p> <p>If this value is not specified, it is set using the existing channel configuration option, <b>inbound-route-by-call</b>.</p> <p>The format is &lt;tenant-id&gt;:&lt;Chat Server endpoint name&gt; . For example, 1:session-WA-sendSR.</p>
TerminateSessionFlag	String	Optional	<p>Indicates if the chat session created by the request must be ended immediately. For example, this parameter can be used to send out notifications (using outbound templates) that are terminated immediately after they served their purpose.</p> <p>For the value true, it is recommended to use a dedicated <b>ChatQueueKey</b> with a routing strategy which stops unneeded chats placed in this queue. The strategy should also implement removal of occasional lost chats from this queue.</p> <p>The default value is false.</p>
Contact	KVList	Optional	Data used for field codes rendering.
Agent	KVList	Optional	Data used for field codes rendering.
CustomProperties	KVList	Optional	Data used for field codes rendering.

### Attached user data

In addition to the above listed attributes, you can place any user data in the request to implement

specific business tasks, for example, to support routing to the agent who initiated the call. DMS passes all user data received with this ESP request to Chat Server. The following user data attributes are recommended:

- **User data attribute:** Allows Chat Server or UCS to identify (find or create) contact using these values. A standard interaction attribute such as **PhoneNumber** or a specific one that is configured by a customer, may be used. For information on how to perform contact identification, see [Contact Identification](#).  
The following data can be placed in the attribute:
  - Apple Business Chat: Conversation ID
  - WhatsApp: Client phone number
  - SMS: Client phone number (Currently, not supported)
- **GCTI\_Chat\_SubmitAsOutbound = true** : Instructs Chat Server to create interaction with type=Outbound and subtype=OutboundNew. The default value (if absent) is false and it indicates the type=Inbound and subtype=InboundNew.
- **LastCalledAgent\_EmployeeID** : Allows the use of Last Call Agent Routing technique.

### ESP response

Parameter	Value Type	Mandatory	Value Description
_umsChatSessionId	String	Mandatory	The Chat ID by Chat Server. It is the same as <b>InteractionId</b> of the newly created interaction.

### Example

This section provides an example of a WhatsApp notification sent using an IRD strategy.

### Standard response

Prepare a standard response with the required field codes. The example demonstrates custom field codes of the section **Contact** and field codes of the section **CustomProperties**.

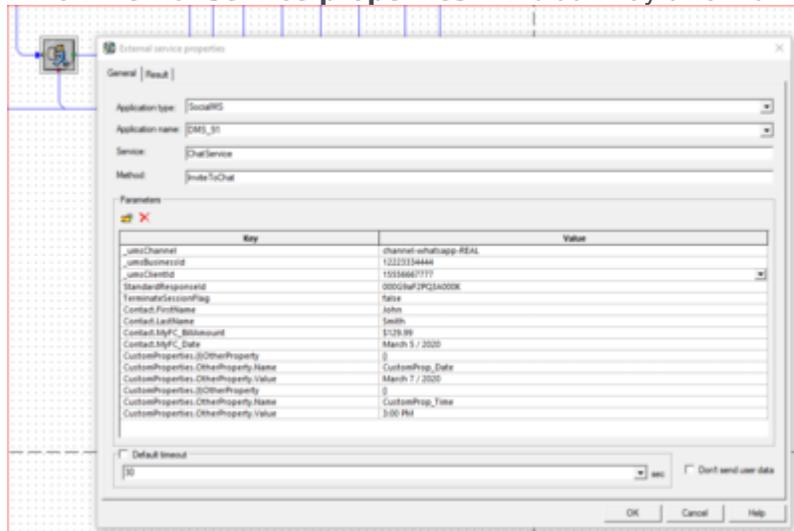
```

{
  "contentType": "notification",
  "content": {
    {
      "index": 2,
      "type": "notification",
      "language": "en_US",
      "id": "96729df3_ba05_1452_ebe0_cabe40762a81@appointment_new_1",
      "text": "Hello {{1}},\nYour bill of {{2}} is due on '{{3}}'.",
      "parameters": [
        {
          "name": "Customer Name",
          "index": "1",
          "value": "<${ Contact.FirstName $} <${ Contact.LastName $}"
        },
        {
          "name": "Amount",
          "index": "2",
          "value": "<${ Contact.MyFC_BillAmount $} "
        },
        {
          "name": "Due Date",
          "index": "3",
          "value": "<${ CustomProp_Date $} <${ CustomProp_Time $}"
        }
      ]
    },
    {
      "index": 1,
      "type": "image",
      "url": "genexya://ucs/document/00095aF2PQ3A00HY?file-name=Hulibut-png-192k.png",
      "name": "Hulibut-png-192k.png",
      "mime": "image/png"
    },
    {
      "index": 4,
      "type": "quick-reply",
      "text": "Got It"
    },
    {
      "index": 5,
      "type": "quick-reply",
      "text": "Wrong Addressee"
    },
    {
      "index": 3,
      "type": "text",
      "text": "Your next bill reminder."
    }
  ]
}

```

### Interaction Routing Designer (IRD)

Specify ESP request parameters in the **External service properties** IRD block. Pay attention to use



lowercase L in the elements {L}. The above IRD block produces the following ESP request (taken from the DMS log file):

```
12:12:13.139 Std 50003 Request received:
'Request3rdServer' (500) attributes:
ReferenceId [int] = 9
UserData [bstr] = [output suppressed]
Request [bstr] = KVLList:
  'Version' [str] = "1.0"
  'AppName' [str] = "DMS_91"
  'Service' [str] = "ChatService"
  'Method' [str] = "InviteToChat"
  'Parameters' [lst] = KVLList:
    '_umsChannel' [str] = "channel-whatsapp-REAL"
    '_umsBusinessId' [str] = "12223334444"
    '_umsClientId' [str] = "15556667777"
    'StandardResponseId' [str] = "000G9aF2PQ3A000K"
    'TerminateSessionFlag' [str] = "false"
    'Contact' [lst] = KVLList:
      'FirstName' [str] = "John"
      'LastName' [str] = "Smith"
      'MyFC_BillAmount' [str] = "$129.99"
      'MyFC_Date' [str] = "March 5 / 2020"
    'CustomProperties' [lst] = KVLList:
      'OtherProperty' [lst] = KVLList:
        'Name' [str] = "CustomProp_Date"
        'Value' [str] = "March 7 / 2020"
      'OtherProperty' [lst] = KVLList:
        'Name' [str] = "CustomProp_Time"
        'Value' [str] = "3:00 PM"
```

### WhatsApp client

The following message is displayed on the WhatsApp client:



---

# DMS Scalability

Digital Messaging Server allows a scalable functional mode that can be used when a flow of inbound messages from a media channel exceeds the processing capacity of a standalone DMS. The scalable functional mode is implemented by running multiple DMS instances. The DMS instances that participate in the scalable cluster of DMS's share the load of processing messages between them.

## Prerequisites

- The DMS Scalability feature is supported in the following versions
  - 9.1.007.06
  - 9.1.008.02
  - 9.1.008.06
  - 9.1.008.07
- Genesys Driver for use with Genesys Hub 9.1.007.06 or higher

## Implementation

The scalable functional mode uses the Redis server as a central data exchange element for the DMS instances of the DMS cluster. Redis server version 6.2 and Redisson (Redis client) are used in the implementation.

Two types of DMS instances participate in the processing of messages:

- Dispatcher (DMS-d)
- Workers (DMS-w1, DMS-w2, et al.)

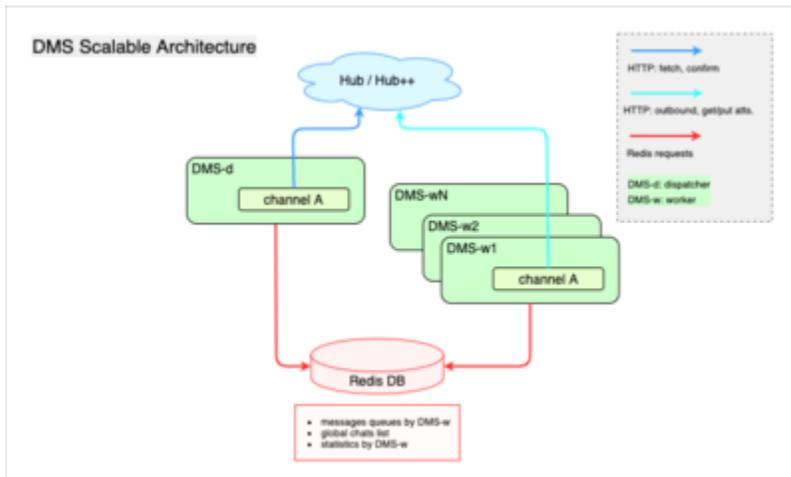
All DMS instances involved in one scalable cluster are restricted to handle one media channel.

## Architecture

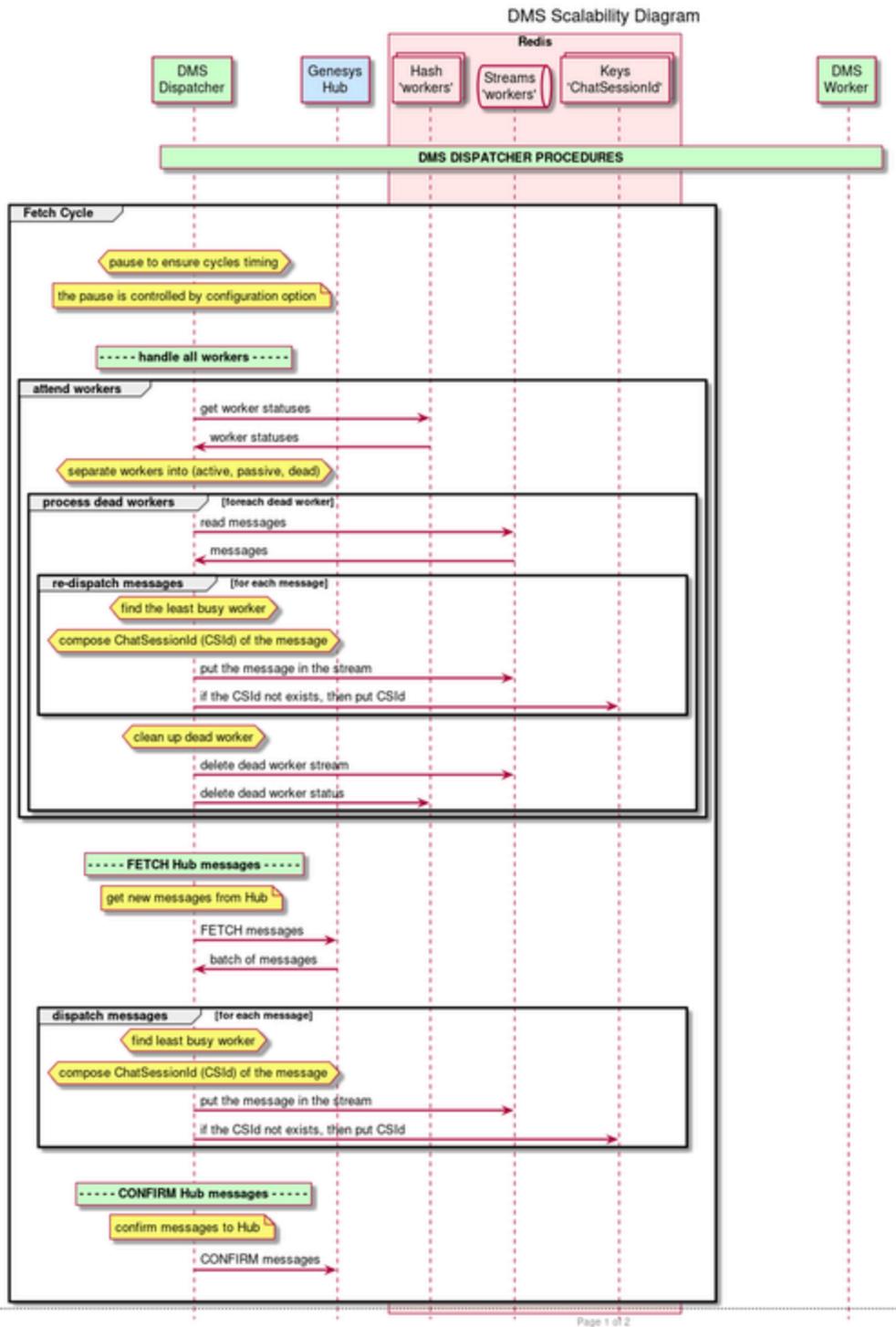
### Important

- The current implementation of DMS High Availability works without any additional changes. Each DMS-w relies on its backup to restore those established chat sessions in redundant Chat Servers.

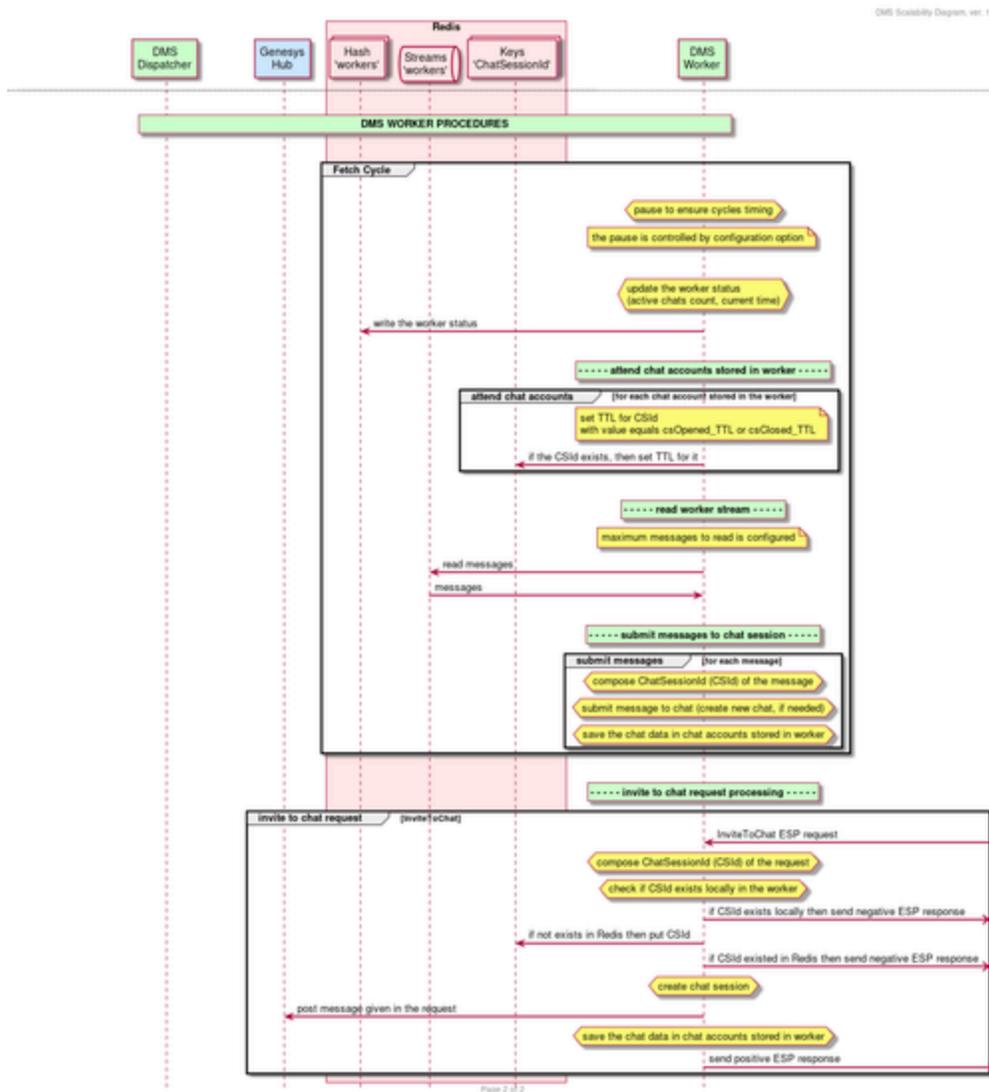
- The architecture diagram shown below do not represent backups.



### Dispatcher workflow



### Worker workflow



### DMS-d responsibilities

- Gets authorization token similar to how a regular DMS does.
- Fetches messages from Genesys Hub and puts them into Redis.
- Confirms fetched messages to Hub after it saved them in Redis.
- Distributes messages among the available workers.
- Detects inactive workers to re-dispatch messages between active workers.

## DMS-w responsibilities

- Gets authorization token similar to how a regular DMS does.
- Records its status in Redis regularly to indicate its state.
- Reads the messages assigned to it from Redis and creates chat sessions.
- Updates the records in Redis with the chat session IDs and its own name.
- Reads messages from Redis, places them into the existing chats, and marks them as processed in Redis so that if a session needs to be restored in another worker, then the dispatcher can identify which messages would require re-sending.
- Sends outbound messages to Genesys Hub directly.

## Configuration

The following options in the **settings** section can be used to configure DMS scalability:

- The `server-mode` option indicates the mode of operation for DMS.
- The `redis-client-config` option specifies the file name of the Redis client configuration (.yaml file).

## Redisson client configuration

Genesys recommends that you use a secured connection with the Redis server. To ensure the secured connection, the Redisson client configuration must specify the address of the Redis server in the form of `rediss://`. Refer to the example for a sample address.

Additionally, you must describe the JKS key store (parameter `sslTruststore`) with the CA certificate which is used to verify the Redis server's certificate. Example:

```
singleServerConfig:
  idleConnectionTimeout: 10000
  connectTimeout: 10000
  timeout: 3000
  retryAttempts: 3
  retryInterval: 1500
  password: "password"
  subscriptionsPerConnection: 5
  clientName: "DMS"
  sslEnableEndpointIdentification: true
  sslProvider: "JDK"
  # sslTruststore: "file:///C:/Redis_TLS/RedisCA_Test.keystore"
  pingConnectionInterval: 30000
  keepAlive: false
  tcpNoDelay: false
  address: "redis://127.0.0.1:6379"
  # address: "rediss://wsl-win10desktop:6379"
  subscriptionConnectionMinimumIdleSize: 1
  subscriptionConnectionPoolSize: 5
  connectionMinimumIdleSize: 5
  connectionPoolSize: 10
  database: 0
  dnsMonitoringInterval: 5000
```

```

threads: 16
nettyThreads: 32
#codec: !<org.redisson.client.codec.JsonJacksonCodec> {}
referenceEnabled: true
transportMode: "NIO"
lockWatchdogTimeout: 30000
reliableTopicWatchdogTimeout: 600000
keepPubSubOrder: true
useScriptCache: false
minCleanupDelay: 5
maxCleanupDelay: 1800
cleanupKeysAmount: 100
nettyHook: !<org.redisson.client.DefaultNettyHook> {}
useThreadClassLoader: true
addressResolverGroupFactory: !<org.redisson.connection.DnsAddressResolverGroupFactory> {}
    
```

### JVM parameters for DMS-d

```

-Dgenesys.mcr.stdserverex.apptype=CFGSocialMS
-Dgenesys.mcr.stdserverex.espserver=false
-Dgenesys.mcr.stdserverex.itxrequired=false
-Dgenesys.mcr.stdserverex.scsrequired=true
-Dgenesys.mcr.stdserverex.chatrequired=false
-Dgenesys.mcr.stdserverex.ucsrequired=false
-Dgenesys.mcr.stdserverex.ucscclusterrequired=false
-Dgenesys.mcr.stdserverex.espserver.multipleesports=false
-Dgenesys.mcr.stdserverex.lmsfile=dmsserver.lms
-Dgenesys.mcr.stdserverex.infoproviderservice=com.genesyslab.mcr.smsserver.OwnInfoProvider
-Dgenesys.mcr.stdserverex.flexchatprotocol=true
    
```

### JVM parameters for DMS-w

```

-Dgenesys.mcr.stdserverex.apptype=CFGSocialMS
-Dgenesys.mcr.stdserverex.espserver=true
-Dgenesys.mcr.stdserverex.itxrequired=true
-Dgenesys.mcr.stdserverex.scsrequired=true
-Dgenesys.mcr.stdserverex.chatrequired=true
-Dgenesys.mcr.stdserverex.ucsrequired=true
-Dgenesys.mcr.stdserverex.ucscclusterrequired=false
-Dgenesys.mcr.stdserverex.espserver.multipleesports=false
-Dgenesys.mcr.stdserverex.lmsfile=dmsserver.lms
-Dgenesys.mcr.stdserverex.infoproviderservice=com.genesyslab.mcr.smsserver.OwnInfoProvider
-Dgenesys.mcr.stdserverex.flexchatprotocol=true
    
```

### Configuration layer connections

You must ensure the following connections in the Configuration Layer:

DMS instance	Connects to
DMS-d	<ul style="list-style-type: none"> <li>• Solution Control Server</li> <li>• Message Server, if required</li> </ul>
DMS-w	<ul style="list-style-type: none"> <li>• Chat Servers</li> </ul>

DMS instance	Connects to
	<ul style="list-style-type: none"> <li>• Contact Server or cluster</li> <li>• Interaction Server</li> <li>• Solution Control Server</li> <li>• Message Server, if required</li> </ul>

### (Optional) Genesys Hub Plug-in for Workspace Desktop Edition configuration

You can specify multiple DMS servers to be used for DMS scalability. If the DMS app name is not specified in the following section, DMS app name from user data of a parent interaction is used.

#### Important

Genesys Hub Plug-in for Workspace Desktop Edition 9.1.007.07 or later is required for configuring the following sections.

Configure the following sections in the WDE application for the corresponding media channels:

Section	Channel	Value	Changes take effect	Type
applebcsession.appname	Channel name for Apple Business Chat	DMS app name. Multiple servers can be specified by using comma.	Immediately	String
whatsappsession.appname	Channel name for WhatsApp	DMS app name. Multiple servers can be specified by using comma.	Immediately	String