



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

[Log Common Block](#)

# Log Common Block

## Contents

- [1 Log Common Block](#)
  - [1.1 Name Property](#)
  - [1.2 Block Notes Property](#)
  - [1.3 Exceptions Property](#)
  - [1.4 Condition Property](#)
  - [1.5 Label Property](#)
  - [1.6 Logging Details Property](#)
  - [1.7 Log Level Property](#)
  - [1.8 Enable Status Property](#)

Use a Log block to record information about an application. For example, you can log caller-recorded input collected while an application is running or error messages. You can use the Log block for any of the following purposes:

1. Informational – To log the application specific data
2. Error – for logging error details
3. Warning – to flag any warnings
4. Debug – for debugging

The categories in the Log Level property correspond to the above.

When used for a callflow, the Log block writes the log to the Genesys Voice Platform logs (Media Control Platform) using the VoiceXML <log> tag.

The Log block has the following properties:

The Log block has no page exceptions.

## Name Property

Find this property's details under [Common Properties for Callflow Blocks](#) or [Common Properties for Workflow blocks](#).

## Block Notes Property

Find this property's details under [Common Properties for Callflow Blocks](#) or [Common Properties for Workflow Blocks](#).

## Exceptions Property

Find this property's details under [Common Properties for Callflow Blocks](#) or [Common Properties for Workflow Blocks](#).

For callflows, invalid ECMAScript expressions may raise the following exception events: error.semantic. For workflows, invalid ECMAScript expressions may raise the following exception events:

- error.log.ReferenceError
- error.illegalcond.SyntaxError

- `error.illegalcond.ReferenceError`

You can use [custom events](#) to define the ECMAScript exception event handling.

## Condition Property

Find this property's details under [Common Properties for Callflow Blocks](#) or [Common Properties for Workflow Blocks](#).

## Label Property

This property applies to workflows only. It provides meta-data for the logging details.

## Logging Details Property

Find this property's details under [Common Properties for Callflow Blocks](#) or [Common Properties for Workflow Blocks](#).

## Log Level Property

Find this property's details under [Common Properties for Callflow Blocks](#) or [Common Properties for Workflow Blocks](#).

## Enable Status Property

Find this property's details under [Common Properties for Callflow Blocks](#) or [Common Properties for Workflow Blocks](#).