



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.

# Workspace Desktop Edition Developer's Guide

Introducing Extensions

5/2/2025

# Introducing Extensions



**Purpose:** To provide information about how to customize Interaction Workspace by using extensions.

## Contents

- [1 Introducing Extensions](#)
  - [1.1 Introducing Extensions](#)
  - [1.2 Developer Interfaces](#)
  - [1.3 Customization Options Using Extensions](#)
  - [1.4 SDK Dependencies](#)
  - [1.5 Read Next](#)

## Introducing Extensions

You can customize Interaction Workspace by using two approaches:

- **Configuration**-You can change the appearance and behavior of the application by configuring parameters (such as, options, annexes and tasks) in Genesys Administrator or any other Genesys Graphic User Interface (GUI), when applicable. The *Interaction Workspace 8.1 User's Guide* provides detailed information about configuration options.
- **Extensions**-You can extend the out-of-the-box GUI and business logic through development. Activation of an extension is performed through configuration.

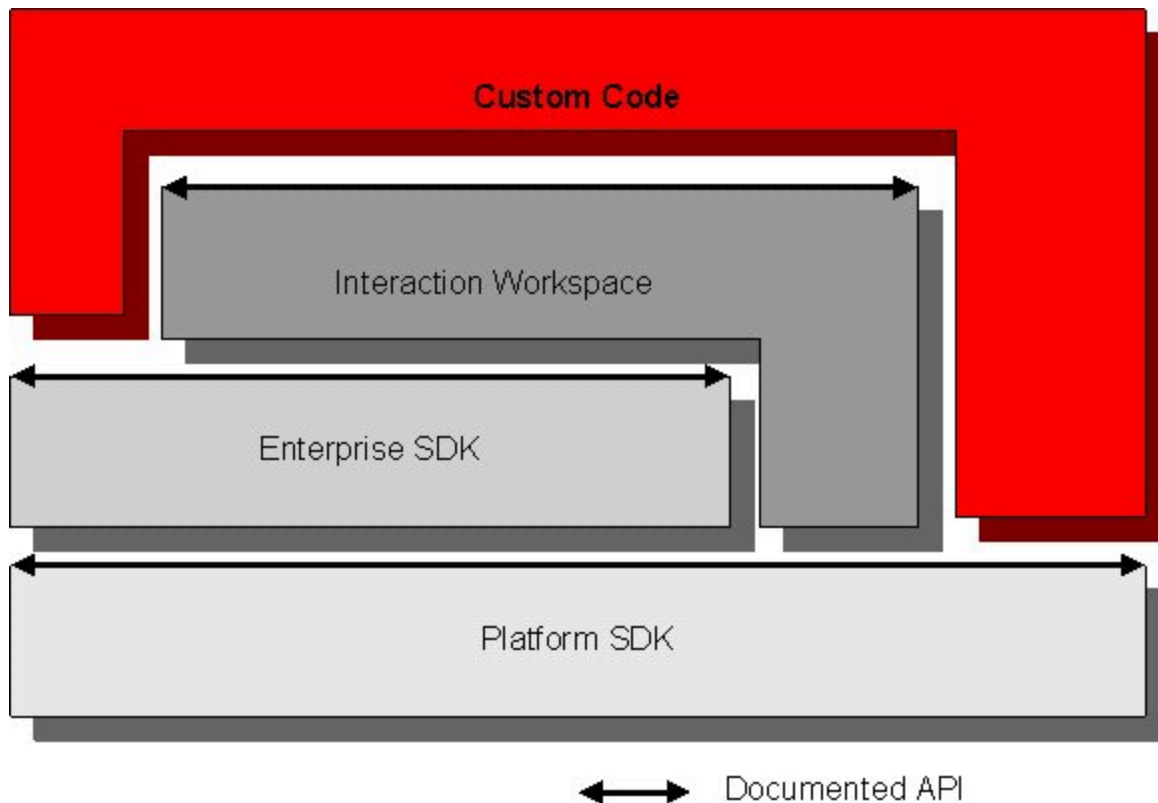
## Developer Interfaces

Build custom modules by using the Genesys developer stack, which is composed of the following SDKs:

- Platform SDK
- Enterprise SDK |Enterprise SDK
- Interaction Workspace API

Enterprise SDK includes its own customization capabilities. Enterprise SDK customization is complementary to Interaction Workspace customization capabilities, so you can use both concurrently.

The custom code lies on top of the developer stack, as shown in the following figure.



Custom Code On Top of Developer Stack

In addition to the Genesys developer stack, any third-party SDK can be used with Interaction Workspace, either directly if it is compatible, or by using an adapter if it is not. The Interaction Workspace API offers a set of services that are used to build internal modules. Services include:

- Event Broker/Aggregator
- Logger
- Extension Manager
- Style Manager
- Brand Manager

## Customization Options Using Extensions

Extensions provide a wide range of customization options, such as:

- Add actions to an existing View or ViewModel module by adding buttons to a toolbar, or items to a contextual menu.
- Add a view to a region of an existing composite view, and populate this view with data from the Genesys back-end (through the appropriate SDK) or through a third-party API.
- Trigger a presentation action on a notification of change in model by using a pop-up window.

- Introduce custom code to execute an action by modifying the action parameters.
- Modify the data-set of the `ViewModel` class before it is rendered in the `View` object. You can do this by using custom logic to remove certain items from a list that is gathered from the back-end.
- Write a custom view based on an existing `ViewModel` class.
- Populate an existing `View` object based on a custom-built `ViewModel` class.

To implement the use cases that are described in the previous paragraph, modify the following levels of the architecture:

- **Model**--Extend Enterprise SDK, extend the Interaction Workspace model, or build new independent model elements. You can use Platform SDK or Enterprise SDK to influence event flow, action flow, or the model itself.
- **ViewModel**--Append new actions and data to an existing `ViewModel` class and add the appropriate logic so that it is rendered in the `View` object.
- **View**--Use an existing `ViewModel` object to rewrite a custom `View` class.
- **Command**--Add, replace, or remove an elementary command in an existing chain of command. For example, the `InteractionVoiceReleaseCall ChainOfCommand` contains only one elementary command (`ReleaseCommand`), but you can add other elementary commands.

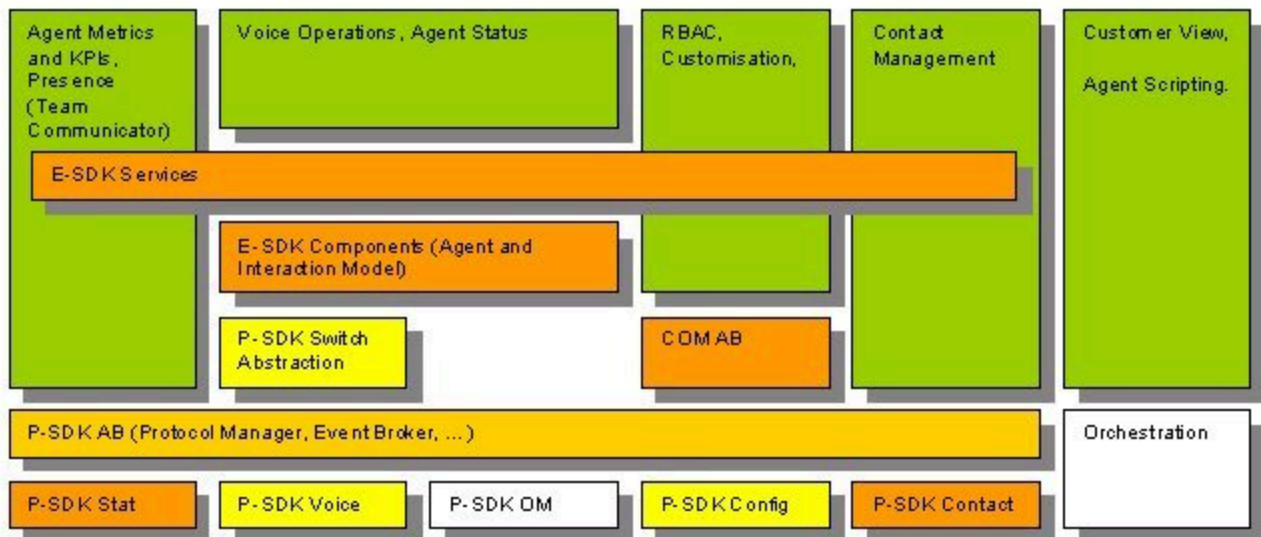
Certain extension use cases support two approaches to implementation, such as:

- Static definition in an XML document
- Dynamic definition by using code that is based on the appropriate Interaction Workspace API.

## SDK Dependencies

The figure below depicts the SDK stack that is embedded in Interaction Workspace 8.1 by using the following color key:

- **Green**--Interaction Workspace functional areas
- **Orange**--SDKs that are used directly by Interaction Workspace
- **Light Orange**--SDKs that are used indirectly by Interaction Workspace
- **Yellow**--SDKs that are not used directly by Interaction Workspace (but are available for a very specific need or customization)
- **White**--SDKs that are not used in the 8.1 version of Interaction Workspace



SDK Dependencies

Read Next

 [About the Extension Samples](#)