



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.

Composer Help

Facebook Block

Contents

- 1 Facebook Block
 - 1.1 Name Property
 - 1.2 Block Notes Property
 - 1.3 Exceptions Property
 - 1.4 Condition Property
 - 1.5 Logging Details Property
 - 1.6 Log Level Property
 - 1.7 Result Property
 - 1.8 Application Property
 - 1.9 Method Name Property
 - 1.10 Method Parameters Property
 - 1.11 Service Timeout Property
 - 1.12 Enable Status Property

Facebook Block

Use the Facebook block to handle incoming Facebook interactions. Facebook handles can be monitored through pre-defined events and messages filtered based on keywords to determine the next action.

The Facebook block has the following properties:

Name Property

Find this property's details under [Common Properties](#).

Block Notes Property

Find this property's details under [Common Properties](#).

Exceptions Property

Find this property's details under [Common Properties](#).

Condition Property

Find this property's details under [Common Properties](#).

Logging Details Property

Find this property's details under [Common Properties](#).

Log Level Property

Find this property's details under [Common Properties](#).

Result Property

Click the down arrow under **Value** and select the variable to hold the raw data (JSON structure) returned by the Social Messaging Server.

Application Property

Specify the Application Name of the Social Messaging Server.

1. Click under **Value** to display the ... button.
2. Click the ... button to open the Application Selection dialog box.
3. Click the down arrow opposite **Type** and select **Literal, Variable, or Configuration Server**.
4. If you selected *Configuration Server*, select the Social Messaging Server application from the drop-down of the **Value** field and click **OK**.
5. If you selected **Literal** or **Variable**, enter the literal or select a variable, and click **OK**.

Method Name Property

Select the method to be invoked. This is a mandatory field and the value can be a variable, literal, or method name enum (SubmitComment, DeleteObject, SubmitPost, SubmitPrivateMessage, SubmitPrivateMessageFromLiveChatSession, ShareObject, ChangeObjectVisibility, LikePublication, EditPublication).

[+] The input and output parameters for the different methods are listed here (mandatory parameters are in bold font).

Method Name	Description	Input Paramaters	Output Paramaters
MergeltxData	<p>This ESP request is used to combine the attached data of two Facebook interactions related to the same post, update attached data of the specified interaction and return the final attached data subset that was added to the specified Facebook interaction.</p> <p>The necessity in such request appears when the routing business process detects that there is more than one interaction related to the same post (for example, a new comment(s) was added while the interaction was sitting in a Genesys</p>	<p>_umsChannel, _facebookMergeltxId</p>	<p>_facebookMergedUserData</p>

Method Name	Description	Input Parameters	Output Parameters
	interaction queue). In this case, instead of delivering two interactions to an agent, the routing business process will try to merge them and deliver updated data to an agent.		
GetComments	This ESP request is used to get Facebook comments for particular post, for a specified time period.	_umsChannel, _umsChannelMonitor, _facebookPostId, _facebookStartTime, _facebookEndTime, _facebookLimit, _facebookOffset, _facebookOrder, UserData (any key)	_facebookXML
SubmitComment	This ESP request is used to publish Facebook comments for a particular post or a reply on a particular comment. The parameter <i>_facebookObjectId</i> is mandatory, and its value is either a post ID or a comment ID. If the parameter value is a comment id, then a reply to a comment will be published to Facebook, otherwise a comment to a post will be published to Facebook.	_umsChannel, _umsChannelMonitor, _facebookPostId, _facebookMessageText, UserData (any key)	_facebookCommentId, _facebookTargetObjectId
DeleteObject	This ESP request is used to delete a Facebook object. This object could be a post, comment, or reply.	_umsChannel, _umsChannelMonitor, _facebookObjectId, UserData (any key)	_facebookObjectId
SubmitPost	This ESP request is used to publish a (new) Facebook post.	_umsChannel, _umsChannelMonitor, _facebookSourceId, _facebookMessageText, _facebookPictureBody, _facebookPictureName, _facebookLink, _facebookDescription, _facebookTimelineVisibility, UserData (any key) Tip An empty post could be published if the <i>_facebookPictureBody</i> or <i>_facebookLink</i> parameters are supplied. Otherwise the <i>_facebookMessageText</i> parameter is mandatory.	_facebookPostId

Method Name	Description	Input Parameters	Output Parameters
SubmitPrivateMessage	This ESP request is used for submitting Facebook unsolicited outbound private messages.	_umsChannel, _umsChannelMonitor, _facebookPMThreadId, _facebookMessageText, UserData (any key)	_facebookPMMessageId
SubmitPrivateMessageFromLiveChatSession	This ESP request is invoked by Social Messaging Server, only for live Facebook ChatSession and should not be used by other clients. As result, the Facebook driver loaded within Social Messaging Server creates a Facebook private message in the specified chat thread.	_umsMsgPlainText, _umsChannelMonitor, _umsAttachedData Important The <i>_umsAttachedData</i> parameter has the following additional parameters: <i>_umsChannelMonitor,</i> <i>_facebookPMThreadId,</i> <i>_facebookLink,</i> <i>_facebookDescription.</i>	_facebookPMMessageId
ShareObject	This ESP request is used to share a Facebook post, photo, link to a web page or an HTTP link to publicly available file.	_umsChannel, _umsChannelMonitor, _facebookObjectId, _facebookMessageText, _facebookDescription	_facebookPostId
ChangeObjectVisibility	This ESP request is used to change Facebook post visibility. However, note that Facebook doesn't allow to hide Comments/Replies through the Facebook API.	_umsChannel, _umsChannelMonitor, _facebookObjectId, _facebookObjectType, _facebookVisibility	_facebookObjectId
LikePublication	This ESP request is used to like and unlike a Facebook post, comment or reply. The object will be liked/unliked on behalf of the publish access token owner specified in the configured monitor specified in the ESP request input parameter <i>_umsChannelMonitor</i> .	_umsChannel, _umsChannelMonitor, _facebookObjectId, _facebookLikeObject	_facebookObjectId
EditPublication	This ESP request is used to edit an existing Facebook post, comment, or reply.	_umsChannel, _umsChannelMonitor, _facebookObjectId, _facebookMessageText	_facebookObjectId

Method Parameters Property

List of input parameters populated automatically based on the selected method in the **Method**

Name property. The automatically populated values can be edited, if required. You can also add the new parameters by clicking the **ADD** button. The Method Parameters (key and value) can be specified either as a literal or variable.

Composer supports special keys such as, FACEBOOK_CHANNEL (equivalent to `_genesys.ixn.interactions[system.InteractionID].udata['_umsChannel']` in the code sample), which can be used to map values for the parameters.

Tip

Refer to the table in the **Method Name Property** section above for more information on custom parameters.

Service Timeout Property

Specify the timeout in seconds to invoke the ESP service on a timeout error. The default value is 10 seconds. You can specify the timeout value either as a literal or variable, but it must be a positive integer.

Enable Status Property

Find this property's details under [Common Properties](#).

[+] Facebook Block Code Sample

```
<state id="Facebook1">
  <onentry>
    <log expr="_sessionid + ': Inside Facebook Block: Facebook1'" />
    <script><![CDATA[
      App_Facebook1.content = {};
      App_Facebook1.content.params = {
        _umsChannel:
_genesys.ixn.interactions[system.InteractionID].udata['_umsChannel'],
        _umsChannelMonitor:
_genesys.ixn.interactions[system.InteractionID].udata['_umsChannelMonitor'],
        _facebookPostId:
_genesys.ixn.interactions[system.InteractionID].udata['_facebookPostId'],
        _facebookMessageText: 'Thanks'
      };
    ]]></script>
    <session:fetch requestid="App_Facebook1['requestid']"
srcexpr="'SocialMessagingServer_854' + '\\CFGSocialMS\\Facebook\\' + 'SubmitComment'"
attach_ixn_data="true" timeout="30" method="'esp'">
      <content _expr="App_Facebook1.content" />
    </session:fetch>
  </onentry>
  <transition event="session.fetch.done">
```

Facebook Block

```
cond="_event.data.requestid==App_Facebook1['requestid']" target="$$_MY_PREFIX_$.Exit1">
  <log expr="'Composer Application:default Block: Facebook1'" />
  <log expr="'Session FETCH DONE'" />
  <log expr="'Data Fetched:' + _event.data.content" />
  <script>storeEvent("Facebook1", _event);</script>
  <script>App_Facebook1['data'] = eval('(' + _event.data.content + ')');</script>
</transition>
</state>
```