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

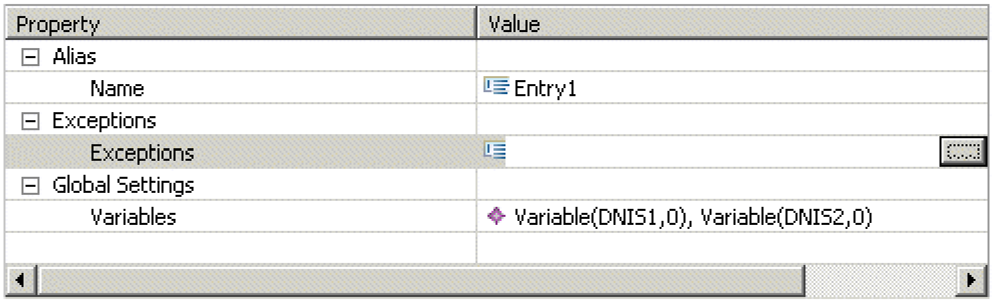
Exception Events

4/3/2025

Contents

Exception Events

Certain callflow exception events and routing exception events application blocks have an Exceptions property. The Properties view for the routing Entry block is shown below as an example.



Property	Value
⊟ Alias	
Name	Entry1
⊟ Exceptions	
Exceptions	...
⊟ Global Settings	
Variables	Variable(DNIS1,0), Variable(DNIS2,0)

Clicking the ... button opens an Exceptions dialog box where you can select events to be supported. The voice Entry block also has an Exceptions property with a different set of exceptions that can be supported.

Exception Support Levels

Exceptions can be configured at two levels:

1. At the top level in the Entry block using the **Exceptions** property.
2. At the individual block-level for local exception handling using the **Exceptions** property. These block level exceptions may not be present in all blocks.

To support any of these exceptions globally throughout your application:

1. Select the Entry block's **Exceptions** property.
2. Select an event from the **Not Supported** pane, and then click the **>** button to move the event to the **Supported** pane.

Note: If the same exception is defined at the top level and the block level, the block-level exception takes precedence for that block.

Callflow Event Handling Guidelines

1. For Main callflow:
    - Always handle the default event handlers--disconnect, error, and all.

    - If selecting a specific error event type, always place it before any other errors. For example, error.badfetch.http must be placed before error.badfetch, which must be placed before the error event in the application.
    - Select specific handlers like error.badfetch only if the application has to handle that exception differently than the generic error or all event handler.

2. For Subcallflows:

    - Unless special processing is required at the local level, it is best to handle events at the global level in the Main callflow.

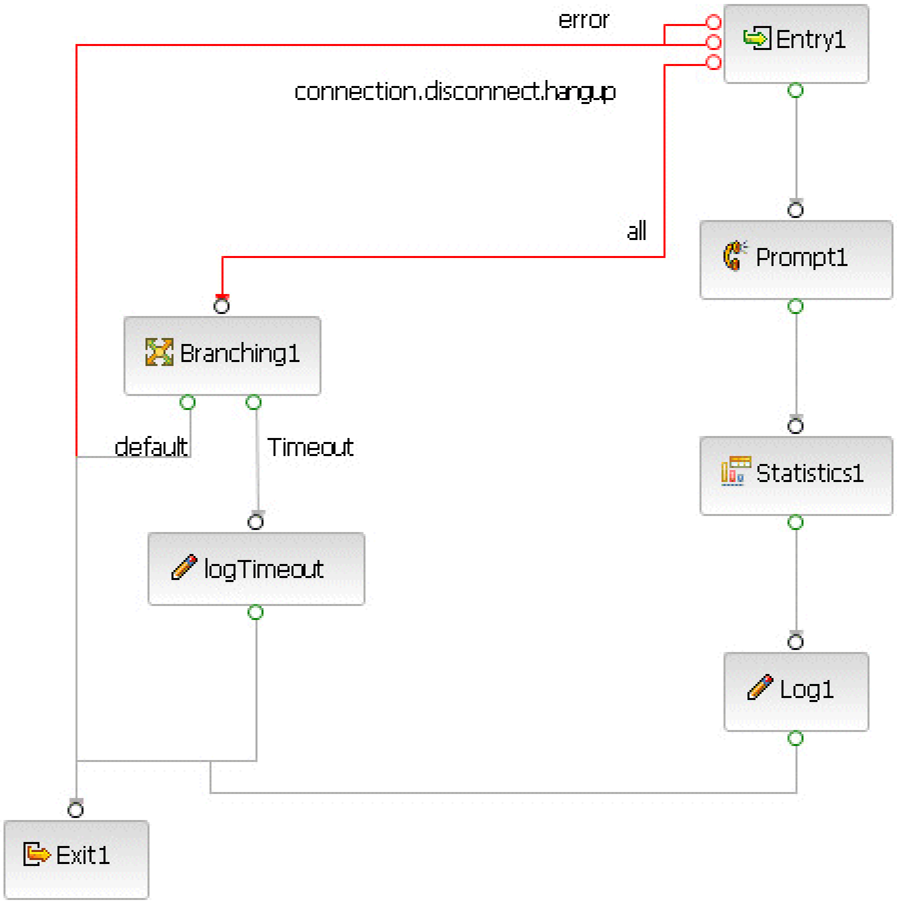
Exception Handling

This section discusses Event Handling and provides some guidelines. The VXML/SCXML interpreters throw exceptions when they encounter errors or conditions for which exceptions are needed so that the condition may be communicated back to the VXML/SCXML application. For example, for voice applications, events such as NOINPUT or NOMATCH in an Input block fall under this category. Composer exposes exceptions at two levels:

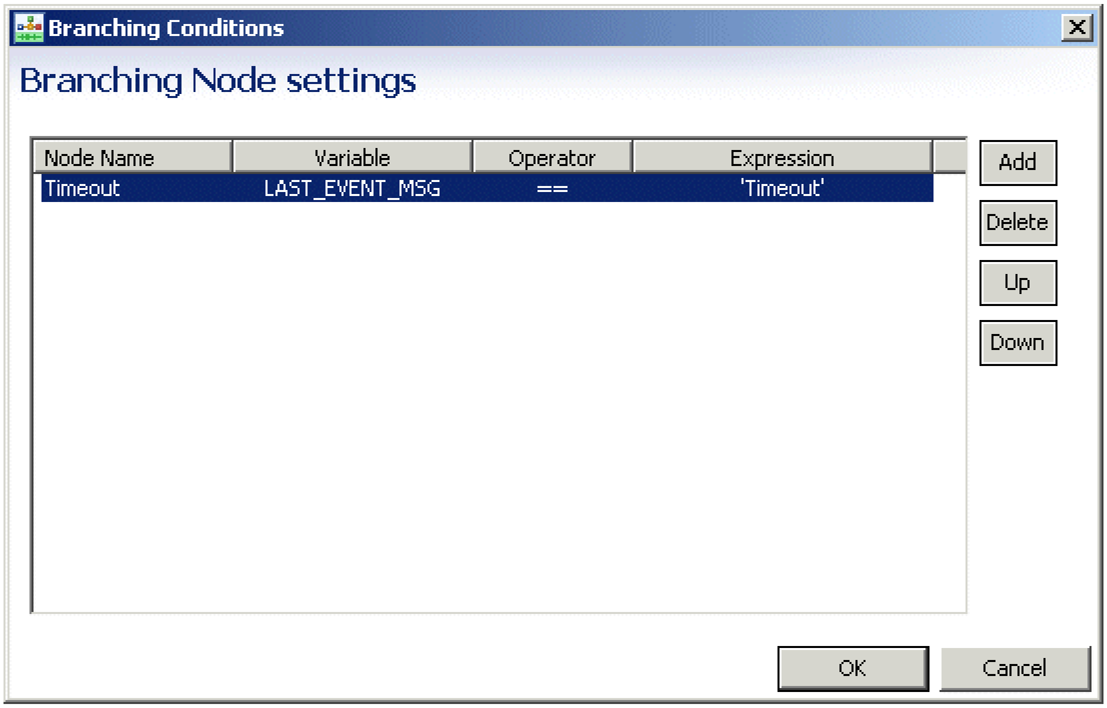
- the block level
- the overall callflow level

Any exceptions exposed at the block level can be handled at the block level itself. If the exception is thrown, control does not wind its way all the way back to the Entry block. For example: the DB Data block exposes a `dbconnectionerror` exception if connection to a database fails. This exception is thrown and handled at the block level itself: `<form id="DBData1">` `<block>` `<data name="DBData1Data" src="../include/dbrequest.aspx" method="post" namelist="AppStateString db_query db_name db_timeout" />` … `<throw event="error.com.genesyslab.composer.dbconnectionerror" messageexpr="DBData1Data.errorMsg" />` `</block>` `<catch event="error.com.genesyslab.composer.dberror">` `<!--handle exception here-->` `</catch>` `</form>` The Entry blocks for routing and voice applications also expose a number of exceptions. These are handled in the Entry block itself. The following are some recommendations about good practices in handling exceptions for callflows.

- For the Main callflow, always handle the default event handlers--`disconnect`, `error`, and `all`.
    - If selecting a specific error event type, always place it before any other errors. For example, `error.badfetch.http` must be placed before `error.badfetch`, which must be placed before the error event in the application.
    - Select specific handlers like `error.badfetch` only if the application has to handle that exception differently than the generic error or all event handler.
- For Subcallflows, it is straightforward to handle exceptions that Composer does not expose directly. Any exceptions that are not listed explicitly are thrown as the `all` exception. A branching block can be added to this exception path and a different execution path can be chosen based on the contents of the exception. The conceptual diagram below shows this approach:



The branch condition will look something like this (variables are described in the Entry block):



Branching Conditions

Branching Node settings

Node Name	Variable	Operator	Expression
Timeout	LAST_EVENT_MSG	==	'Timeout'

Unless special processing is required at the local level, it is best to handle events at the global level in the Main callflow.

Exception Event Descriptions

The table below names and describes Composer exception events for both callflows and workflows. For callflow VXML events, additional information on events can be found in the GVP 8.1 Legacy Genesys VoiceXML 2.1 Reference Manual.

Exception Event Name	Description
all	This is a generic catch-all exception handler that will catch any exception not handled by the Entry block. It should be the last catch handler in the sequence of exceptions for a block.
cancel	The caller has requested to cancel playing of the current prompt. (Available when the Universals property includes Cancel.)
com.genesyslab.composer.toomanynoinputs	Occurs when the number of no inputs exceeds the maximum retries in Menu, Input, and Record blocks, and blocks do not have local noinput exception ports.
com.genesyslab.composer.toomanynomatches	Occurs when the number of no matches exceeds the maximum retries in Menu, Input, and Record blocks, and blocks do not have a local nomatch exception port.
com.genesyslab.externalmessage	For handling asynchronous external events.

	(Available when com.genesyslab.externalevents.enable is set to true.)
connection.disconnect.hangup	The caller hangs up. Applies at any time except during blind transfers.
connection.disconnect.transfer	The call was "blind transferred" to another line and will not return.
error	The platform encountered any error, such as error.noresource.
error.badfetch	A fetch of a resource failed due to semantic errors in the VoiceXML page.
error.badfetch.badxmlpage	The page that was fetched is an invalid XML page.
error.badfetch.grammar.load	The platform failed to load a grammar.
error.badfetch.grammar.syntax	A grammar has a syntax error.
error.badfetch.grammar.uri	The platform failed to fetch a grammar uri.
error.badfetch.http	A fetch of a resource failed, and the platform returned an HTTP Response failure code.
error.com.genesyslab.composer.badfetch	Bad request.
error.com.genesyslab.composer.badgateway	Bad gateway.
error.com.genesyslab.composer.customernotfound	Context Services Identify Customer Block, Suppress Customer Not Found Exception Property: If set to false (default), this exception is raised when no matching customer is found.
error.com.genesyslab.composer.dbconnectionerror	This error indicates that there was an error encountered while attempting to connect to the database.
error.com.genesyslab.composer.dberror	All database errors (other than dbtimeout and dbconnectionerror) result in this exception. The error text associated with this exception will contain the exact error returned by the database.
error.com.genesyslab.composer.dbtimeout	This error indicates that there was a timeout while waiting for query results to be received from the database. The timeout period is controlled by the Timeout property
error.com.genesyslab.composer.dbemptyresultset	This error indicates that the query or a stored procedure execution resulted in an empty result set, i.e., No records were returned. If the application will handle this condition and the exception is not required to be thrown then it can be suppressed using the Suppress Empty Result Set Exception property.
error.com.genesyslab.composer.forbidden	Forbidden plus specific error message from the server.
error.com.genesyslab.composer.invalidkey	This is the event error for handling an invalid key name. For example, the userdata key being accessed by the Interaction Data block is not a valid key.

error.com.genesyslab.com.composer.notautorized	Not authorized.
error.com.genesyslab.com.composer.notfound	Not found.
error.com.genesyslab.composer.operationtimeout	This error occurs when the request for Userdata, AccessNumGet, or Statistics times out at the Iserver.
error.com.genesyslab.composer.receiveerror	This error occurs when the Iserver is down at the udata(getstat,AccessNumget) request.
error.com.genesyslab.composer.recordCapture.failure	Error occurred while saving the recorded file on the Web server side.
error.com.genesyslab.composer.servererror	This error occurs when there is an error in processing the request for the server side blocks such as "Invalid parameters" or "Web server is down." For Context Services blocks: Internal Server Error plus specific error message from the server.
error.com.genesyslab.composer.serviceunavailable	Service unavailable.
error.com.genesyslab.composer.webservice.batchfetch	For attribute descriptions, see the SCXML Language Reference
error.com.genesyslab.customernotfound	For attribute descriptions, see the SCXML Language Reference
error.com.genesyslab.composer.unsupported	This error occurs when the CTI applications designed for CTIC supported features like AccessNumget, Statistics, Interaction-Delete, Replace are called against a SIP Server environment.
error.com.genesyslab.subdialog.maxdepthexceeded	Error when subdialog depth limit is exceeded.
error.connection	This is the base exception for any connection-related error. For example, error.connection.protocol.nnn is thrown when the protocol stack for the connection raised an exception in the case of a bridged transfer. This error can be caught using this base exception.
error.connection.baddestination	The destination URI is malformed in the Transfer block.
error.connection.noauthorization	The caller is not allowed to call the destination, after initiating a transfer using the Transfer block.
error.connection.noresource	The platform could not allocate resources to place the call initiated by the Transfer block.
error.connection.noroute	The platform was not able to route the call to the destination, in a case where the destination URI (phone number) has the correct format.
error.dialog.collect	For attribute descriptions, see the SCXML Language Reference.
error.dialog.continue	For attribute descriptions, see the SCXML Language Reference.

error.dialog.createann	For attribute descriptions, see the SCXML Language Reference.
error.dialog.deleteann	For attribute descriptions, see the SCXML Language Reference.
error.dialog.play	This indicates that an error occurred while trying to perform the <play> request. For attribute descriptions, see the SCXML Language Reference.
error.dialog.playandcollect	This indicates that an error occurred while trying to perform the <playandcollect> request. For attribute descriptions, see the SCXML Language Reference.
error.dialog.playandverify	For attribute descriptions, see the SCXML Language Reference.
error.dialog.playsound	This indicates that an error occurred while trying to perform the <playsound> request. For attribute descriptions, see the SCXML Language Reference.
error.dialog.remote	For attribute descriptions, see the SCXML Language Reference.
error.dialog.setdialogdefaultdest	This indicates that an error occurred while trying to perform the <setdialogdefaultdest> request. For attribute descriptions, see the SCXML Language Reference .
error.dialog.start	This indicates that an abnormal condition occurred while trying to perform the start request. This event will be sent as a result of a timeout of the request as well as problems with the request itself. For attribute descriptions, see the SCXML Language Reference.
error.dialog.stop	This indicates that an error occurred while trying to perform the <stop> request. For attribute descriptions, see the SCXMLLanguage Reference
error.interaction.redirect	For attribute descriptions, see the SCXML Language Reference.
error.msgbased.createmessage	This event is sent when the <createmessage> request has failed for some reason.
error.msgbased.sendmessage	This event is sent when the <sendmessage> request has failed for some reason.
error.noresource	The specified language is not supported by the TTS/ASR server, or the TTS/ASR server/service is down.
error.noresource.recording	Occurs when MCP fails to start the GIR recording for any reason. For callflow VXML events, additional information on events can be found in the GVP 8.1 Legacy Genesys VoiceXML 2.1 Reference Manual.
error.queue.cancel	For attribute descriptions, see the SCXML Language

	Reference.
error.queue.default	For attribute descriptions, see the SCXML Language Reference.
error.queue.query	For attribute descriptions, see the SCXMLLanguage Reference.
error.queue.start	For attribute descriptions, see the SCXML Language Reference.
error.queue.stop	For attribute descriptions, see the SCXML Language Reference.
error.queue.submit	This indicates that an abnormal condition occurred while trying to perform the submit request. This event will be sent as a result of a timeout of the request as well as problems with the request or interaction itself. Busy treatment exceptions are raised as the error.queue.submit exception and not as exceptions listed in individual treatment blocks. See the routing application Target block for more information.
error.queue.update	For attribute descriptions, see the SCXML Language Reference.
error.semantic	See the GVP 8.1 Legacy Genesys VoiceXML 2.1 Reference Manual.
error.session.fetch	An error occurred while trying to perform the fetch of a resource in a workflow.
error.session.start	For attribute descriptions, see the SCXML Language Reference.
error.session.terminate	For attribute descriptions, see the SCXML Language Reference.
error.unsupported	The platform encounters any error of type unsupported, such as error.unsupported.builtin.
error.unsupported.builtin	The built-in grammar is not supported by the platform or the ASR engine.
error.unsupported.format	An unsupported grammar format or audio format is encountered.
error.unsupported.language	The platform does not support the language for either speech synthesis or speech recognition.
error.unsupported.objectname	The requested object is not supported.
error.unsupported.receive	The application’s access to messaging is disabled in MCP. This may be received as a VXML event for <receive> blocks.
error.unsupported.send	The application’s access to messaging is disabled in MCP. This may be received as a VXML event for <send> blocks. Note: This will be applicable to all blocks that use <send>, such as the Interaction Data block.
error.unsupported.transfer	This is the base exception for any unsupported

	transfer settings.
error.unsupported.transfer.blind	The platform (configuration) does not support blind transfers, and the application specifies that it wants to do a blind transfer.
error.unsupported.transfer.consultation	The platform does not support consultative transfers, and the application specifies that it wants to do a consultative transfer.
error.unsupported.transrec.type	The platform does not support the media format.
error.unsupported.uri	The platform does not support the URI format (for example, fax://...).
exit	The caller has asked to exit. (Available when the universals property includes exit).
help	The caller has asked for help. (Available when the universals property includes help).
maxspeechtimeout	The caller input was too long, exceeding the property maxspeechtimeout.
noinput	The application expects voice or DTMF input, but it has received none from the caller within the timeout interval.
nomatch	The caller input something, but it was not recognized.

Exception Events for eServices UCS Blocks

The following eService blocks for Universal Contact Server use the exception events listed below:

- Identify Customer
- Update Customer
- Create Customer

See the individual block topics for the list of supported exceptions.

Exception Event Name	Error Code	Error Description
error.com.genesyslab.composer.badfetch	400	Bad Request.
error.com.genesyslab.composer.notfound	404	Not Found.
error.com.genesyslab.composer.servererror	500	Internal Server Error + specific error message from Composer.
error.com.genesyslab.composer.notext	105	No text found in standard response : <UCS error message>.
error.com.genesyslab.composer.missingparameter	201	Missing parameter name.
error.com.genesyslab.composer.incompatibleparameter	202	Parameter ‘1’ and ‘2’ are not allowed.
error.com.genesyslab.composer.invalidparametertype	203	Incorrect type for parameter

		<parameter_name>, expected type 1 but was type 2.
error.com.genesyslab.composer.invalidparametervalue	204	Incorrect value for parameter <parameter_name>, expected value 1 but was value 2.
error.com.genesyslab.composer.invalidmessagetype	502	Invalid third party message type.
error.com.genesyslab.composer.objectnotfound	510	Object not found in database.
error.com.genesyslab.composer.incorrectsubtype	512	Incorrect subtype for interaction <interaction_name>, expected type 1 but was type 2.
error.com.genesyslab.composer.servererror	701	Unexpected error exception message.
error.com.genesyslab.composer.dberror	710	Connection to database failed.
error.com.genesyslab.composer.serveroverloaded	716	Server overloaded, request rejected.
error.com.genesyslab.composer.noattributes	730	No searchable attribute.
error.com.genesyslab.composer.invalidtenant	732	Invalid Tenant <tenant_name>.