



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

Workspace Desktop Edition 8.5.1

7/7/2022

Table of Contents

Workspace Desktop Edition 8.5 Developer's Guide	4
Introduction to Workspace customization	8
Introducing Extensions	12
About the Extension Samples	16
Write Custom Applications	22
Customizing Work Items	29
Advanced Customization	38
Migrate Custom Applications from 8.1 to 8.5	49
Configure Your Custom Theme	62
Frequently Asked Questions	69
Use Customizable Commands	77
Reference for Commands	84
Active Recording	94
Callback	97
Chat	100
E-Mails	113
IM	132
Open Media	136
Outbound	148
Place and Media	152
SMS	164
Supervision	175
UCS	179
Voice	183
Windows	198
Workbins	211
Work Items	216
Customize Views and Regions	223
Best Practices for Views	234
Reference for Windows	246
Interactions Window	248
Gadget Windows	268
Toaster Window	280
Toolbar Window	283
Windows Utilities	296

Workspace Desktop Edition 8.5 Developer's Guide



Purpose: To provide recommendations and code samples that will help you to customize the out-of-the-box version of the **Genesys Workspace Desktop Edition** (Workspace) to meet your business needs.

Description

This .NET Developer's Guide includes the following introductory material that should help you to understand and use the Workspace Desktop Edition API and Extensions:

- [Introducing Workspace Desktop Edition](#)
- [What are Workspace Desktop Edition Extensions](#)
- [About Workspace Desktop Edition Extension Samples](#)
- [Writing Your Custom Applications](#)
- [Best Practices for Creating/Using Views](#)
- [Views with Their Regions](#)
- [Customizable Commands](#)

Limitation: Usage of Enterprise Services that is provided with this release of Workspace Desktop Edition is supported only for the purpose of Workspace Desktop Edition customization.

New in This Document

Release 8.5.000.55

New Pages:

- [Migrate Custom Applications from 8.1 to 8.5](#)
- [Configure Your Custom Theme](#)

Updated Pages:

- [Gadget Windows](#)

- [Interactions Window](#)
- [Toaster Window](#)
- [Toolbar Window](#)
- [Reference for Commands](#)
- [Windows Utilities](#)
- [Reference for Windows](#)
- [About the Extension Samples](#)
- [Chat](#)
- [E-Mails](#)
- [Open Media](#)
- [Place and Media](#)
- [SMS](#)
- [Voice](#)

Release 8.1.300.17

New pages and sections:

- [Callback](#)
- [Supervision](#)
- [Reference for Commands](#)
- [Windows Utilities](#)
- [Reference for Windows](#)

Updated pages:

- [Advanced Customization](#)
- [About the Extension Samples](#)
- [Toolbar Window](#)
- [Gadget Windows](#)
- [Interactions Window](#)
- [Toaster Window](#)
- [Chat](#)
- [E-Mails](#)
- [Open Media](#)
- [Outbound](#)
- [UCS](#)
- [Work Items](#)

- [SMS](#)
- [Voice](#)
- [Work Items](#)

Release 8.1.200.16

New pages and sections:

- [Active Recording](#)

Updated pages:

- [Gadget Windows](#)
- [Interactions Window](#)
- [Toaster Window](#)
- [Toolbar Window](#)
- [Chat](#)
- [E-Mails](#)
- [IM](#)
- [Open Media](#)
- [Outbound](#)
- [Place and Media](#)
- [SMS](#)
- [UCS](#)
- [Voice](#)
- [Windows](#)
- [Work Items](#)

Release 8.1.100.14

New pages and sections:

- [Frequently Asked Questions](#)
- [Advanced Customization](#)
- [Outbound](#)
- [SMS](#)
- [UCS](#)
- [Hiding and Selecting Custom Views](#)
- [Manage SMS Media](#)

Updated pages:

- [Toolbar Window](#)
- [Interactions Window](#)
- [Gadget Windows](#)
- [Toaster Window](#)
- [Windows](#)
- [Voice](#)
- [Open Media](#)
- [E-Mails](#)
- [Chat](#)

Additional Information

In 8.0, this developer's guide was part of the Workspace Desktop Edition API Reference chm.

Introduction to Workspace customization



Purpose: Presents the architecture and design concepts of the Workspace Desktop Edition.

Introducing Workspace Desktop Edition

Workspace Desktop Edition is the customer interaction interface for the Genesys 8 software suite. Workspace Desktop Edition contains many components that you can use to enrich the content of Workspace Desktop Edition with custom views and functionality. You can use the following Workspace Desktop Edition components to customize your interface:

- **Platform SDK**—The low-level SDK that is used to access Genesys back-end servers
- **Enterprise Services**—The high-level SDK that is built on top of Platform SDK and is used to render models and services
- **Workspace Desktop Edition API**—The methods that are used to implement extensions for Workspace Desktop Edition

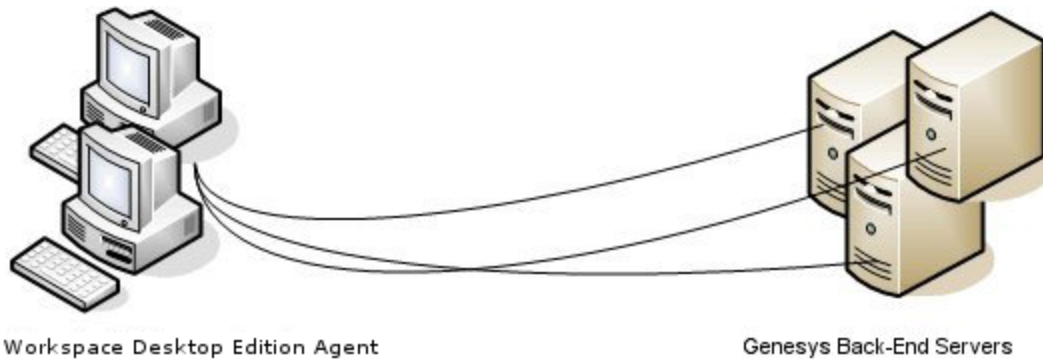
A set of Workspace Desktop Edition Extension Samples is also provided to illustrate the best coding practices for Workspace Desktop Edition customization.

Important

Usage of Enterprise Services that is provided with this release of Workspace Desktop Edition is supported only for the purpose of Workspace Desktop Edition customization.

Architecture

The following figure illustrates a minimal deployment that consists of agent workstations that are connected directly to the Genesys back-end servers.



Simple Client-Server Architecture

Dependency Injection Container Application Block

The Dependency Injection Container Application Block is available for use when you use loosely-coupled applications to develop Workspace Desktop Edition. This lightweight, extensible Dependency Injection container enables developers to build loosely-coupled applications and provides the following advantages:

- Simplified object creation, especially for hierarchical object structures and dependencies
- Abstraction of requirements, enabling developers to specify dependencies at run time or in configuration, and to simplify management of crosscutting concerns
- Increased flexibility by deferring component configuration to the container
- Service location capability, enabling clients to store or cache the container
- Instance and type interception

Important

Genesys Enterprise Services also implements Dependency Injection recommendations, which makes integration easier.

Technical and Design Concepts	Application to Customization
Dependency Injection and Inversion of Control	Used by developers to declare and retrieve alternative implementation of services, models, views, and presenters. Developers can use the Dependency Injection and Inversion of Control when they are developing software.

Loosely-coupled Application Library

The Workspace Desktop Edition is built by using a loosely-coupled application library. This library is used by developers to create composite **Windows Presentation Foundation (WPF)** applications. It is designed to help architects and developers achieve the following objectives:

- Create a complex WPF application from modules that can be built, assembled, and optionally, deployed by independent teams.
- Minimize cross-team dependencies and enable teams to specialize in different areas, such as UI design, business logic implementation, and infrastructure code development.
- Use an architecture that promotes reusability across independent teams.
- Increase the quality of applications by abstracting common services that are available to all the teams.
- Incrementally integrate new capabilities.

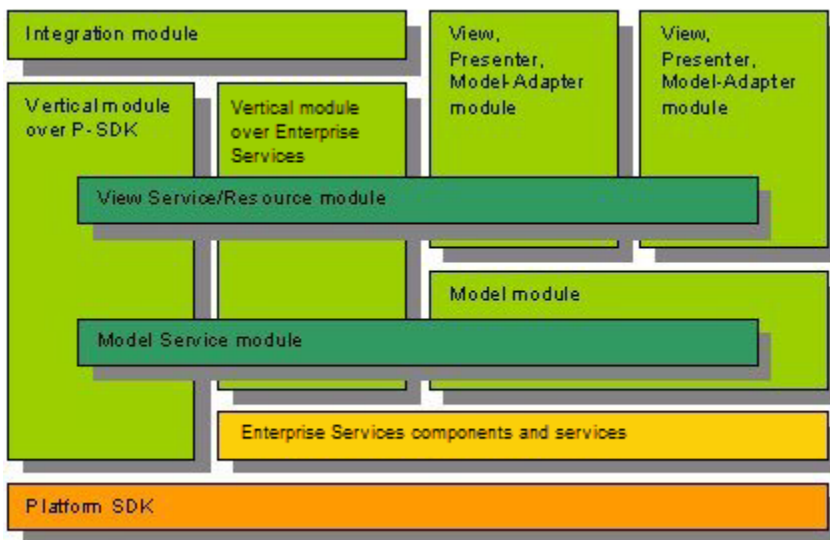
The Workspace Desktop Edition provides guidance and implements patterns that make customization easier. In general, all of these concepts have a common aim, which is to implement loosely-coupled applications and ease extensibility.

Technical and Design Concepts	Description	Application to Customization
Model View ViewModel (MVVM)	<p>The MVVM can separate the responsibilities of the visual display and the responsibilities of user interface state and behavior into different classes named View and View Model, respectively.</p> <ul style="list-style-type: none"> • The View class manages the controls on the user interface. • As a <i>facade</i> on the model, the View Model class provides you with UI-specific state and behavior: <ul style="list-style-type: none"> • It encapsulates the access to the model. • Its public interface is easy to consume from the View (for example, for using data binding). 	You can build an alternate custom View for any given out-of-the-box ViewModel, and you can build an alternate custom ViewModel for any given out-of-the-box View.
Module	A module can be individually developed, tested, and deployed by different teams.	Customization can be implemented by partners or by customers, in a reusable or single-use purpose.
Region Manager	Regions enable a compositional pattern and are commonly used in template layouts and multiple view layouts.	Integration of custom views into out-of-the-box named and documented Regions is simplified, even without knowledge of the application

Technical and Design Concepts	Description	Application to Customization
		construction. For example, in a typical application, a region can be a tab area.

Workspace Desktop Edition Modules

As defined in the software development kit (SDK), a module is a software element that can be individually developed, tested, and deployed by different teams. Workspace Desktop Edition contains several modules that can cover one or several layers of the application. They can contain views, presenters (ViewModel), or models. In general, the functional modules cover the full stack, whereas the service modules focus on a particular layer. By using a modular approach in the application, Genesys is able to provide an SDK to developers who are planning to add customized code into Workspace Desktop Edition. The figure below summarizes the various types of modules in Workspace Desktop Edition.



Types of Modules in Workspace Desktop Edition

Introducing Extensions



Purpose: To provide information about how to customize Workspace Desktop Edition by using extensions.

Introducing Extensions

You can customize Workspace Desktop Edition 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 *Workspace Desktop Edition 8.5 Deployment 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.

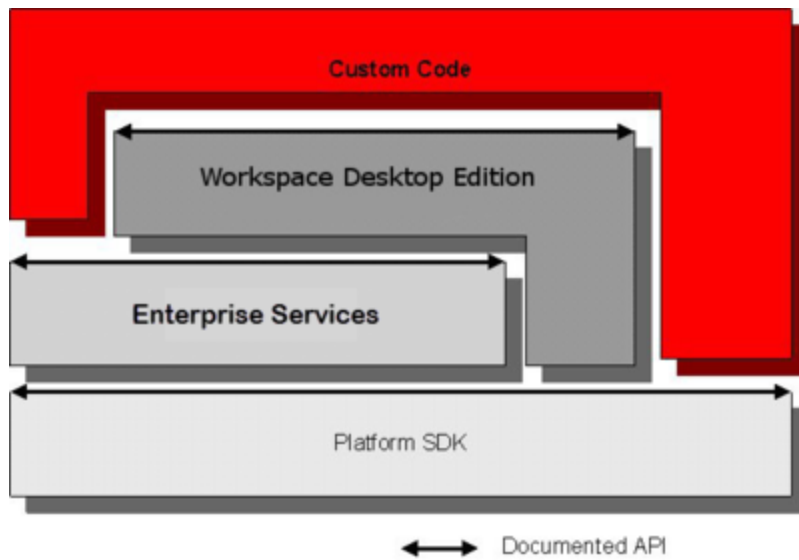
Developer Interfaces

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

- Platform SDK
- Enterprise Services
- Workspace Desktop Edition API

Enterprise Services includes its own customization capabilities. Enterprise Services customization is complementary to Workspace Desktop Edition 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 Workspace Desktop Edition, either directly if it is compatible, or by using an adapter if it is not. The Workspace Desktop Edition 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 Services, extend the Workspace Desktop Edition model, or build new independent model elements. You can use Platform SDK or Enterprise Services 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.

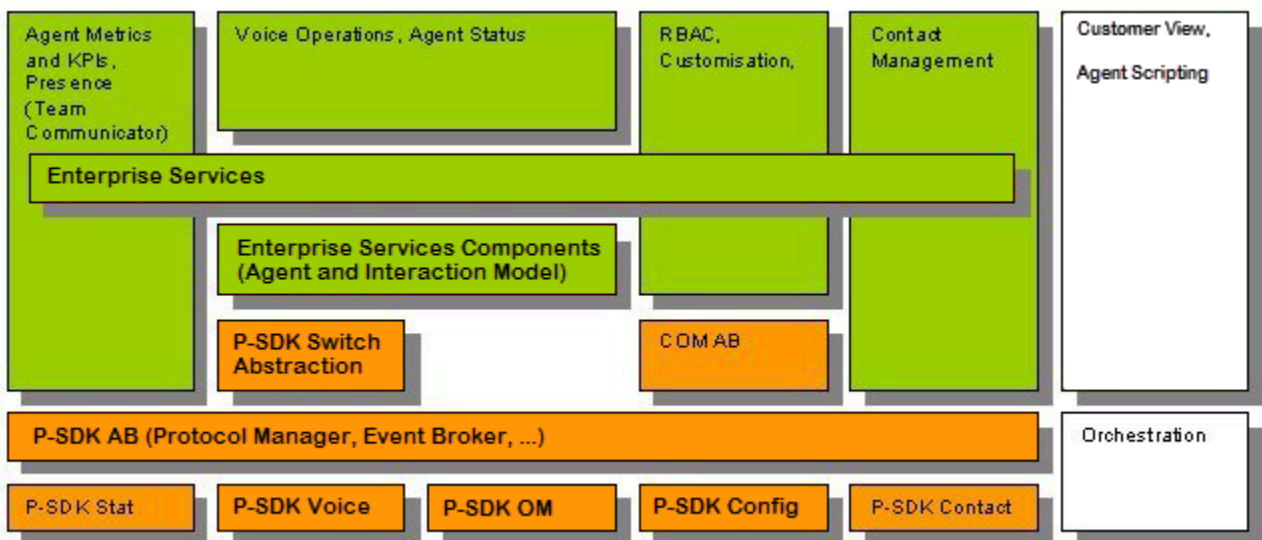
Certain extension use cases support two approaches to implementation, such as:

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

SDK Dependencies

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

- **Green**—Workspace Desktop Edition functional areas
- **Orange**—SDKs that are used directly by Workspace Desktop Edition
- **White**—Components not shipped in out-of-the-box version of Workspace Desktop Edition



SDK Dependencies

About the Extension Samples



Purpose: To provide details about the Workspace Desktop Edition Extension Samples.

Important

Refer to [Deploying the Developer Toolkit on your development workstation](#) in the *Workspace Desktop Edition Deployment Guide* for information about installing the Developer Toolkit.

Introducing Workspace Desktop Edition Extension Samples

The **Genesys Workspace Desktop Edition** (Workspace) Extension Samples provide developers with examples of various use cases. Recommended best practices to modify the out-of-the-box version of Workspace Desktop Edition are used in these code samples. Genesys recommends that you examine the samples before making changes to Workspace Desktop Edition.

You can also download here the [Workspace Desktop Edition Extension Samples](#).

Locating the Extension Samples

The Workspace Desktop Edition Extension Samples are included in the Workspace Desktop Edition API along with the Workspace Desktop Edition API Reference documentation.

The Workspace Desktop Edition API contains everything that a software developer requires for customizing Workspace Desktop Edition, including:

- A Bin directory that contains the Workspace Desktop Edition API
- A Samples directory that contains code samples for developers that demonstrate Genesys' best practices recommendations
- An InteractionWorkspace directory that contains Workspace Desktop Edition application files

- A Doc directory that contains the API Reference as a chm file
- An InteractionWorkspaceDeploymentManager directory that contains the application files that are required to deploy customized code, including the Deployment Manager application—InteractionWorkspaceDeploymentManager.exe. The InteractionWorkspaceDeploymentManager directory also contains a sub-folder:
 - WebPublication—Contains publish.htm and setup.exe files (the bootstrap files for client-side prerequisites)

Detailed Content of the Code Samples

- **Genesyslab.Desktop.Modules.ExtensionSample:** Provides basic examples for this documentation.
 - **MySample:** provides views to extend the ToolbarWorkplaceRegion.
 - **CustomCommand:** Implements a command which prompts a confirmation dialog before executing the ReleaseCall command.
 - **DispositionCodeEx:** provides code to replace the Disposition Code view in the Interaction Window.
- **Genesyslab.Desktop.Modules.InteractionExtensionSample:** Adds a button to the region CaseViewSideButtonRegion and its associated view in the region InteractionWorksheetRegion (Right panel of the interaction window).
- **Genesyslab.Desktop.Modules.CustomWorkItemSample:** Adds a view to manage custom work items.

Use Cases Detailed Online

The following use cases are included in the samples:

- [Creating a New Module](#)
 - [Replacing an Existing View](#)
 - [Creating a New View](#)
 - [Customizing a Command](#)
-

Deploying and Executing the Extension Samples

1. Run the `setup.exe` program to use the wizard to install the Workspace Desktop Edition.
2. Click Next in the Welcome dialog box.
3. Select `Install Workspace Desktop Edition Developer's Toolkit` from the `Select Options` dialog.
4. Click Next. The `Ready to Install` dialog box opens.
5. Click `Install`.
6. When installation completes, the `Installation Complete` window opens. Click `Finished`. For more information, see the online [Workspace Desktop Edition Deployment Guide](#).
7. Verify that the following directories are installed:
 - `C:\Program Files\GCTI\Workspace Desktop Edition\InteractionWorkspace\` (This folder contains all of the required binaries).
 - `C:\Program Files\GCTI\Workspace Desktop Edition\Samples\Genesyslab.Desktop.Modules.ExtensionSample\` (This folder contains the sample solution file).
8. To build and debug your custom module in Workspace Desktop Edition combined with "Workspace Desktop Edition SIP Endpoint" or any Workspace Desktop Edition Plug-in that is delivered by Genesys, such as the Twitter, Facebook, and RSS plug-ins, or one of several Language Packs, follow these steps:
 - Install the corresponding add-on on the workstation used for custom module development.
 - Copy the installed content, maintaining the directory structure, of the add-on into the following directory: `"C:\Program Files\GCTI\Workspace Desktop Edition\InteractionWorkspace\"`.

For example:

- For Workspace Desktop Edition SIP Endpoint, after you install this add-on copy the directory `"C:\Program Files\GCTI\Workspace Desktop Edition\InteractionWorkspaceSIPEndpoint"` into the following location: `"C:\Program Files\GCTI\Workspace Desktop Edition\InteractionWorkspace\"`.
- For the Twitter plug-in, after you install the plug-in, copy the following files `"C:\Program Files\GCTI\Workspace Desktop Edition\Genesyslab.Desktop.Modules.Twitter.dll"` and `"C:\Program Files\GCTI\Workspace Desktop Edition\Genesyslab.Desktop.Modules.Twitter.module-config"` into the following location: `"C:\Program Files\GCTI\Workspace Desktop Edition\InteractionWorkspace\"`, and `"C:\Program Files\GCTI\Workspace Desktop Edition\Languages\Genesyslab.Desktop.Modules.Twitter.en-`

US.xml" into the following location: "C:\Program Files\GCTI\Workspace Desktop Edition\InteractionWorkspace\Languages".

9. To open the Extension Sample in Visual Studio 2008, click the Genesyslab.Desktop.Modules.ExtensionSample.sln solution file.
10. Build the solution. **Note:** Building the solution also copies the content of "C:\Program Files\GCTI\Workspace Desktop Edition\InteractionWorkspace\" to the following location: "C:\Program Files\GCTI\Workspace Desktop Edition\Samples\Genesyslab.Desktop.Modules.ExtensionSample\bin\Debug".
11. Open the project property dialog box, and click the Debug tab.
12. In the Start Action section, select the Start external program option, and in the text field type: C:\Program Files\GCTI\Workspace Desktop Edition\Samples\Genesyslab.Desktop.Modules.ExtensionSample\bin\Debug\InteractionWorkspace.exe (the default debug build location).
13. Run the sample code to see the customization.
14. Follow the steps in the section titled [Deploying Your Custom Module into the Genesys Out-Of-The-Box Application](#) to deploy the customization samples.

Write Custom Applications



Purpose: To provide information on how to implement the basic functions that are needed to write and deploy simple customized applications for Workspace Desktop Edition.

Using Workspace Desktop Edition API

You can use the Workspace Desktop Edition API to write your custom-built .NET applications. After you have reviewed the information in this section, it might be useful to refer to the [Extension Samples](#), upon which this document was based. The samples are working applications which contain and execute the functionality outlined here. When you are ready to write more complex applications, refer to the classes and methods described in the Workspace Desktop Edition API Reference.

Important

The Workspace Desktop Edition API Reference is available in the installation package of the Workspace Desktop Edition.

Basically, the API provides you with several managers which enables you to register your customized interfaces and behaviors:

- `IObjectContainer`—manages mapping of types and components (the same as `IUnityContainer`).
- `IViewManager`—manages the references on out-of-the-box views and regions.
- `ICommandManager`—manages the commands of your applications.
- `IViewEventManager`—manages the events launched by the views.

Through these managers, you can handle components detailed in [Views and Regions](#) to create views, regions, commands, or replace the existing objects. Additional manager interfaces provide access to the modules of the Workspace Desktop Edition. You can find these interfaces in the feature's associated namespace. For instance, you can retrieve the `Genesyslab.Desktop.Modules.Core.Model.Broadcast.IBroadcastManager` interface as follow:

[C#]

```
public MyNewSampleClass(IObjectContainer container, ILogger log)
{
    IBroadcastManager broadcastManager = container.Resolve<IBroadcastManager>();
    //...
}
```

Because the Workspace Desktop Edition is agent-oriented, the `Genesyslab.Desktop.Modules.Core.Model.Agents.IAgent` interface is also an important entry

point, which can be retrieved in the same way:

[C#]

```
public MyNewSampleClass(IObjectContainer container, ILogger log)
{
    IAgent myAgent = container.Resolve<IAgent>();
    //...
}
```

Creating a New Module

The module is a top level component that can be loaded in the Workspace Desktop Edition.

The Workspace Desktop Edition application is a modular application, so Genesys recommends that you create a new module which manages your customized behavior or appearance. The module can register and add new custom views and regions, or manage commands. The following steps explain how to create your new module which will handle the customized components. It is based on the `Genesyslab.Desktop.Modules.ExtensionSample`, which contains the new class `ExtensionSampleModule`. The `ExtensionSampleModule` class inherits from the `IModule` interface.

1. Create a new assembly .dll, and then a class that inherits from the `IModule` interface. In the following code sample, the module keeps a reference on the main manager instances that enable the customization.

[C#]

```
// File: ExtensionModule.cs
namespace Genesyslab.Desktop.Modules.ExtensionSample
{
    public class ExtensionSampleModule : IModule
    {
        readonly IObjectContainer container;
        readonly IViewManager viewManager;
        readonly ICommandManager commandManager;
        readonly IViewEventManager viewEventManager;
        public ExtensionSampleModule(IObjectContainer container,
            IViewManager viewManager,
            ICommandManager commandManager,
            IViewEventManager viewEventManager)
        {
            this.container = container;
            this.viewManager = viewManager;
            this.commandManager = commandManager;
            this.viewEventManager = viewEventManager;
        }
        public void Initialize()
        {
            // Use the Initialize method to register your code, as described in further sections
            // In this example, the default DispositionCodeView is replaced with the DispositionCodeExView
            container.RegisterType<IDispositionCodeView, DispositionCodeExView>();
        }
    }
}
```


Important

The purpose of the current module is to replace an existing view with a new view (detailed below).

2. You must modify the Modules section of the `ExtensionSample.module-config` file to include your new module. This can be done in two ways:

- **Method 1** You add the following content to the modules section:
[XML]

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <configSections>
    <section name="modules"
      type="Microsoft.Practices.Composite.Modularity.ModulesConfigurationSection, Microsoft.Practices.Composite" />
    </configSections>
    <modules>
      <module assemblyFile="Genesyslab.Desktop.Modules.ExtensionSample.dll"
        moduleType="Genesyslab.Desktop.Modules.ExtensionSample.ExtensionSampleModule"
        moduleName="ExtensionSampleModule">
        <dependencies>
          <dependency moduleName="WindowsModule" />
        </dependencies>
      </module>
    </modules>
  </configuration>
```

- **Method 2** You can load the module according to a task presence in the tasks section, as shown here:
[XML]

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <configSections>
    <section name="tasks"
      type="Genesyslab.Desktop.Infrastructure.Config.TasksSection, Genesyslab.Desktop.Infrastructure" />
    <section name="modules"
      type="Microsoft.Practices.Composite.Modularity.ModulesConfigurationSection, Microsoft.Practices.Composite" />
    </configSections>
    <tasks>
```

```
<task name="InteractionWorkspace.ExtensionSample.canUse"
  clickOnceGroupsToDownload="ExtensionSample"
  modulesToLoad="ExtensionSampleModule" />
</tasks>
<modules>
<module assemblyFile="Genesyslab.Desktop.Modules.ExtensionSample.dll"
  moduleType="Genesyslab.Desktop.Modules.ExtensionSample.ExtensionSampleModule"
  moduleName="ExtensionSampleModule"
  startupLoaded="false" />
</modules>
</configuration>
```

Customizing Views and Regions

To customize your application, you can also create customized views and regions as described in [Customize Views and Regions](#). The list of covered use cases is the following:

- [Replacing an Existing View](#)
- [Creating a New View](#)

Customizing a Command

To customize your application, you can also create customized commands in the existing chains of commands or use existing commands in your views. Help about using the commands is provided in the [Use Customizable Commands](#) page, which also links all the available commands. The list of covered use cases is the following:

- [Creating a Command](#)
- [Inserting a Command in a Chain](#)
- [Multiple Commands and Overlapping](#)

Deploying Your Custom Module into the Genesys Out-Of-The-Box Application

To deploy your custom module in a Click-Once deployment environment:

1. Write your custom code by using the provided APIs.
2. Compile the project as an assembly.
3. Unit test the project using a Unit Test framework or a simple test application.
4. Package the custom assembly with the Genesys out-of-the-box Workspace Desktop Edition (by using Deployment Manager if Click-Once is used, or by using a custom IP).
5. In the Workspace Desktop Edition application in Management Framework, configure a mapping between any custom tasks and the custom assembly.
6. Use the Genesys RBAC model to assign the custom task to users or to an access group of users.
7. Restart the Genesys Workspace Desktop Edition.

For all other types of deployment, use the following steps to load and execute your custom assembly in Workspace Desktop Edition:

1. Add your .dll file to the same directory as the one containing the InteractionWorkspace.exe file.
-

2. Create a new `<module>` section in the `ExtensionSample.module-config` file. See the examples in the previous section [Creating a New Module](#).

Customizing Work Items



Purpose: To provide information about the customization of WorkItems.

Before You Start

Work items are interactions which are not predefined in Workspace Desktop Edition, in opposition to e-mail, instant messaging, voice, and chat interactions. You can define these work items to extend the Workspace Desktop Edition Specificities; for instance, if your solution integrates sms, or fax, or additional media, you can define new work items to handle these interactions. The following step-by-step tutorial explains how to configure your solution to support work items, and then, how to customize your Workspace Desktop Edition to integrate them. The code snippets are extracted from the `Genesyslab.Desktop.Modules.CustomWorkItemSample`, available in [the Workspace Desktop Edition Extension Samples](#).

Set Up System Configuration

First of all, set up the System Configuration as described in [Enabling an agent to use Workitems to handle custom media types](#) of the Deployment Guide to create a new Media Type.

Displaying the content of Workitems

To display the content of work item interactions, you need to implement a customized view as explained in [Customize Views and Regions](#). The `Genesyslab.Desktop.Modules.CustomWorkItemSample.Window.Content` shows how to create a new view handling this task:

- `ICustomWorkItemViewModel.cs` and `ICustomWorkItemView.cs` implement the MVVM pattern.
- `CustomWorkItemViewModel.cs` manages the workitem interaction as `IInteractionOpenMedia`.
- `CustomWorkItemView.xaml` and `CustomWorkItemView.xaml.cs` display the interaction's data and implement the `ICustomWorkItemView` interface, in addition to using the model.

In `CustomWorkItemView.xaml.cs`, the `CustomWorkItemView` instance is initialized with the model which contains the open media interaction.

[C#]

```
// file CustomWorkItemView.xaml.cs
public partial class CustomWorkItemView : UserControl, ICustomWorkItemView
{
    public CustomWorkItemView(ICustomWorkItemViewModel customWorkItemViewModel)
    {
        Model = customWorkItemViewModel;
        InitializeComponent();
        //...
    }
    //...
}
```

At creation, this class retrieves the open media interaction from the application's context dictionary, as follows:

[C#]

```
// file CustomWorkItemView.xaml.cs
///IView members
public object Context { get; set; }
public void Create()
{
    Model.Interaction = contextDictionary.TryGetValue("Interaction") as IInteractionOpenMedia;
}
```

Then, you can bind the properties in your XAML code. For instance, in the CustomWorkItemSample, the content grid displays a label media type and the associated value, defined as follows: **[XAML]**

```
<!-- file CustomWorkItemView.xaml -->
<Label localization:Translate.Uid="Windows.CustomWorkItemView.MediaType"
HorizontalAlignment="Left" VerticalAlignment="Center" Grid.Column="0" Grid.Row="0"
Content="{localization:Translate Default=Media Type:}"/>
<Label HorizontalAlignment="Left" VerticalAlignment="Center" Grid.Column="1" Grid.Row="0"
Content="{Binding Interaction.EnterpriseInteractionCurrent.IdType.SubMediaType}"/>
```

The CustomWorkItemSampleModule class implements the IModule interface and adds the new customized view to the list of active views, as shown here in the RegisterViewsAndServices() method:

[C#]

```
// file CustomWorkItemSampleModule.cs
protected void RegisterViewsAndServices()
```

```
{
    //...
    container.RegisterType<ICustomWorkItemViewModel, CustomWorkItemViewModel>();
    container.RegisterType<ICustomWorkItemView, CustomWorkItemView>();
}
```

Handle the work item

Using [Work Items](#) as described in [Inserting a Command in a Chain](#), you can perform the following actions on work items:

- Accept
- Decline
- Mark done
- Attach data
- Search and/or move to workbin
- And so on.

Or, you can use the Open Media Service of the Enterprise Services API to perform additional actions (see [Advanced Customization](#)). In that case, you must create your commands. The `CustomWorkItemToolBarView.xaml` in `Genesyslab.Desktop.Modules.CustomWorkItemSample.Toolbar` implements a stop processing button, by using the `CustomWorkitemStopProcessingCommand` command created in the `Genesyslab.Desktop.Modules.CustomWorkItemSample.CustomWorkitemCommand.cs` file. The creation of this command follows the guidelines of [Creating a Command](#). At first, the sample implements the `IElementOfCommand` interface which retrieves the `Genesyslab.Enterprise.Services.IOpenMediaService`, as shown here:

[C#]

```
// file CustomWorkitemCommand.cs
class CustomWorkitemCommand : IElementOfCommand
{
    protected readonly IUnityContainer container;
    protected ILogger log;
```

```

protected IOpenMediaService openMediaService;
protected const int timeout = 10000;
public CustomWorkitemCommand(IUnityContainer container)
{
    this.container = container;
    this.log = container.Resolve<ILogger>();
    this.log = log.CreateChildLogger("CustomWorkitemCommand");
    IEnterpriseServiceProvider enterpriseServiceProvider = container.Resolve<IEnterpriseServiceProvider>();
    this.openMediaService = enterpriseServiceProvider.Resolve<IOpenMediaService>("openmediaService");
}
public virtual string Name { get; set; }
#region IElementOfCommand Members
public virtual bool Execute(IDictionary<string, object> parameters, IProgressUpdater progressUpdater)
{
    return false;
}
#endregion
}

```

The `CustomWorkitemCommand` class is a base class for further specific commands. In this example, it uses the Open Media Service to stop processing the interaction, as shown in the following code snippet.

[C#]

```

// file CustomWorkitemCommand.cs
class CustomWorkitemStopProcessingCommand : CustomWorkitemCommand
{
    public CustomWorkitemStopProcessingCommand(IUnityContainer container) : base(container) { }
    public override bool Execute(IDictionary<string, object> parameters, IProgressUpdater progressUpdater)
    {
        log.Info("CustomWorkitemStopProcessingCommand");
        IInteractionOpenMedia interactionOpenMedia =
            parameters.TryGetValue("CommandParameter") as IInteractionOpenMedia;
        try
        {
            if ((interactionOpenMedia != null)
                && (interactionOpenMedia.EntrepriseOpenMediaInteractionCurrent != null))
            {
                if (!interactionOpenMedia.EntrepriseOpenMediaInteractionCurrent.IsInWorkflow)
                {
                    return false;
                }
            }
        }
    }
}

```

```

        }
        openMediaService.StopProcessing(interactionOpenMedia.EntrepriseOpenMediaInteractionCurrent,
            parameters.TryGetValue("Reason") as KeyValueCollection,
            parameters.TryGetValue("Extensions") as KeyValueCollection);
    }
    return false;
}
catch (Exception exp)
{
    log.Error("CustomWorkitemStopProcessingCommand StopProcessing, Exception "
        + interactionOpenMedia, exp);
    return true;
}
}
}
}

```

The `Genesyslab.Desktop.Modules.CustomWorkItemSample.CustomWorkItemSampleModule` implements the `IModule` interface and is responsible for adding the new command to the chain of commands.

[C#]

```

/// file CustomWorkItemSample.cs
void RegisterCommands()
{
    log.Debug("RegisterCommands()");
    ICommandManager commandManager = container.Resolve<ICommandManager>();
    commandManager.AddCommandToChainOfCommand("InteractionCustomWorkitemStopProcessing",
        new List<CommandActivator>() { new CommandActivator() {
            CommandType = typeof(CustomWorkitemStopProcessingCommand) ,
            Name="StopProcessing" } });
}
}

```

In the `CustomWorkItemToolBarView` class, the event-handler of the Stop Processing button retrieves and executes the command:

[C#]

```

// File: CustomWorkItemSample.Windows.ToolBarView.cs
private void StopProcessingButton_Click(object sender, System.Windows.RoutedEventArgs e)
{

```

```

IChainOfCommand Command = container.Resolve<ICommandManager>().
    GetChainOfCommandByName("InteractionCustomWorkitemStopProcessing");
Utils.ExecuteAsynchronousCommand(Command,
    new Dictionary<string, object>() { { "CommandParameter", Model.Interaction } },
    StopProcessingButton);
}
    
```

Customizing Workitem Icons

As an additional customization step for new work items, you can create icons to facilitate the identification of the information related to work items. In that purpose, you need to create or edit a dictionary file, such as for instance, the `Genesyslab.Desktop.Modules.CustomWorkItemSample.en-US.xml` xml file in the `Genesyslab.Desktop.Modules.CustomWorkItemSample` project. Declare your new icon as follows:

```
<Value Id="<myworkitemobjectId>.<workitemchannel>" ImageUrl=<ImagePath> IcoUrl=<IcoPath> />
```

- where `myworkitemobjectId` is the object ID customized in the Workspace Desktop Edition.

Object Id	Customization example
Media Channel Icon	
Channel.OpenMedia.WorkItem.Image	<Value Id="Channel.OpenMedia.WorkItem.Image.customworkitem" ImageUrl="<ImagePath>" />
Interaction Icon	
InteractionWorkItem.Image	<Value Id="InteractionWorkItem.Image.customworkitem" Ico="<IcoPath>" ImageUrl="<ImagePath>"/>
Transfer Button	
Windows.ToolbarWorkitemView.ButtonTransfer	<Value Id="Windows.ToolbarWorkitemView.ButtonTransfer.customworkitem" Tooltip="Custom WorkItem Transfer" ImageUrl="<ImagePath>"/>
Transfer Menu	
Windows.ToolbarWorkitemView.MenuItemOneStepTransferDialer	<Value Id="Windows.ToolbarWorkitemView.MenuItemOneStepTransferDialer.customworkitem" Text="Custom Workitem _Transfer" Tooltip="Custom Workitem Transfer"

Object Id	Customization example
	ImageUrl="<ImagePath>" />
'Move to Workbin' button	
Windows.ToolbarWorkitemView.ButtonMoveToWorkbin	<Value Id="Windows.ToolbarWorkitemView.ButtonMoveToWorkbin.<workitemchannel>"ToolTip="Custom Move to workbin" ImageUrl="<ImagePath>"/>
New Custom Button in WorkItem ToolBar	
Windows.ToolbarCustomWorkItemView.<ButtonName>	<Value Id="Windows.ToolbarCustomWorkItemView.Button1" ToolTip="Stop Processing" Text="Stop Processing"/>
Interaction icon in the Interaction History	
Contacts.ContactHistoryView.InteractionWorkItem	<Value Id="Contacts.ContactHistoryView.InteractionWorkItem.<workitemchannel>" ImageUrl="<ImagePath>"/>
Interaction Icon in Workin management	
InteractionWorkItem.MediaType.Image	<Value Id="InteractionWorkItem.MediaType.Image.<workitemchannel>" ImageUrl="<ImagePath>"/>

For instance, here is the Genesyslab.Desktop.Modules.CustomWorkItemSample.en-US.xml xml file in the Genesyslab.Desktop.Modules.CustomWorkItemSample project, which customize label, images, and icons.

[XML]

```
<?xml version="1.0" encoding="utf-8" ?>
<Dictionary EnglishName="English" CultureName="English" Culture="en-US" CustomDictionary="true">
  <Value Id="Channel.OpenMedia.WorkItem.Image.customworkitem"
    ImageUrl="pack://application:,,,/Genesyslab.Desktop.Modules.CustomWorkItemSample;component/Images/CustomWorkItem.png" />
  <Value Id="Windows.ToolbarCustomWorkItemView.Button1" ToolTip="Stop Processing" Text="Stop Processing"/>
  <Value Id="Windows.CustomWorkItemView.MediaType" Content="Media Type:" />
  <Value Id="Windows.CustomWorkItemView.InteractionType" Content="Interaction Type:" />
  <Value Id="Windows.CustomWorkItemView.InteractionSubType" Content="Interaction Sub-Type:" />
  <Value Id="Windows.CustomWorkItemView.MyCustomContent" Content="My custom content" />
  <Value Id="InteractionWorkItem.Image.customworkitem"
    Ico="pack://application:,,,/Genesyslab.Desktop.Modules.CustomWorkItemSample;component/Images/CustomWorkItem.ico"
    ImageUrl="/Genesyslab.Desktop.Modules.CustomWorkItemSample;component/Images/CustomWorkItem.png"/>
  <Value Id="Windows.ToolbarWorkitemView.ButtonTransfer.customworkitem" ToolTip="Custom WorkItem Transfer"
    ImageUrl="/Genesyslab.Desktop.Modules.CustomWorkItemSample;component/Images/TransferCustomWorkItem.png"/>
```

```
<Value Id="Windows.ToolbarWorkitemView.ButtonMoveToWorkbin.customworkitem"
  Tooltip="Move To Workbin In Progress"
  ImageUrl="/Genesyslab.Desktop.Modules.CustomWorkItemSample;component/Images/CustomMoveToWorkbin.png"/>
<Value Id="Windows.ToolbarWorkitemView.MenuItemOneStepTransferDialer.customworkitem"
  Text="Custom Workitem _Transfer"
  Tooltip="Custom Workitem Transfer"
  ImageUrl="/Genesyslab.Desktop.Modules.CustomWorkItemSample;component/Images/TransferCustomWorkItem.png" />
<Value Id="Contacts.ContactHistoryView.InteractionWorkItem.customworkitem"
  ImageUrl="/Genesyslab.Desktop.Modules.CustomWorkItemSample;component/Images/MediaTypeCustomWorkItem.png"/>
<Value Id="InteractionWorkItem.MediaType.Image.customworkitem"
  ImageUrl="pack://application:,,,/Genesyslab.Desktop.Modules.CustomWorkItemSample;component/Images/MediaTypeCustomWorkItem.png"/>
</Dictionary>
```

Advanced Customization

Purpose: To provide information about the advanced customization with the Enterprise Service API.

Get the Enterprise Service API Reference

The Enterprise Services are core components used by the modules, views, and resources to connect to Genesys Servers and maintain the information flow consistent with the state of Workspace Desktop Edition. The main entry point is available through the `EnterpriseService` property of the `Genesyslab.Desktop.Modules.Core.Model.Agents.IAgent` interface. This interface enables you to access all of the available Enterprise services. All of these services handle the core objects that Interaction workspace creates and displays. Modifications to these objects through Workspace Desktop Edition API should be fine; however, if you create new instances or alter objects through the Enterprise API, your customization is responsible for maintaining the information flow consistent with the data displayed in Workspace Desktop Edition. The following table contains the list of downloadable API References available:

Workspace Desktop Edition Release Version	Enterprise Service Release Version	Release Date	Enterprise Service CHM
8.5.000.55	8.5.000.18	04/17/2013	Download file

Warning

If you encounter difficulties with opening the .chm files, please check the [known issues and solutions of Dr. Explain](#).

Get EnterpriseService

The main entry point is available through the `EnterpriseService` property of the `Genesyslab.Desktop.Modules.Core.Model.Agents.IAgent` interface. The `Resolve` methods of the `IEnterpriseServiceProvider` simplify the retrieval of a service instance.

[C#]

```
public MyNewSampleClass(IUnityContainer container, ILogger log)
{
```

```

IAgent myAgent= container.Resolve<IAgent>();
IEnterpriseServiceProvider enterpriseService = myAgent.EntrepriseService;
//...
INameService nameService = enterpriseService.Resolve<INameService>("key");
}

```

- Where Name is the service name, and key is the mapping key that is predefined in the native source of the Enterprise API.

Service Name	Service Key	Associated Protocols<ref>Protocols can be used when you are managing channels. </ref>
IAgentService	agentService	<ul style="list-style-type: none"> • AgentProtocolRequest-"agent" • DeviceProtocolRequest-"device"
IChannelService	channelService	<i>none</i>
IDeviceService	deviceService	<ul style="list-style-type: none"> • DeviceProtocolRequest-"device"
IIdentityService	identityService	<ul style="list-style-type: none"> • OpenMediaProtocolRequest-"openmedia"
IIMService	IMService	<ul style="list-style-type: none"> • VoiceProtocolRequest-"voice" • DeviceProtocolRequest-"device" • IMProtocolRequest-"im"
IContactService	contactService	<ul style="list-style-type: none"> • ContactProtocolRequest -- "contacts"
IInteractionService	interactionService	<i>none</i>
IChatService	chatService	<ul style="list-style-type: none"> • OpenMediaProtocolRequest-"openmedia"

Service Name	Service Key	Associated Protocols<ref>Protocols can be used when you are managing channels .</ref>
		<ul style="list-style-type: none"> WebMediaProtocolRequest -"webmedia"
IOpenMediaService	openmediaService	none
IMonitorService	monitorService	<ul style="list-style-type: none"> OpenMediaProtocolRequest-"openmedia"
IWorkbinService	workbinService	none <ul style="list-style-type: none"> OpenMediaProtocolRequest-"openmedia"
IPSTService	PSTService	none
ICampaignService	campaignService	none
IOutboundService	outboundService	none

Additional Entry Points

Workspace Desktop Edition API provides additional entry points through properties in the specific classes that are listed in the table below:

Class Name	Property	Description
Genesyslab.Desktop.Modules.OpenMedia.Model. Agents.IAgentMultimedia	EnterpriseAgent	IAgent instance which contains the agent data.
Genesyslab.Desktop.Modules.Core. Model.Interactions.IInteraction	EnterpriseInteractionCurrent	Current interaction processed by Workspace Desktop Edition.
IList<Genesyslab.Enterprise.Model.Interaction.IInteraction>	EnterpriseInteractions	The history of interactions.
Genesyslab.Platform.Commons.Protocols.IMessage	EnterpriseLastInteractionEvent	The last interaction event.

Class Name	Property	Description
Genesyslab.Desktop.Modules.OpenMedia. Model.Interactions.Chat.IInteractionChatCommon	EnterpriseChatInteractionCurrent	Current chat interaction processed by Workspace Desktop Edition.
Genesyslab.Desktop.Modules.OpenMedia. Model.Interactions.Email.IInteractionEmail	EnterpriseEmailAttachments	E-mail attachments.
Genesyslab.Desktop.Modules.OpenMedia. Model.Interactions.Email.IInteractionEmail	EnterpriseEmailInteractionCurrent	Current e-mail interaction processed by Workspace Desktop Edition.
Genesyslab.Desktop.Modules.OpenMedia.Model. Interactions.IInteractionOpenMedia	EnterpriseOpenMediaInteractionCurrent	Current open media interaction processed by Workspace Desktop Edition.
Genesyslab.Desktop.Modules.OpenMedia. Model.Interactions.Sms.IInteractionSms	EnterpriseSmsInteractionCurrent	Current sms interaction in page mode processed by Workspace Desktop Edition.
	EnterpriseSmsSessionInteractionCurrent	Current sms interaction in session mode processed by Workspace Desktop Edition.

Enterprise Extensions

The `Genesyslab.Enterprise.Extensions` namespace defines a list of extensions classes which provide the switch-specific action areas of each related service.

Service	Extension	Related features
<code>IIdentityService</code>	<code>AgentServiceExtensions</code>	Manage login, Ready, Not Ready
<code>IDeviceService</code>	<code>DeviceServiceExtensions</code>	Manage the call-forward and Do Not Disturb features.
<code>IIMService</code>	<code>IMServiceExtensions</code>	Manage the messages and transcripts of instant messaging sessions.

Service	Extension	Related features
InteractionService	InteractionServiceExtensions	Manage the requests on interactions (Make the call, answer the call, transfer the call, and so on.)
IMonitorService	PAMExtensions	Manage subscriptions and statistic notifications.

Important

Add the `Genesyslab.Enterprise.Extensions` namespace to your code to enable the extension methods of your service.

Managing Connections and Channels

Workspace Desktop Edition manages the connections defined in the application configuration. You can access them through the `Genesyslab.Desktop.Modules.Core.SDK.Protocol.IChannelManager`. You can retrieve the connection by passing the configured application name at the registration of the channel, as shown below:

```
IChannelManager channelManager = container.Resolve<IChannelManager>();  
Genesyslab.Enterprise.Model.Channel.IClientChannel tserverChannel = channelManager.Register("YourApplicationName", "MyClientName");
```

Four application types are supported:

- TServer
- StatServer
- InteractionServer
- UCSServer

Through the `IChannelManager` interface, you can open channels for applications of these types without burdening Workspace Desktop Edition. However, if you wish to open new channels for other application types, you can use the `IChannelService` of the Enterprise API. Genesys

recommends that you name those channels according to their configuration's application name.

Connect your Channel

1. Retrieve the channel service

```
IChannelService channelService = EnterpriseService.Resolve<IChannelService>("channelService");
```

2. Create a new channel for each connection to open.

```
string channelName = "configName";  
TServerConfiguration configuration = new TServerConfiguration(channelName);  
configuration.ClientName = channelName;  
configuration.Uri = new Uri("tcp://hostname:port");  
configuration.WarmStandbyAttempts = 10;  
configuration.WarmStandbyTimeout = 5;  
configuration.WarmStandbyUri = new Uri("tcp://hostname:port");  
configuration.UseAddp = false;  
channelService.CreateChannel(channelName, configuration, SwitchModelType.LucentDefinityG3);
```

3. Register the channel's event handler before you open the connection, to ensure that your application does not miss any events. The following code snippet shows also how to retrieve the channel instance created.

```
Genesyslab.Enterprise.Model.Channel.IClientChannel channel = channelService.GetChannel(channelName);  
//Register for Channel events  
channelService.RegisterEvents(channel, new Action<Genesyslab.Enterprise.Model.Channel.IClientChannel>(ChannelEvent));
```

4. To make the connection to all of the channels, call the `IChannelService.Connect()` method..

```
channelService.Connect();
```

The code snippet uses the `channelName` string as a label to identify your connection. Your application will use this label later to access this channel.

Get the Protocol

The table in [Get EnterpriseService](#) provides the key for the protocols that associated with channels. You can retrieve the protocols once they are

connected, as shown in the following code snippet.

```
IEnterpriseProtocol media = voiceChannel.EnterpriseProtocols["voice"];
```

Getting Additional Service Events

In the Enterprise API, all services that allow event subscription include the following pair of self-describing methods: `RegisterEvents` and `UnRegisterEvents`. For instance, the following code snippet shows the registration of a `DeviceEvent` handler for the device service:

```
IDeviceService deviceService = EsdkService.Resolve<IDeviceService>("deviceService");  
IDevice device = deviceService.CreateDevice("myDevice", DeviceType.Extension);  
deviceService.RegisterEvents(device, new Action<IEnvelope<IDN>>(DeviceEvent));
```

To read the envelope content take advantage of the fact that the type of object published is specified in the handler declaration (which must match the registration requirements).

```
protected void DeviceEvent(IEnvelope<IDN> tsp)  
{  
    if (tsp != null)  
    {  
        //Retrieve the published object  
        IDevice device = (IDevice) tsp.Body;  
        System.Console.WriteLine("Name : " + device.Name + " Status: " + device.State.ToString());  
        switch (tsp.Header.CurrentContext.ContextState)  
        {  
            case ContextStateType.Error:  
                //...  
                break;  
                //...  
        }  
    }  
}
```

Threading Recommendations


When you write your handler code, you should process the event's `Envelope` in a separate thread that can take appropriate actions. Design your handlers to return as quickly as possible, because the library core works with all handlers sequentially-waiting for each handler to return, before working with the next handler. This recommendation is extremely important to ensure that:

- Your application remains synchronized with up-coming events.
- Your application remains synchronized with the real-time time line of external devices.

Attributes and Filters

You can define callback and filter attributes when declaring your event handlers.

- A callback attribute is used to hard-code the automatic registration of the handler method for a given channel.
- A filter attribute is used to hard-code the filtering of events that your application receives.

Attribute name	Type	Dependency
EnterpriseAgentEvent	Callback, Filter	Channel name
EnterpriseChannelEvent	Callback	Channel Name
EnterpriseDeviceEvent	Callback	Channel Name
EnterpriseFilter	Filter	Object parameters
EnterpriseInteractionEvent	Callback	Channel Name
EnterpriseMonitorEvent	Callback	Channel Name
EnterpriseStrategy	Filter	Strategy instance
EnterpriseService	Filter	 See below

Callback Attribute Syntax

If you use a callback attribute, callback registration is automatic. The following code snippet shows how to use method attributes by defining an interaction event handler for a SIP channel. The first part of the snippet shows the creation of the *TServerSIPChannel* channel. The second part shows the attribute's declaration.

```
//Channel Definition
IChannelService channelService = ESDKService.Resolve<IChannelService>("channelService");
TServerConfiguration myConfiguration = new TServerConfiguration("TServerSIPChannel");
```

```
channelService.CreateChannel("TServerSIPChannel", myConfiguration, mySwitchType);
//...
[EnterpriseInteractionEvent("TServerSIPChannel")]
protected void InteractionEvent(IEnvelope<IInteraction> tsp)
{
    //...
}
```

Filter Attribute Syntax

If you are using filter attributes, the callback registration is not automatic; therefore, you must implement it.

```
//Example of Filters:
//Callback active for the Agent 1001 when status is ready
[EnterpriseFilter("1001", "ready")]
protected void AgentEvent(IEnvelope<IAgent> tsp)
{
    //...
}
//Uses the AgentCallBackFilterStrategy strategy for calling this handler (or not)
[EnterpriseStrategy("genericFilter", typeof(AgentCallBackFilterStrategy))]
protected void AgentEvent(IEnvelope<IAgent> tsp)
{
    //...
}
//Callback active when ready status events.
[EnterpriseAgentEvent("ready")]
protected void AgentEvent(IEnvelope<IAgent> tsp)
{
    //...
}
```


Migrate Custom Applications from 8.1 to 8.5

Tip

If you want to highlight the functional areas in the Workspace UI with the red boxes and descriptions shown in the screen shots in this topic, on a *non-production* developer workstation, configure the value of the `keyboard.shortcut.action.test.hidden` developer option to a keyboard short cut, such as: `Ctrl+P`. When you press the keyboard shortcut, the functional areas are highlighted.

Migrate an existing 8.1.x Project

Before you start migrating your custom code, you must update Microsoft Visual Studio:

1. Download and install the latest update of Microsoft Visual Studio 2012.
2. Open the solution file of your 8.1 customization from Microsoft Visual Studio 2012.
3. If Visual Studio 2012 suggests that you migrate your project, you must follow the Microsoft Visual Studio 2012 migration wizard.

Issues related to .NET 4.5

Some XAML Tags

Some XAML tags used for isolated strings and localization can generate compilation errors. To avoid this issue, you must find all XAML tags with this kind of attribute format:

```
<MenuItem>
  ...
  Header="{loc:Translate Uid=Menu.MyBigMenu, Default=My Menu}"
  ...
</MenuItem>
```

And replace the formatting as follows:

```
<MenuItem>
  ...
  loc:Translate.Uid="Menu.MyBigMenu"
  Header="{loc:Translate Default=My Menu}"
  ...
</MenuItem>
```

Project References

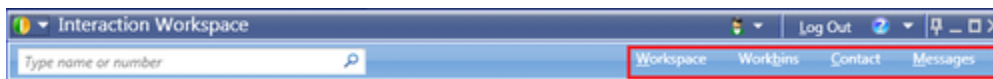
Microsoft Visual Studio 2012 (using .NET 4.5) is not as permissive with Project references as Microsoft Visual Studio 2008 (using NET 3.5). If your code includes using directives for a namespace implemented in a separate assembly, you must now add a reference to this assembly to the project's reference list.

Changes in the Main Window Regions

The 8.5.0 UI design reorganizes the regions that are contained in the Main Window.

Area ToolbarWorksheetButtonRegion

The ToolbarWorksheetButtonRegion area no longer includes Workspace, Workbins, Contact, and Messages buttons and the associated out-of-the-box functions.

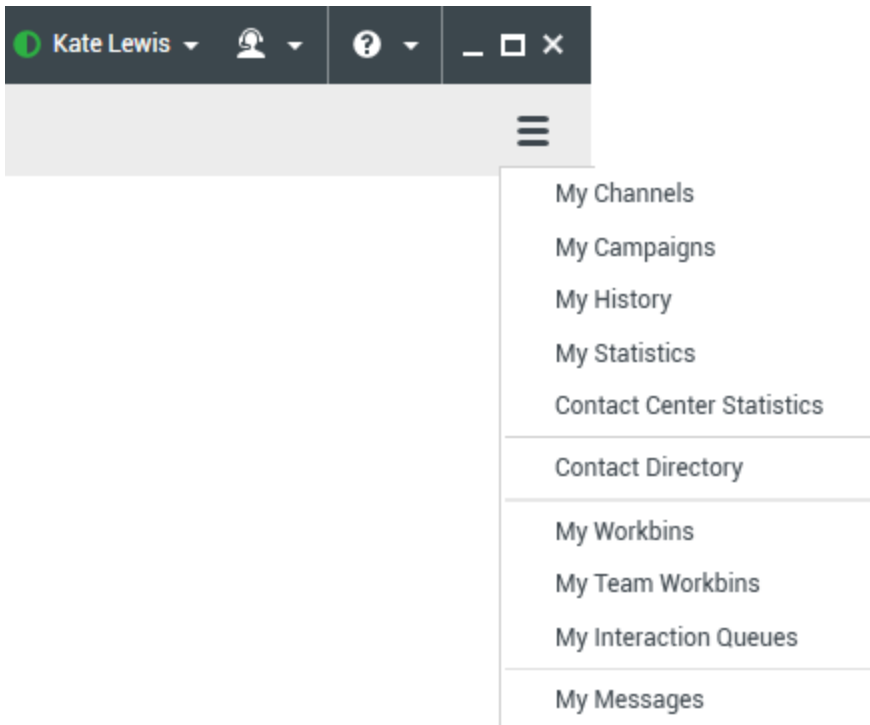


The 8.1 ToolbarWorksheetButtonRegion area.

Now, the ToolbarWorksheetButtonRegion area is empty by default and you can still include Custom Modules and Views in this region. Custom modules can no longer extend the drop-down menus of the former Workspace button through the WorkspaceMenuRegion region. Instead, you should use the ToolbarHamburgerButtonRegion area to add menu items in the hamburger menu along with the out of the box items of this area, such as My Channels, My Campaigns, and so on.



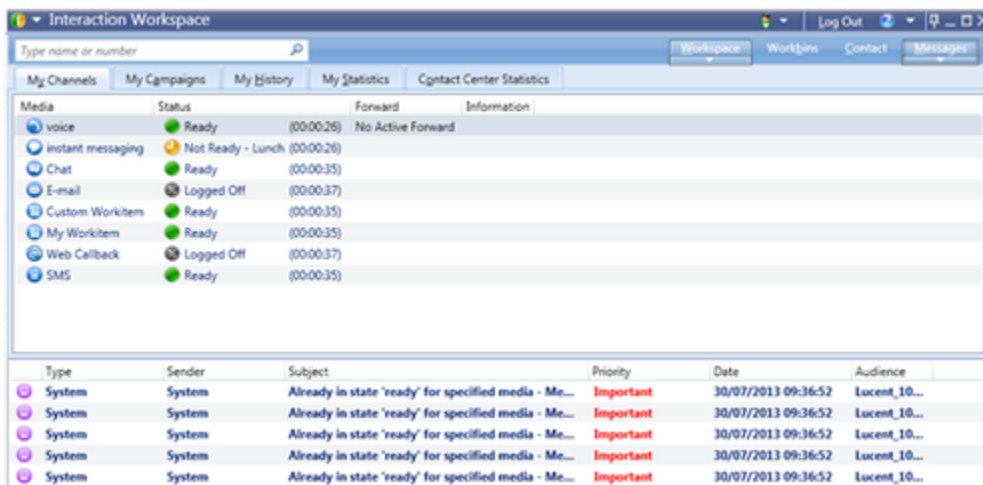
The 8.5 ToolbarWorksheetButtonRegion area.



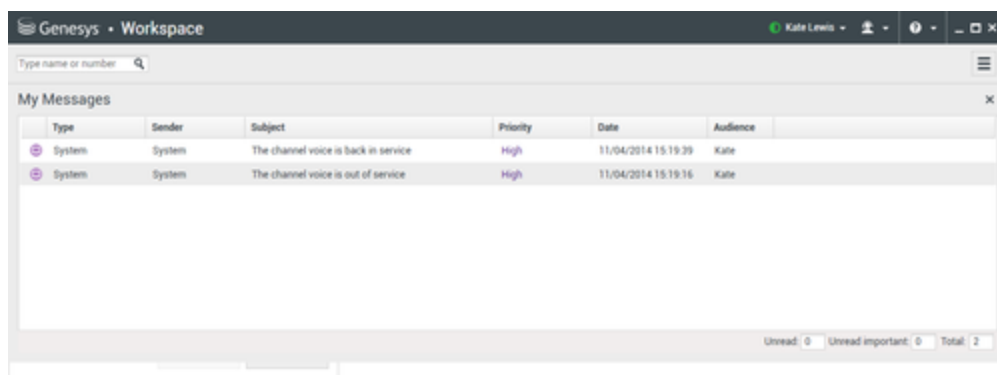
The new `ToolbarHamburgerButtonRegion` area.

Area `ToolbarWorksheetRegion`

The `ToolbarWorksheetRegion` region no longer includes the `'ToolbarStatusRegion'` and `'MyMessageRegion'` regions. The Message View was moved to the `ToolbarWorksheetRegion` region.



`ToolbarWorksheetRegion` in 8.1



ToolBarWorksheetRegion in 8.5.0

Area ToolBarInteractionBarRegion

The ToolBarInteractionBarRegion region is relocated between the ToolBarDialerRegion and ToolBarWorksheetButton regions.



ToolBarInteractionBarRegion in 8.1



ToolBarInteractionBarRegion in 8.5

Area CustomToolBarRegion

The CustomToolBarRegion region is a new SortableItemsControl region available on the left side of the Agent Status.



The new Customizable region in the toolbar.

The New Interaction Presentation Model

In 8.5.0, the Interaction Window View is replaced with pinned and floating interaction views.

Important

The Gadget mode is no longer compatible with the floating and pinned mode.

Pinned Interaction View

The CaseView root container of the Pinned Interaction View includes the same regions and views than the CasesRegion root container of the 8.1 Interaction Window View. If your customization requires the CasesRegion root container (not displayed in Pinned mode), Genesys recommends that you work in the legacy interaction window presentation mode.

Important

All other sub-regions and sub-views of the former Interaction Window View are still available in the pinned interaction mode.

Floating Interaction View

The Region structure of the Floating Interaction View is a subset of the Pinned Interaction View described above:

- Its root container is CaseView
- The CaseViewSideButtonRegion and InteractionWorksheetRegion regions and all the cascading material that they contain are not instantiated
- This mode displays the left part of the Interaction View
- The CaseViewSideButtonRegion region is replaced with the new CaseViewPopupSideButtonRegion region which enables you to insert side buttons in floating mode

If you implement side buttons in CaseViewPopupSideButtonRegion region, your custom code must trigger a programmatic switch to display the associated views when the buttons are clicked. You can do this by calling the following method:

```
IManagerInteractionContainer.ShowCaseViewInMainToolBar(<case_id>, <side_view_name>);
```

New Theme

Inheriting Theme's Style

You can use named styles in your XAML code to render the common controls (such as Buttons, Textbox, Textblock, Scrollbar, and so on) which are used in custom views in the same way as the out-of-the-box views.

See also [Configure your custom theme](#).

Images

```
<tabber> Image size=
```

Supported Image Sizes

In 8.5, the new theme uses Vector images, while the legacy theme used PNG. The image size changed for some components:

- Interaction control buttons in the main toolbar, consultation sections, and interaction windows, are now 24x24 pixels (versus 16x16 in 8.1).
- Most other buttons and icons are 16x16 pixels (same as in 8.1), except small icons and controls for pagination in contact and history views which are 12x12 pixels.

In 8.5, you can design your customization images (for icons, toolbar buttons, and so on.) as Vector or PNG resources. Vector images properly scale in 16x16 and have no size restrictions.

Important

If your custom code displays a 16x16 PNG image resource in one of the interaction control toolbars, Genesys recommends that you do not make it "stretch" to fit the real estate, but rather keep it 16x16. The `MagicImage` button control will automatically display empty space.

|> Images in XAML=

Implementing Images in XAML

You can use the new `commonControls:MagicImageUid` XAML control for `MagicImage` to integrate PNG (Source attribute) or Vector (ResourceKey attribute) images.

|> Vector Images=

Integrating Vector Images in Microsoft Visual Studio Project

If you must convert vector-based images to XAML, as for example, images created with Adobe Illustrator, Genesys recommends to use Microsoft Expression Design 4. A free version of this software is available for download here which is available for free from [Official Microsoft Download Center](#).

Important

The .AI file format is for Adobe Illustrator images.

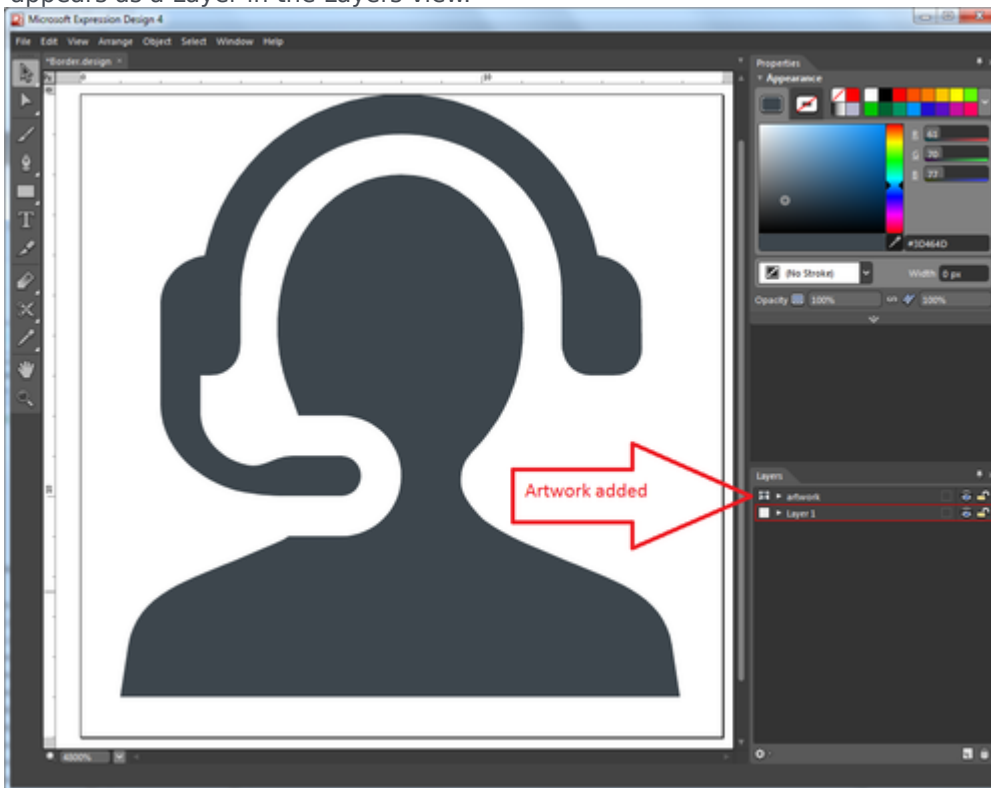
Converting an .AI Image to XAML DrawingBrush

Pre-requisite

- Download the [Border.design.zip](#) file.

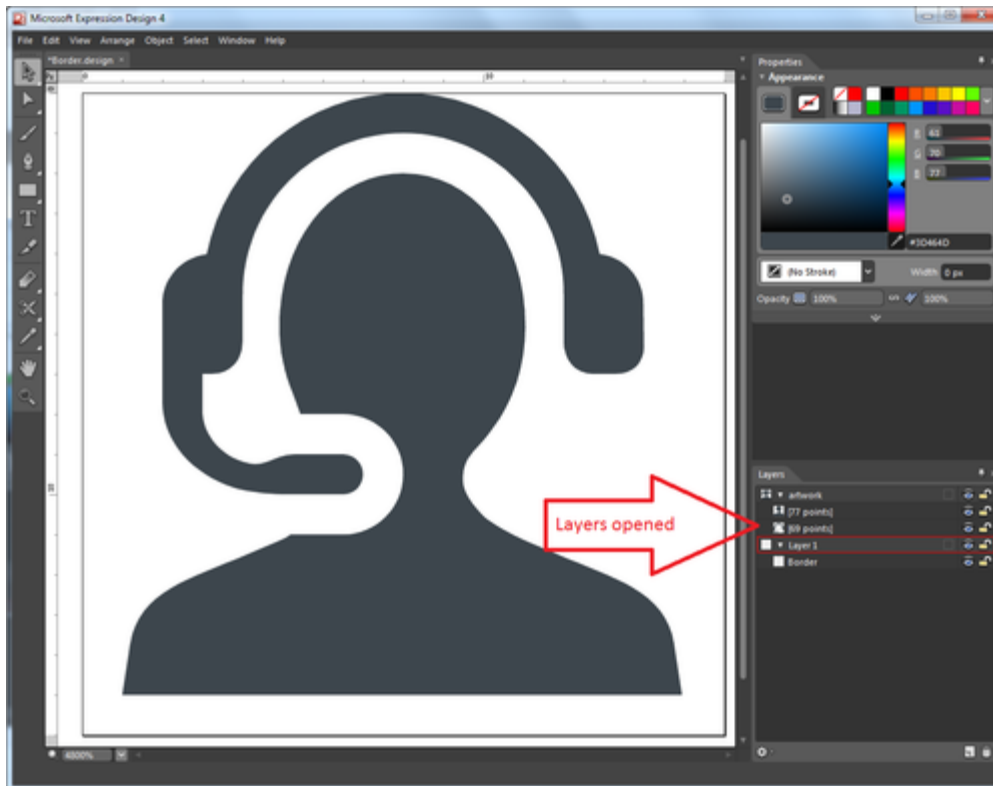
Start

1. Open the Border.design image in Microsoft Expression Design 4.
2. Drag and drop your Adobe Illustrator graphic image into Expression Design 4 artboard. Your image appears as a Layer in the Layers view.



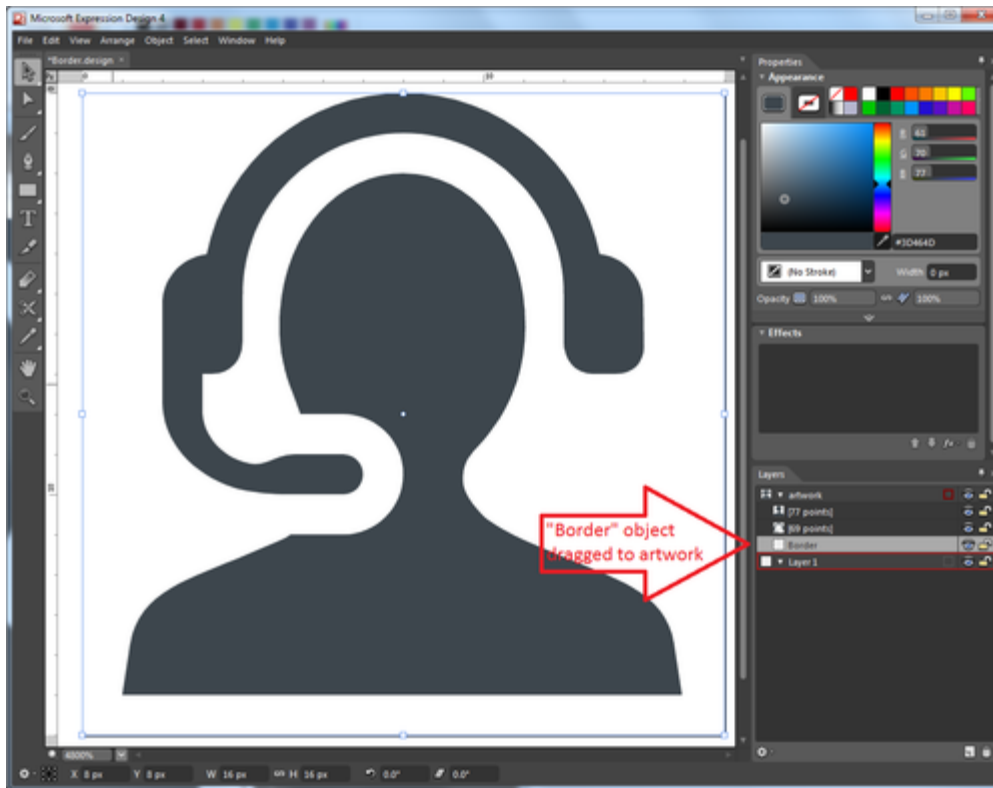
After a drag and drop, the artwork image is added to the Layers' list.

3. In the Layers view, expand each layer of the Layers' list to display the objects available for each of them.



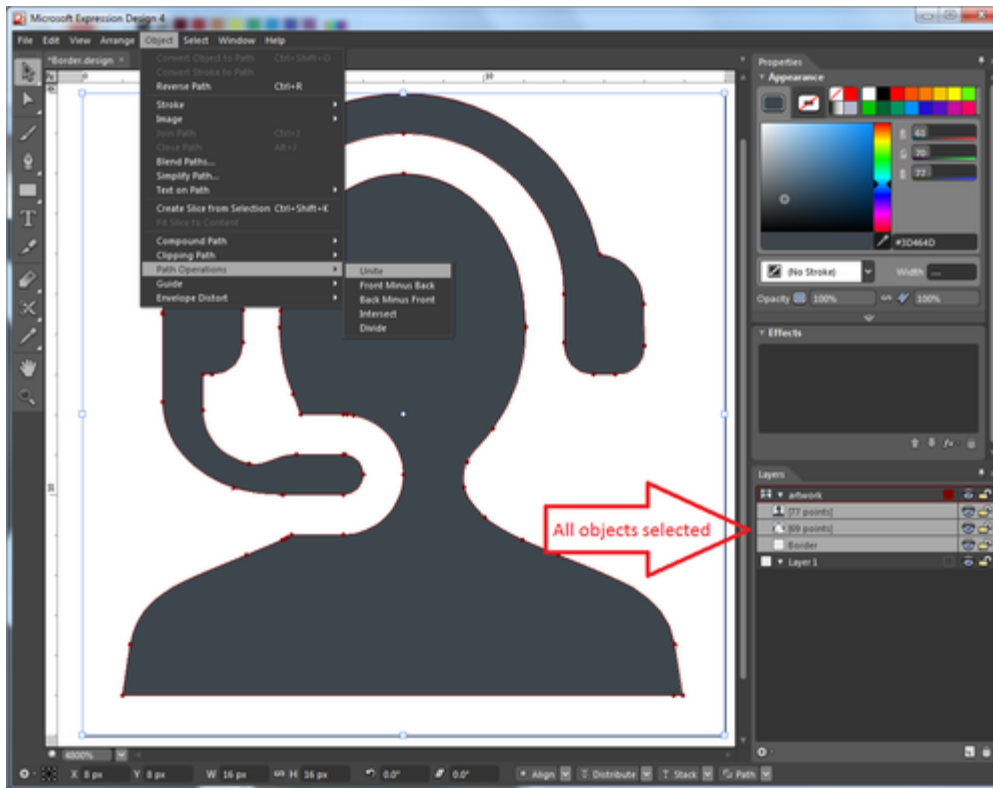
Your image may consist of one or several objects.

4. In the Layers view, select the Border object and drag and drop it into your image's layer.



The Border object was dragged to the Artwork's layer and appears in its Object list.

5. Navigate to the Object > Path Operations menu and apply the Unite Command. If the image is not altered (except in terms of colors), you can go to step 7 to export the image.



Applying the Unite command to the Artwork's objects.

Important

The Unite command turns multicolored images into single-colored images.

- If the Unite command altered your image, undo it and try the Make Compound Path command on the objects' selection. Navigate to Object > Compound Path > Make.

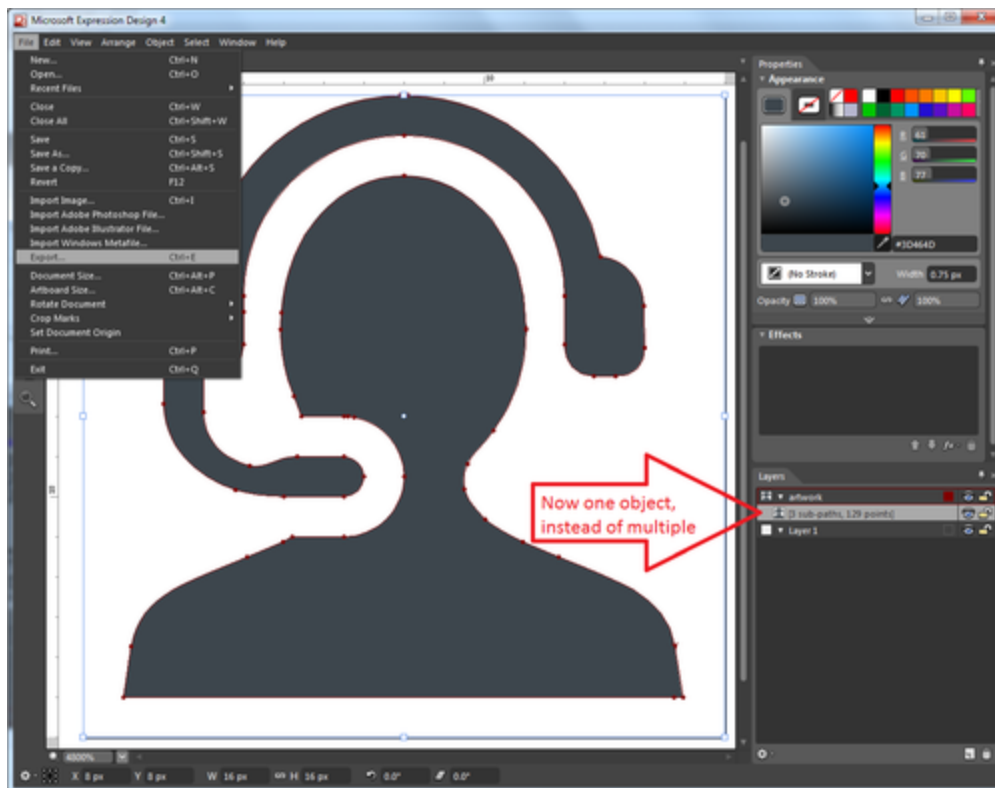


Apply the Make command.

Important

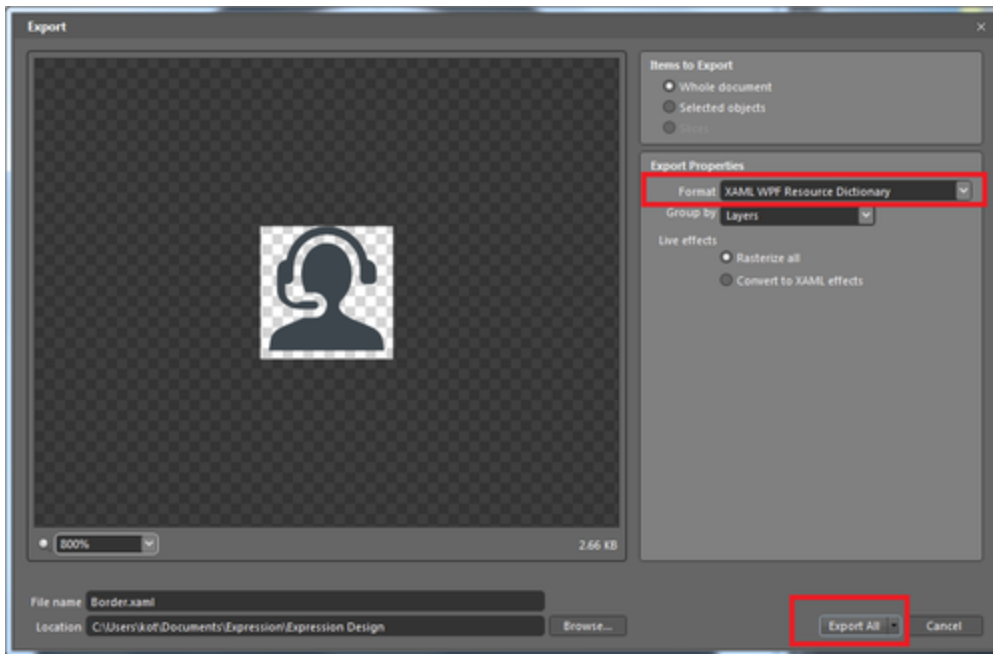
If the image is altered by the Make Compound Path command, the image cannot be converted to XAML Path. Ask the image's provider to recreate the image by using a single path.

7. If your image is not altered and includes only one object in its layer, proceed to the Export dialog.
 - Navigate to the File menu. Select Export...



Select File > Export.

- In the Export Properties section of the Export dialog box, set the format to XAML WPF Resource Dictionary.
- Enter a filename for your destination XAML DrawingBrush, browse a Location, and click Export All.



Set up your export properties and proceed to Export.

Converting .AI Image to XAML Path

1. Follow steps 1 to 6 of [the previous section](#).
2. Navigate to the File menu. Select Export...
3. In the Export Properties section of the Export dialog box, set the format to XAML Path.
4. Enter a filename for your destination XAML library, browse a Location, and click Export All.
5. Open the resulting XAML library and copy the path data available in the Geometry attribute of the GeometryDrawing item from the sample below into your code.

```
<?xml version="1.0" encoding="utf-8"?>
<ResourceDictionary xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
  <DrawingBrush x:Key="artwork" Stretch="Uniform">
    <DrawingBrush.Drawing>
      <DrawingGroup>
        <DrawingGroup.Children>
          <GeometryDrawing Brush="#FF3D464D" Geometry="F1 M 12.4147,15.7534L 15.488,18.66L
0.821333,18.66L 3.89467,15.7534L 3.896,15.7534C 5.19333,16.46 6.62667,16.9507
8.15333,16.9507L 8.15467,17.048L 8.156,16.9507C 9.68267,16.9507 11.116,16.46 12.3733,15.7747L
12.4147,15.7534 Z M 11.6107,6.58936L 11.3693,6.12006L 6.22933,11.2614C 6.792,11.6
7.44933,11.7987 8.15467,11.7987C 9.70933,11.7987 11.0413,10.8547 11.6107,9.50806C
11.8013,9.06006 11.9053 Z "/>
        </DrawingGroup.Children>
      </DrawingGroup>
    </DrawingBrush.Drawing>
  </DrawingBrush>
</ResourceDictionary>
```

Configure Your Custom Theme

As detailed in [Branding section of the Developer's Guide](#), you can easily add your own branding to Workspace Desktop Edition.

In 8.5, you can also create your own custom theme, with new colors and icons, as detailed in the following sections.

Add a New Theme

To add a new theme to the Workspace Desktop Edition, add your theme to the `<themes>` section of a configuration file.

- Best practice: Define a theme in a plugin configuration file; for instance, in the Custom Theme of the Workspace Desktop Edition sample, you can define your theme in the `Genesyslab.Desktop.Modules.CustomThemeSample.module-config` file.
or
- Add your theme to the `<themes>` section in the `InteractionWorkspace.exe.config` file .

For each new theme, the tag `<theme>` includes the list of dictionaries to load for the theme:

- The `<theme>` tag include three properties:
 - `name`: The name of theme, which must be unique.
 - `displayNameKey`: The entry key defined in the language dictionaries and used to display menu items.
 - `mainResourceDictionary`: Path of the XAML resources used in the dictionaries (styles and colors).

Important

The attribute values for `displayNameKey` and `mainResourceDictionary` can be overridden.

- The `<xmlDictionaries></xmlDictionaries>` tags include the list of XML dictionaries which contain all the paths for icons and images. Its `<xmlDictionary></xmlDictionary>` subtags each define an XML dictionary with the two following attributes:
 - `name`: The name of the dictionary.
 - `path`: The path to the dictionary file.

The sample themes are available in the Workspace Desktop Edition's installation Package. The following XML code shows the list of default themes

delivered with the Workspace Desktop Edition:

```
<themes>
  <!-- Default theme -->
  <theme name="Default" displayNameKey="Theme.Default.DisplayName" xamlDictionary="/Genesyslab.Desktop.WPFCommon;component/themes/generic.xaml">
    <xmlDictionaries>
      <xmlDictionary name="iw" path=".\\Resources\\ResourcesDefinition85.xml"></xmlDictionary>
    </xmlDictionaries>
  </theme>
  <!-- Additional themes -->
  <theme name="Royale" displayNameKey="Theme.Royale.DisplayName" mainResourceDictionary="/Genesyslab.Desktop.WPFCommon;component/themes/royal.xaml">
    <xmlDictionaries>
      <xmlDictionary name="iw" path=".\\Resources\\ResourcesDefinition.xml"></xmlDictionary>
    </xmlDictionaries>
  </theme>
  <theme name="Fancy" displayNameKey="Theme.Fancy.DisplayName" mainResourceDictionary="/Genesyslab.Desktop.WPFCommon;component/themes/fancy.xaml">
    <xmlDictionaries>
      <xmlDictionary name="iw" path=".\\Resources\\ResourcesDefinition.xml"></xmlDictionary>
    </xmlDictionaries>
  </theme>
</themes>
```

You can extend and override the values or XML dictionaries, as shown below in the Custom Theme Sample.

```
<themes>
  <!-- Extend and Override the Default theme -->
  <theme name="Default" displayNameKey="Theme.MyDefault.DisplayName"><!-- override displayNameKey -->
    <xmlDictionaries>
      <xmlDictionary name="rebranding" path=".\\Resources\\RebrandingTheme.xml"></xmlDictionary> <!-- Extend the XML Dictionaries-->
    </xmlDictionaries>
  </theme>

  <!-- Add a new Custom Theme -->
  <theme name="CustomTheme" displayNameKey="Theme.Custom.DisplayName"
mainResourceDictionary="/Genesyslab.Desktop.Modules.CustomThemeSample;component/Resources/themes/CustomTheme.xaml">
    <xmlDictionaries>
      <xmlDictionary name="iw" path=".\\Resources\\ResourcesDefinitionCustom.xml"></xmlDictionary>
    </xmlDictionaries>
  </theme>
```



```
</themes>
```

The CustomTheme.Xaml file must declare the main resource dictionary of the new Workspace Edition style in addition to the Custom Color dictionary.

```
<ResourceDictionary xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
                    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">

    <ResourceDictionary.MergedDictionaries>
        <!-- New IW Style -->
        <ResourceDictionary Source="/Genesyslab.Desktop.WPFCommon;component/Resources/NewStyles/NewStylesResourceLibrary.xaml"/>
        <ResourceDictionary Source="/Genesyslab.Desktop.Modules.CustomThemeSample;component/Resources/ColorBrushes/
CustomDefaultColorTheme.xaml"/>
    </ResourceDictionary.MergedDictionaries>
</ResourceDictionary>
```

You can also set up the theme at the Workspace Desktop Edition startup (SplashScreen, First Login Panel, and so on) by adding the `gui.theme` key to the `appSettings` section of the `InteractionWorkspace.exe.config` configuration file.

```
<appSettings>
    <add key="gui.theme" value="Default"/><!-- Name of theme-->
</appSettings>
```

Customize the Interaction Bar Framework

The Workspace Desktop Edition implements the same Interaction Toolbar for the Interaction Bar and Bundle Bar areas with different colors; the Interaction Bar's components are dark and the Bundle Bar's components are light.

You can customize the Interaction Toolbar without creating a new one. You can set up several buttons associated with the following components by setting the `ButtonStyle` dependency property with the following values:

- `InteractionToolBarButton`
- `InteractionToolBarSplitButton`
- `InteractionToolBarToggleButton`
- `InteractionToolBarDropDownButton`

The following XAML sample shows how to customize the style of the `ButtonSingleStepTransfer` component.

```
<commonControls:InteractionToolBarButton localization:Translate.Uid="Windows.VoiceView.ButtonSingleStepTransfer" ButtonStyle="{Binding
ButtonStyle}"
    Name="buttonSingleStepTransfer"
    Click="SingleStepTransferCommand_Click"
    ToolTip="{Binding Path=ToolTipButtonSingleStepTransfer, ElementName=callView}"
    AutomationProperties.Name="{Binding Path=PropertiesNameButtonSingleStepTransfer, ElementName=callView}">
  <commonControls:InteractionToolBarButton.Visibility>
    <MultiBinding Converter="{StaticResource visibilityBooleanORConverter}" Mode="OneWay" FallbackValue="Collapsed">
      <Binding Path="Interaction.IsItPossibleToOneStepTransfer" FallbackValue="Visible" Mode="OneWay"
Converter="{StaticResource visibilityConverter}"/>
      <Binding Path="Interaction.IsItPossibleToCompleteTransferActiveConsultation" FallbackValue="Visible" Mode="OneWay"
Converter="{StaticResource visibilityConverter}" />
    </MultiBinding>
  </commonControls:InteractionToolBarButton.Visibility>
  <StackPanel Orientation="Horizontal">
    <commonControls:MagicImage localization:Translate.Uid="Common.Images.Interaction.Voice.Transfer"
Source="{localization:Translate}"
    ResourceKey="{localization:Translate}" Width="24" Height="24" RenderOptions.BitmapScalingMode="NearestNeighbor"
/>
    <Path VerticalAlignment="Center" Margin="3" Data="M0,0L3,3 6,0z" Width="{Binding Source=8, Converter={StaticResource
relativeSizeConverter}}"
    Height="{Binding Source=4, Converter={StaticResource relativeSizeConverter}}"
    Fill="{Binding Foreground, ElementName=buttonSingleStepTransfer}">
  </Path>
</StackPanel>
</commonControls:InteractionToolBarButton>
```

The view model must also implement the `IButtonStyle` interface.

```
ButtonStyle buttonStyle;
public ButtonStyle ButtonStyle
{
    get { return buttonStyle; }
    set
    {
        if (buttonStyle != value)
        {
            buttonStyle = value;
        }
    }
}
```

```
        OnPropertyChanged("ButtonStyle");
    }
}
```

A helper provides the view context's information about the toolbar style to the `ModelView`, as shown below.

```
public void Create()
{
    IDictionary<string, object> contextDictionary = Context as IDictionary<string, object>;
    HelperToolbarFramework.SetButtonStyle(contextDictionary, Model);
}
```


Frequently Asked Questions

Purpose: Frequently asked questions (FAQ) about Interaction Workspace's customization. If your question is neither answered here nor in the documentation, then please ask for help in the [Genesys forums](#).

Is it possible to hide or select custom views?

You can do this by using a condition when adding your view with the `IViewManager`, as described in [Hiding and Showing Custom Views](#).

How can I use a URI passed in attached data?

The following code snippet adds a WPF `WebBrowser` control to the view. The `Case` is extracted from the context dictionary of the view and the URL is retrieved from the attached data of the main `Interaction`:

```
public partial class MyCustomView : UserControl, IMyCustomView
{
    // ...
    public void Create()
    {
        IDictionary<string, object> contextDictionary = (Context as IDictionary<string, object>);
        object caseObject;
        if(contextDictionary.TryGetValue("Case", out caseObject))
        {
            ICase theCase = caseObject as ICase;
            // Get the URL from the interaction attached data
            string urlField = theCase.MainInteraction.GetAttachedData("URL_field") as string;
            // Get URI to navigate to
            Uri uri = new Uri(urlField, UriKind.RelativeOrAbsolute);
            // Create the web browser control and add it to the view (here an UserControl)
            System.Windows.Controls.WebBrowser myWebBrowser = new System.Windows.Controls.WebBrowser();
            this.Content = myWebBrowser;
            myWebBrowser.Navigate(uri);
        }
    }
    // ...
}
```

How do I access to the objects container (IUnityContainer)?

Genesys does not recommend that the global objects containers are used this way, but if your are stuck with no other possibility, you can call the `ContainerAccessPoint.Container.Resolve<T>()` method. For instance, the following code snippet retrieves the global container to get the `IAgent` implementation:

```
// To get the global IAgent implementation from anywhere:  
IAgent agent = ContainerAccessPoint.Container.Resolve<IAgent>();
```

Is it possible to add some permanent text in the case information panel?

If you want to add permanent information here, you can configure a `casedata` in the configuration with the `casedata` business attribute and inject an attached data key/value pair in the corresponding interaction. See [interaction.case-data.format-business-attribute](#). The following code shows you how to handle the interaction events and injects the attached data "Segment" with a value "Hello" into it. "Segment" would be the name of your `casedata` business attribute element.

```
// The start of your extension module  
public void Initialize()  
{  
    // ...  
    container.Resolve<IViewEventManager>().Subscribe(MyEventHandler);  
}  
  
void MyEventHandler(object eventObject)  
{  
    string eventMessage = eventObject as string;  
    if (eventMessage != null)  
    {  
        switch (eventMessage)  
        {  
            case "Login":  
                container.Resolve<IInteractionManager>().InteractionEvent +=  
                    new System.EventHandler<EventArgs<IInteraction>> (ExtensionSampleModule_InteractionEvent);  
                break;  
            case "Logout":  
                container.Resolve<IInteractionManager>().InteractionEvent -=
```

```
        new System.EventHandler<EventArgs<IInteraction>> (ExtensionSampleModule_InteractionEvent);
        viewEventManager.Unsubscribe(MyEventHandler);
        break;
    }
}

void ExtensionSampleModule_InteractionEvent(object sender, EventArgs<IInteraction> e)
{
    //Add a reference to: Genesyslab.Enterprise.Services.Multimedia.dll
    //and Genesyslab.Enterprise.Model.dll object flag;
    IInteraction interaction = e.Value;
    if (!interaction.UserData.TryGetValue("myAttachedDataFlag", out flag))
    {
        Genesyslab.Enterprise.Model.Interaction.IOpenMediaInteraction openMediaInteraction =
            interaction.EnrepriseInteractionCurrent as Genesyslab.Enterprise.Model.Interaction.IOpenMediaInteraction;
        bool add = false;
        if (openMediaInteraction != null) // If an openmedia interaction
            add = openMediaInteraction.IsInWorkflow;
        else
            add = !interaction.IsIdle; // If a voice interaction
        if (add)
        {
            interaction.SetAttachedData("Segment", "Coucou");
            interaction.UserData["myAttachedDataFlag"] = true;
        }
    }
}
```

Is it possible to modify the workitem panel?

This is the exact purpose of the "Genesyslab.Desktop.Modules.CustomWorkItemSample" sample. More details are available in the following pages:

- [About the Extension Samples](#)
- [Customizing Views and Regions](#)
- [Customize Views and Regions](#)

How can I log an exception in Workspace Desktop Edition's logging system?

You may need to add a reference to the assembly: `Microsoft.Practices.Unity.dll`. You can send messages through the `ILogger` that is used by Workspace Desktop Edition to log errors and alerts as shown below:

```
try
{
    // Simulate an exception
    throw new Exception("BIG Exception");
}
catch (Exception exception)
{
    // Create the text message
    string myMessage = string.Format("My message: {0}", exception.Message);
    // Logging the message
    ContainerAccessPoint.Container.Resolve<ILogger>().CreateChildLogger("MyCustomSample").Error(myMessage, exception);
    // Sending the error to the alerting system
    new ExceptionAnalyzer(ContainerAccessPoint.Container).PublishError(AlertSection.Public, myMessage, null, null);
}
```

How can I send an exception through Workspace Desktop Edition's alert system?

You need to add references to the assemblies:

- `Microsoft.Practices.Composite.dll`
- `Microsoft.Practices.Unity.dll`

Then, you can create an alert as follows:

```
// To send any text message
ContainerAccessPoint.Container.Resolve<IEventAggregator>().GetEvent<AlertEvent>().Publish(new Alert()
{
    Section = "Public",
    Severity = SeverityType.Message,
    Id = "My message"
```

```
});
```

Where Section can be:

- "Public" to display the message in the main message panel with a toaster preview.
- "Login" to display the message in the login panel.
- "Forward" to display the message in the forward message box.
- A Caseld, to display the message at the top of a case view.

How can I translate a text message from the dictionary and publish it as an alert?

1. First, declare your text message in the dictionary file. For instance:

```
<Value Id="Windows.ErrorLoginView.NoConnectConfigurationServer" Text="Could not connect to Configuration Server host '{0}' on port '{1}'."/>
```

2. Create an alert which uses the text message:

```
Alert my Alert = new Alert()  
{  
    Section = "Public",  
    Severity = SeverityType.Error,  
    Id = "Windows.ErrorLoginView.NoConnectConfigurationServer",  
    Target = "Text",  
    Parameters = new object[] { "configuration.server.fr", 2020 }  
};
```

Where Section can be:

- "Public" to display the message in the main message panel with a toaster preview.
 - "Login" to display the message in the login panel.
 - "Forward" to display the message in the forward message box.
-

- A CaseId, to display the message at the top of a case view.

3. Resolve the IEventAggregator interface through the unity container and publish your alert:

```
ContainerAccessPoint.Container.Resolve<IEventAggregator>().GetEvent<AlertEvent>().Publish(myAlert);
```

How can I subscribe to/unsubscribe from Workspace Desktop Edition alerts ?

1. You need to implement an AlertEventHandler as follows:

```
void AlertEventHandler(Alert alert)
{ // Do what you have to, for instance:
  Console.WriteLine(alert.Message);
}
```

2. Subscribe or unsubscribe using the following code snippets:

```
// subscribe
ContainerAccessPoint.Container.Resolve<IEventAggregator>().GetEvent<AlertEvent>().Subscribe(AlertEventHandler, ThreadOption.UIThread,
true);
// unsubscribe
ContainerAccessPoint.Container.Resolve<IEventAggregator>().GetEvent<AlertEvent>().Unsubscribe(AlertEventHandler);
```


Use Customizable Commands



Purpose: To provide information about the customizable commands available in the Workspace Desktop Edition.

Before You Start

- The command system is based on the chain of command (or chain of responsibility) design pattern.
- All the code snippets in this page are extracted from the `Genesyslab.Desktop.Modules.ExtensionSample` source files.
- In addition to this page, read:
 - [Creating a New Module](#)
 - [Deploying Your Custom Module into the Genesys Out-Of-The-Box Application](#)

Inserting a Command in a Chain

Each element of command is unique across the given chain. You can use the `ICoMmandManager.InsertCommandToChainOfCommandAfter()` method to insert your command after a specific command by passing its name. The following code snippet shows how to insert the element of command "CloseSample" in the chain of command "BundleClose" after the element of command "IsPossibleToClose":

[C#]

```
commandManager.InsertCommandToChainOfCommandAfter("BundleClose", "IsPossibleToClose", new List<CommandActivator>()
{
    new CommandActivator()
    {
        CommandType = typeof(CloseSampleCommand), Name = "CloseSample"
    }
});
```

Creating a Command

Creating a new command is considered an advanced topic. Genesys recommends that you do so with caution. Consider doing this in consultation with the development community:

- [Genesys Engage DevFoundry](#)
- [Genesys Community](#)

The following example illustrates how to create your own commands by using Genesys best practices. For each new command, create a class which implements the `IElementOfCommand` interface. After creating the command, you must add it to a chain of command in your module (see [Creating a New Module](#)). The custom command created in the following step-by-step example displays a confirmation dialog before executing the `ReleaseCall` command.

1. Create the elementary command class: **[C#]**

```
// File: CustomCommand.cs
namespace Genesyslab.Desktop.Modules.ExtensionSample.CustomCommand
{
    // Custom command which prompts a confirmation dialog before executing the ReleaseCall command
    class BeforeReleaseCallCommand : IElementOfCommand
    {
        readonly IObjectContainer container;
        ILogger log;
        public BeforeReleaseCallCommand(IObjectContainer container)
        {
            this.container = container;
            // Initialize the trace system
            this.log = container.Resolve<ILogger>();
            // Create a child trace section
            this.log = log.CreateChildLogger("BeforeReleaseCallCommand");
        }
        public string Name { get; set; }
        public bool Execute(IDictionary<string, object> parameters, IProgressUpdater progress)
        {
            // To go to the main thread
            if (Application.Current.Dispatcher != null && !Application.Current.Dispatcher.CheckAccess())
            {
                object result = Application.Current.Dispatcher.Invoke(DispatcherPriority.Send,
```

```

        new ExecuteDelegate(Execute), parameters, progress);
    return (bool)result;
}
else
{
    log.Info("Execute");
    // Get the parameter
    IInteractionVoice interactionVoice = parameters["CommandParameter"] as IInteractionVoice;
    // Prompt the alert dialog
    return MessageBox.Show("Do you really want to release this call?\r\nThe call",
        "Release the call?", MessageBoxButton.YesNo) == MessageBoxResult.No;
}
}
}
delegate bool ExecuteDelegate(IDictionary<string, object> parameters, IProgressUpdater progressUpdater);
}
}

```

2. Create the chain of command in the Module initialization by using the CommandManager:

[C#]

```

// File: ExtensionSampleModule.cs
ICommandManager commandManager = container.Resolve<ICommandManager>();
// Add a command before the release call
// Method 1:
commandManager.CommandsByName["InteractionVoiceReleaseCall"].Insert(0, new CommandActivator() {
    CommandType = typeof(BeforeReleaseCallCommand), Name = "BeforeReleaseCall" });
// Method 2 (recommended):
commandManager.InsertCommandToChainOfCommandBefore("InteractionVoiceReleaseCall", "ReleaseCall",
    new CommandActivator() { CommandType = typeof(BeforeReleaseCallCommand), Name = "BeforeReleaseCall" });

```

3. You can add several commands to a chain of command. The order of execution follows the order in which the commands are added. BeforeReleaseCallCommand is executed before ReleaseCallCommand, for example: **[C#]**

```

commandManager.AddCommandToChainOfCommand("InteractionVoiceReleaseCall",
    new List<CommandActivator>()
    {
        new CommandActivator() { CommandType = typeof(BeforeReleaseCallCommand), Name = "BeforeReleaseCall" },
        new CommandActivator() { CommandType = typeof(ReleaseCallCommand), Name = "ReleaseCall" }
    });

```

4. Finally, execute the chain of command by using parameters, as shown in the following example (defined here: Command list): **[C#]**


```
IDictionary<string, object> parameters = new Dictionary<string, object>();
parameters.Add("CommandParameter", interaction);
parameters.Add("Reasons", reasons);
parameters.Add("Extensions", extensions);
commandManager.GetChainOfCommandByName("InteractionVoiceReleaseCall").Execute(parameters);
```

Multiple Commands and Overlapping

When you pass several commands to a given chain, they share the parameters which have identical names. This can lead to over-lapping issues when you execute the command. To by-pass this issue, make sure that your parameters are correct before your application executes the command. For instance, consider using the Command1 and Command2 of MyChain:

Chain of Command	Default commands	Parameters
MyChain	Command1	<ul style="list-style-type: none"> Parameter1: IInteractionChat Parameter2: KeyValueCollection
	Command2	<ul style="list-style-type: none"> Parameter1: IInteractionChat Parameter3: KeyValueCollection

- IInteractionChat: Genesyslab.Desktop.Modules.OpenMedia.Model.Interactions.Chat.IInteractionChat
- KeyValueCollection: Genesyslab.Enterprise.Commons.Collections.KeyValueCollection

When you execute MyChain, you must pass all the parameters of Command1 and Command2. Parameter1 is shared amongst Command1 and Command2.

[C#]

```
IDictionary<string, object> parameters = new Dictionary<string, object>();
parameters.Add("Parameter1", interaction);
parameters.Add("Parameter2", reasons);
```

```
parameters.Add("Parameter3", extensions);  
commandManager.GetChainOfCommandByName("MyChain").Execute(parameters);
```

Subscribing to Events

When you are creating custom commands there is no “command” that you can intercept; however, there is an “event” that you can subscribe to that will notify you when the interaction bar tab selection is changed. The following sample demonstrates how this is done:
[CustoInteractionBarSelectHandler.zip](#)

Reference for Commands



Purpose: To provide the list of customizable commands for Workspace Desktop Edition.

The following table provides the list of commands available for the Interaction workspace. Parameters are detailed in the related pages.

Important

Read [Use Customizable Commands](#) to see code snippets which demonstrate the usage of the commands.

Chains of Commands

Available chains of command	Related pages
<ul style="list-style-type: none"> ActiveRecordingPauseRecordIxnScreenCommand ActiveRecordingPauseRecordScreenCallCommand ActiveRecordingResumeRecordIxnScreenCommand ActiveRecordingResumeRecordScreenCallCommand ActiveRecordingStartRecordIxnScreenCommand ActiveRecordingStartRecordScreenCallCommand ActiveRecordingStopRecordIxnScreenCommand ActiveRecordingStopRecordScreenCallCommand 	<p>Active Recording</p>
<ul style="list-style-type: none"> AskSaveModifiedContact 	<p>UCS</p>
<ul style="list-style-type: none"> ApplicationClose BroadcastMessageClose BroadcastMessageShow 	<p>Windows</p>
<ul style="list-style-type: none"> BundleClose 	<ul style="list-style-type: none"> Chat Callback IM E-Mails

Available chains of command	Related pages
	<ul style="list-style-type: none"> • Voice • Open Media • SMS • Work Items
<ul style="list-style-type: none"> • CampaignGetPreviewRecord 	Outbound
<ul style="list-style-type: none"> • CancelConnectionCommand 	Windows
<ul style="list-style-type: none"> • CancelOutboundChain • ClosePullPreviewRecord 	Outbound
<ul style="list-style-type: none"> • ContactIsModified • ContactNotification • ContactOpenSearchWindow • ContactOpenToDialog • CreateInteractionUCSFromESDKInteraction • DeleteInteractionUCS 	UCS
<ul style="list-style-type: none"> • InteractionCallbackAccept • InteractionCallbackActiveCustomerSession • InteractionCallbackCompleteCustomerSession • InteractionCallbackDecline • InteractionCallbackIsPossibleToClose • InteractionCallbackParkParent • InteractionCallbackPullParent • InteractionCallbackWorkflow 	Callback
<ul style="list-style-type: none"> • InteractionChatAcceptChat • InteractionChatAutoCloseInteraction • InteractionChatAutoDisconnect • InteractionChatCloseInteraction 	Chat

Available chains of command	Related pages
<ul style="list-style-type: none"> • InteractionChatCompleteConferenceAgent • InteractionChatCompleteTransferAgent • InteractionChatConferenceAgent • InteractionChatConsultPromptForEnd • InteractionChatConsultRelease • InteractionChatDeclineChat • InteractionChatDisconnectChat • InteractionChatDisconnectChatSessionChannel • InteractionChatDisconnectMonitoredChat • InteractionChatDisconnectMonitoredCoachChat • InteractionChatInitConsultation • InteractionChatInitiateConferenceOtherType • InteractionChatInviteAddInvitingParty • InteractionChatInviteRemoveInvitingParty • InteractionChatIsPossibleToClose • InteractionChatJoinChat • InteractionChatLeaveInteractionFromConference • InteractionChatPlaceInQueue • InteractionChatPushUrl • InteractionChatReconnect • InteractionChatReleasePartyFromConference • InteractionChatReleasePartyFromConsult • InteractionChatReopenChannel • InteractionChatSendMessage • InteractionChatTransferAgent • InteractionChatTypingStarted • InteractionChatTypingStopped • InteractionChatWorkflow 	
<ul style="list-style-type: none"> • InteractionEmailActionOnOpenedEmailById • InteractionEmailClose • InteractionEmailDelete 	<p>E-Mails</p>

Available chains of command	Related pages
<ul style="list-style-type: none"> • InteractionEmailDeleteById • InteractionEmailForward • InteractionEmailsPossibleToClose • InteractionEmailLoad • InteractionEmailMarkDoneById • InteractionEmailMoveToWorkbin • InteractionEmailPlaceInQueue • InteractionEmailPrint • InteractionEmailPullById • InteractionEmailPullFromWorkbinById • InteractionEmailReply • InteractionEmailReplyById • InteractionEmailResend • InteractionEmailSave • InteractionEmailSaveNewOutbound • InteractionEmailSend • InteractionEmailSendInterimReply • InteractionEmailTransferAgent • InteractionEmailWorkflow • InteractionEmailWorkflowFromInteractionESDK 	
<ul style="list-style-type: none"> • InteractionIMClose • InteractionIMIsPossibleToClose • InteractionIMReleaseIM • InteractionIMSendMessage 	<p>IM</p>
<ul style="list-style-type: none"> • InteractionInboundEmailPossibleToMoveToWorkbin • InteractionInboundEmailSearchAndMoveToWorkbin 	<p>E-Mails</p>
<ul style="list-style-type: none"> • InteractionInternalConferenceInviteAccept • InteractionInternalConferenceInviteConferenceAgent • InteractionInternalConferenceInviteDecline 	<p>Open Media</p>

Available chains of command	Related pages
<ul style="list-style-type: none"> InteractionInternalConferenceInvitePull InteractionInternalConferenceInviteSendInvite InteractionInternalConferenceInviteStopProcessing 	
<ul style="list-style-type: none"> InteractionFilterSelected InteractionManagementMarkDone InteractionManagementMoveToQueueById InteractionManagementMoveToWorkbinById InteractionManagementSetAttachedData 	<p>Workbins</p>
<ul style="list-style-type: none"> InteractionOpenMediaCanPullInteraction InteractionOpenMediaClose InteractionOpenMediaCloseOnOutOfService InteractionOpenMediaConferenceAgent InteractionOpenMediaGetInteractionStateById InteractionOpenMediaHandlingTimeout InteractionOpenMediaMoveToQueue InteractionOpenMediaMoveToWorkbin InteractionOpenMediaPlaceInQueue InteractionOpenMediaPullById InteractionOpenMediaPullFromWorkbinById InteractionOpenMediaPullFromWorkbinByld InteractionOpenMediaTransferAgent InteractionOpenMediaUCSStopProcessing InteractionOpenMediaUCSUpdate InteractionOpenMediaWorkflow 	<p>Open Media</p>
<ul style="list-style-type: none"> InteractionOutboundEmailPossibleToMoveToWorkbin InteractionOutboundEmailSearchAndMoveToWorkbin 	<p>E-Mails</p>
<ul style="list-style-type: none"> InteractionPossibleToMoveToWorkbin 	<p>Workbins</p>

Available chains of command	Related pages
<ul style="list-style-type: none"> • InteractionPushPreviewAccept • InteractionPushPreviewDecline • InteractionPushPreviewWorkflow 	<p style="text-align: center;">Outbound</p>
<ul style="list-style-type: none"> • InteractionSetAttachedDataWithDelay 	<ul style="list-style-type: none"> • Callback • Chat • IM • E-Mails • Voice • SMS • Open Media • Work Items
<ul style="list-style-type: none"> • InteractionSmsAccept • InteractionSmsCreateReplyMessage • InteractionSmsDecline • InteractionSmsDelete • InteractionSmsDisconnectSmsSessionChannel • InteractionSmsGetHistoricMessage • InteractionSmsIsPossibleToClose • InteractionSmsPlaceInQueue • InteractionSmsSendMessage • InteractionSmsSendNewMessage • InteractionSmsSendReplyMessage • InteractionSmsTransferAgent • InteractionSmsWorkflow 	<p style="text-align: center;">SMS</p>
<ul style="list-style-type: none"> • InteractionUCSInQueue 	<p style="text-align: center;">UCS</p>
<ul style="list-style-type: none"> • InteractionVoiceAlternateCall • InteractionVoiceAnswerCall • InteractionVoiceAutoCloseConsult 	<p style="text-align: center;">Voice</p>

Available chains of command	Related pages
<ul style="list-style-type: none"> • InteractionVoiceBeforeClose • InteractionVoiceClose • InteractionVoiceCompleteConference • InteractionVoiceCompleteTransfer • InteractionVoiceDeleteFromConference • InteractionVoiceHoldCall • InteractionVoiceIfPossibleCloseInteraction • InteractionVoiceInitConference • InteractionVoiceInitTransfer • InteractionVoiceIsPossibleToClose • InteractionVoiceListenDisconnect • InteractionVoiceListenReconnect • InteractionVoiceMuteOff • InteractionVoiceMuteOn • InteractionVoicePauseRecordCall • InteractionVoiceReconnectCall • InteractionVoiceRecordCall • InteractionVoiceRedirectCall • InteractionVoiceReleaseCall • InteractionVoiceReportBadQualityOfMedia • InteractionVoiceResume • InteractionVoiceResumeRecordCall • InteractionVoiceRetrieveCall • InteractionVoiceSendDTMF • InteractionVoiceSingleStepConference • InteractionVoiceSingleStepTransfer • InteractionVoiceStopRecordCall 	
<ul style="list-style-type: none"> • InteractionWorkItemAccept • InteractionWorkItemDecline • InteractionWorkitemMarkDoneById • InteractionWorkItemMoveToWorkbin 	<p>Work Items</p>

Available chains of command	Related pages
<ul style="list-style-type: none"> • InteractionWorkItemPossibleToMoveToWorkbin • InteractionWorkitemPutBackToOriginalSource • InteractionWorkItemSearchAndMoveToWorkbin • InteractionWorkitemWorkflowFromInteractionESDK 	
<ul style="list-style-type: none"> • MarkProcessedOutboundChain 	Outbound
<ul style="list-style-type: none"> • MultipleInteractionsActionFromWorkbinById Added in 8.1.401 	Workbins
<ul style="list-style-type: none"> • MediaEmailCreateNewOutboundEmail 	Place and Media
<ul style="list-style-type: none"> • MediaOpenMediaChangeStateReason • MediaOpenMediaDNDOff • MediaOpenMediaDNDOn • MediaOpenMediaLogOff • MediaOpenMediaLogOffPlace • MediaOpenMediaLogOn • MediaOpenMediaLogOnPlace • MediaOpenMediaNotReady • MediaOpenMediaReady • MediaSmsCreateNewOutboundPageMode • MediaVoiceDNDOff • MediaVoiceDNDOn • MediaVoiceLogOff • MediaVoiceLogOn • MediaVoiceMakeCall • MediaVoiceMakeIM • MediaVoiceNotReady • MediaVoiceReady • MediaVoiceStartIMSession 	Place and Media
<ul style="list-style-type: none"> • NotepadInsertTimeStamp 	Windows

Available chains of command	Related pages
<ul style="list-style-type: none"> OpenMediaEsdkInteractionPutBackInOriginalLocation 	Open Media
<ul style="list-style-type: none"> PossibleToCloseAllInteractions PossibleToClosePullPreview PossibleToClosePushPreview 	Windows
<ul style="list-style-type: none"> RejectOutboundChain 	Outbound
<ul style="list-style-type: none"> SetAttachedDataAndPutBack 	Workbins
<ul style="list-style-type: none"> SetLastCalledAgentUCS 	UCS
<ul style="list-style-type: none"> StartPreviewRecordOutbound 	Outbound
<ul style="list-style-type: none"> TeamCommunicatorAddToFavorite TeamCommunicatorModifyFavorite TeamCommunicatorOpenInteraction 	Windows
<ul style="list-style-type: none"> TeamLeadEndMonitoring TeamLeadEndMonitoringChat TeamLeadEndMonitoringVoice TeamLeadMonitorAll TeamLeadSetSupervisorBargeInChat TeamLeadSetSupervisorBargeInVoice TeamLeadSetSupervisorCoachChat TeamLeadSetSupervisorCoachVoice TeamLeadSetSupervisorMuteVoice TeamLeadStopMonitor 	Supervision
<ul style="list-style-type: none"> ToasterAcceptPreview ToasterDeclinePreview ToasterInteractionAccept 	Windows

Available chains of command	Related pages
<ul style="list-style-type: none"> • ToasterInteractionDecline • ToasterInteractionEmailAccept • ToasterInteractionEmailDecline • ToasterInteractionIMAccept • ToasterInteractionIMReject • ToasterInteractionOpenMediaAccept • ToasterInteractionOpenMediaDecline • ToasterInteractionPreviewAccept • ToasterInteractionPreviewDecline • ToasterShowBroadcastMessage 	
<ul style="list-style-type: none"> • UpdateOutboundRecord 	<p style="color: #C00000;">Outbound</p>
<ul style="list-style-type: none"> • UpFrontPauseRecordingVoiceCallCommand • UpFrontPauseRecordingVoiceCallCommandImplicit • UpFrontResumeRecordingVoiceCallCommand • UpFrontResumeRecordingVoiceCallCommandImplicit <p>Added in 8.5</p>	<p style="color: #C00000;">Voice</p>
<ul style="list-style-type: none"> • WindowClose • WindowInteractionClose 	<p style="color: #C00000;">Windows</p>
<ul style="list-style-type: none"> • WorkbinPreSelectedByName • WorkbinPullInteraction • WorkbinPushInteraction • WorkbinSelected 	<p style="color: #C00000;">Workbins</p>

Active Recording



Purpose: To present for each active recording command, the list of default subcommands available and the associated parameters.

Learn about the Active Recording Commands

These commands enable to perform active recording operations on interactions. Read [Use Customizable Commands](#) to see code snippets which demonstrate the usage of the commands.

Managing Active Recording

The commands presented in this section are specific to active recording of interaction and cover the following features (this list is not exhaustive):

- Start recording an interaction;
- Pause recording an interaction;
- Resume recording an interaction;
- Stop recording an interaction.

Chains of Commands to Manage Active Recording of Interactions

Default Commands	Parameters
Chain ActiveRecordingPauseRecordIxnScreenCommand	
<ul style="list-style-type: none"> Pause 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia - Genesyslab.Desktop.Modules.Core.Model.Interactions.IInteractionOpenMedia
Chain ActiveRecordingPauseRecordScreenCallCommand	
<ul style="list-style-type: none"> Pause 	<ul style="list-style-type: none"> CommandParameter: IInteractionVoice - Genesyslab.Desktop.Modules.Core.Model.Interactions.IInteractionVoice
Chain ActiveRecordingResumeRecordIxnScreenCommand	
<ul style="list-style-type: none"> Resume 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia
Chain ActiveRecordingResumeRecordScreenCallCommand	
<ul style="list-style-type: none"> Resume 	<ul style="list-style-type: none"> CommandParameter: IInteractionVoice
Chain ActiveRecordingStartRecordIxnScreenCommand	
<ul style="list-style-type: none"> Start 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia
Chain ActiveRecordingStartRecordScreenCallCommand	
<ul style="list-style-type: none"> Start 	<ul style="list-style-type: none"> CommandParameter: IInteractionVoice
Chain ActiveRecordingStopRecordIxnScreenCommand	
<ul style="list-style-type: none"> Stop 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia

Default Commands	Parameters
Chain ActiveRecordingStopRecordScreenCallCommand	
<ul style="list-style-type: none">• Stop	<ul style="list-style-type: none">• CommandParameter: IInteractionVoice

Callback



Purpose: To provide information about callback commands.

Learn about Callback Commands

The commands in this page are restricted to callback operations. However, callback interactions are extended open media interactions and can be used as such in open media commands.

You can also use specific **Open Media** which apply to open media interactions in some GUI components.

Important

Read [Use Customizable Commands](#) to see code snippets which demonstrate the usage of the commands.

Managing Callback Interactions

Open media commands cover the following features (this list is not exhaustive):

- Accept and decline;
- Place in queue;
- Transfer;
- Move to workbins.

Chains of Commands to Manage Callback Interactions

Default commands	Parameters
Chain InteractionCallbackAccept	
AcceptCallback	<ul style="list-style-type: none"> • CommandParameter: IInteractionOpenMedia - Genesyslab.Desktop.Modules.OpenMedia.Model.Interactions.II... • Extensions: KeyValueCollection - Genesyslab.Enterprise.Commons.Collections.KeyValueCollectio
Chain InteractionCallbackActiveCustomerSession	

Default commands	Parameters
ActiveCustomerSessionCallback	CommandParameter: IInteractionCallback - Genesyslab.Desktop.Modules.Callback.Model.Interactions.IInteract
Chain InteractionCallbackCompleteCustomerSession	
CompleteCustomerSessionCallback	CommandParameter: IInteractionCallback
Chain InteractionCallbackDecline	
DeclineCallback	<ul style="list-style-type: none"> • CommandParameter: IInteractionOpenMedia • Reasons: KeyValueCollection • Extensions: KeyValueCollection
Close	CommandParameter: IInteractionOpenMedia
Chain InteractionCallbackIsPossibleToClose	
IsPossibleToClose	CommandParameter: IInteractionCallback
Chain InteractionCallbackParkParent	
ParkCallback	CommandParameter: IInteraction - Genesyslab.Desktop.Modules.Core.Model.Interactions.IInteraction
Chain InteractionCallbackPullParent	
PullCallback	<ul style="list-style-type: none"> • CommandParameter: IInteraction • IW_WebcallbackInteractionId: String
Chain InteractionCallbackWorkflow	
RescheduleCallback	CommandParameter: IInteractionCallback
UpdateInteractionUCS	CommandParameter: IInteraction
<ul style="list-style-type: none"> • MarkProcessedCallback 	CommandParameter: IInteractionCallback
Commands common to all interactions	
Chain InteractionSetAttachedDataWithDelay	
SetAttachedData	<ul style="list-style-type: none"> • CommandParameter: IInteraction • AttachedData: KeyValueCollection
Chain BundleClose	
GetAttachedDataInformationUCS	CommandParameter: IInteractionsBundle -

Default commands	Parameters
	Genesyslab.Desktop.Modules.Core.Model.Interactions.IInteractionsBundle
UpdateNotePadForVoice	<ul style="list-style-type: none"> • CommandParameter: IInteractionsBundle
ResetInteractionChatConsultation	CommandParameter: IInteractionsBundle
IsContactModified	<ul style="list-style-type: none"> • CaseId: String
IsPossibleToClose	CommandParameter: IInteractionsBundle
CompleteDispositionCodeOnBundle	CommandParameter: IInteractionsBundle
Close	CommandParameter: IInteractionsBundle
StopInteractionVoiceUCS	<ul style="list-style-type: none"> • CommandParameter: IInteractionsBundle
GetOutboundPreviewRecord	CommandParameter: ICampaign - Genesyslab.Enterprise.Model.Interaction.ICampaign

Chat

Purpose: To present for each chat command, the list of default subcommands available and the associated parameters.

Learn about the Chat Commands

The chat commands presented in this page are restricted to chat interactions. However, chat interactions are extended open media interactions and can be used as such in open media commands. In addition to the chat commands, you can use:

- [Open Media](#) to place interactions in a given queue, move them to workbins, and so on.
- [Manage Toaster](#) which also deals with chat interactions.
- [Manage Open Media](#) to manage the chat media.

Important

Read [Use Customizable Commands](#) to see code snippets which demonstrate the usage of the commands.

Managing Chat Interactions

The commands presented in this section are specific to chat interactions and cover the following features (this list is not exhaustive):

- Accept or decline a chat session;
- Make chat consultations or transfer the session;
- Release a party;
- Close or leave the session.

Chains of Commands to Manage Chat Interactions

Default Commands	Parameters
Chain InteractionChatAcceptChat	
<ul style="list-style-type: none"> Accept 	<ul style="list-style-type: none"> CommandParameter: IInteractionChat - Genesyslab.Desktop.Modules.OpenMedia.Model.Interactions.Chat.IInteractionChat
Chain InteractionChatAutoCloseInteraction	
<ul style="list-style-type: none"> IsContactModified 	<ul style="list-style-type: none"> CasId: String- System.String
<ul style="list-style-type: none"> ResetInteractionChatConsultationCommand 	<ul style="list-style-type: none"> CommandParameter: IInteractionsBundle - Genesyslab.Desktop.Modules.Core.Model.Interactions.IInteractionsBundle
<ul style="list-style-type: none"> Close 	<ul style="list-style-type: none"> CommandParameter: IInteractionChat
<ul style="list-style-type: none"> CloseConsultation 	<ul style="list-style-type: none"> CommandParameter: IInteractionChat
Chain InteractionChatAutoDisconnect	
<ul style="list-style-type: none"> IsContactModified 	<ul style="list-style-type: none"> CasId: String- System.String
<ul style="list-style-type: none"> ReleaseMyParty 	<ul style="list-style-type: none"> CommandParameter: IInteractionChat AfterActionMode: String- System.String
Chain InteractionChatCloseInteraction	
<ul style="list-style-type: none"> IsContactModified 	<ul style="list-style-type: none"> CasId: String- System.String

Default Commands	Parameters
<ul style="list-style-type: none"> IsPossibleToClose 	<ul style="list-style-type: none"> CommandParameter: IInteractionChat
<ul style="list-style-type: none"> Close 	<ul style="list-style-type: none"> CommandParameter: IInteractionChat
Chain InteractionChatCompleteConferenceAgent	
<ul style="list-style-type: none"> CompleteConference 	<ul style="list-style-type: none"> CommandParameter: IInteractionChatCommon - Genesyslab.Desktop.Modules.OpenMedia.Model.Interactions.Chat.IInteractionChatCommon Destination: String Reason: KeyValueCollection - Genesyslab.Enterprise.Commons.Collections.KeyValueCollection Extensions: KeyValueCollection
Chain InteractionChatCompleteTransferAgent	
<ul style="list-style-type: none"> RequestSaveNotepad 	<ul style="list-style-type: none"> CommandParameter: IInteraction - Genesyslab.Desktop.Modules.Core.Model.Interactions.IInteraction NotepadViewModel: INotepadViewModel - Genesyslab.Desktop.Modules.Windows.Interactions.INotepadViewModel
<ul style="list-style-type: none"> CompleteTransfer 	<ul style="list-style-type: none"> CommandParameter: IInteractionChatCommon Destination: String Reason: KeyValueCollection Extensions: KeyValueCollection
Chain InteractionChatConferenceAgent	

Default Commands	Parameters
<ul style="list-style-type: none"> ConferenceAgent 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia - Genesyslab.Desktop.Modules.OpenMedia.Model.Interactions.IInteractionOpenMedia Visibility: String Destination: String Extensions: KeyValueCollection
<ul style="list-style-type: none"> AddRecentCall 	<ul style="list-style-type: none"> RecentIndex: ObjectTC - Genesyslab.Desktop.Modules.TeamCommunicator.ObjectTC
Chain InteractionChatConsultPromptForEnd	
<ul style="list-style-type: none"> PromptEnd 	<ul style="list-style-type: none"> CommandParameter: IInteraction ConfirmParentWindow: Window - System.Window.Window
Chain InteractionChatConsultRelease	
<ul style="list-style-type: none"> SetSupervisorMuteChat 	
<ul style="list-style-type: none"> Release 	<ul style="list-style-type: none"> CommandParameter: IInteractionChatConsult - Genesyslab.Desktop.Modules.OpenMedia.Model.Interactions.Chat.IInteractionChatConsult Reason: KeyValueCollection Extensions: KeyValueCollection
Chain InteractionChatDeclineChat	
<ul style="list-style-type: none"> Decline 	<ul style="list-style-type: none"> CommandParameter: IInteractionChat

Default Commands	Parameters
	<ul style="list-style-type: none"> • Reasons: KeyValueCollection • Extensions: KeyValueCollection
<ul style="list-style-type: none"> • Close 	<ul style="list-style-type: none"> • CommandParameter: IInteractionOpenMedia
Chain InteractionChatDisconnectChat	
<ul style="list-style-type: none"> • PromptEnd 	<ul style="list-style-type: none"> • CommandParameter: IInteraction • ConfirmParentWindow: Window
<ul style="list-style-type: none"> • SetSupervisorMuteChat 	
<ul style="list-style-type: none"> • Disconnect 	<ul style="list-style-type: none"> • CommandParameter: IInteractionChat • Reason: KeyValueCollection • Extensions: KeyValueCollection
Chain InteractionChatDisconnectChatSessionChannel	
<ul style="list-style-type: none"> • IsContactModified 	<ul style="list-style-type: none"> • CaselId: String- System.String
<ul style="list-style-type: none"> • Disconnect 	<ul style="list-style-type: none"> • CommandParameter: IClientChannel - Genesyslab.Enterprise.Model.Channel.IClientChannel
Chain InteractionChatDisconnectMonitoredChat	

Default Commands	Parameters
<ul style="list-style-type: none"> IsContactModified 	<ul style="list-style-type: none"> CasId: String- System.String
<ul style="list-style-type: none"> Disconnect 	<ul style="list-style-type: none"> CommandParameter: IInteractionChat Reason: KeyValueCollection Extensions: KeyValueCollection
Chain InteractionChatDisconnectMonitoredCoachChat	
<ul style="list-style-type: none"> IsContactModified 	<ul style="list-style-type: none"> CasId: String - System.String
<ul style="list-style-type: none"> DisconnectCoach 	<ul style="list-style-type: none"> CommandParameter: IInteractionChat Reason: KeyValueCollection Extensions: KeyValueCollection
Chain InteractionChatInitConsultation	
<ul style="list-style-type: none"> SetTransferInformation 	<ul style="list-style-type: none"> CommandParameter: IInteraction NotepadViewModel: INotepadViewModel
<ul style="list-style-type: none"> RequestSaveNotepad 	<ul style="list-style-type: none"> CommandParameter: IInteraction NotepadViewModel: INotepadViewModel
<ul style="list-style-type: none"> InitConsultation 	<ul style="list-style-type: none"> CommandParameter: IInteractionChatCommon

Default Commands	Parameters
	<ul style="list-style-type: none"> • Destination: String • Reasons: KeyValueCollection • Extensions: KeyValueCollection
<ul style="list-style-type: none"> • AddRecentCall 	<ul style="list-style-type: none"> • RecentIndex: ObjectTC
Chain InteractionChatInitiateConferenceOtherType	
<ul style="list-style-type: none"> • SetCollaborationQueueContext 	<ul style="list-style-type: none"> • CommandParameter: IInteractionChat • System.String: Queue, output parameter - Queue, output parameter
<ul style="list-style-type: none"> • SetCollaborationContext 	<ul style="list-style-type: none"> • CommandParameter: IInteractionChat • Media: IMediaOpenMedia output parameter - Genesyslab.Desktop.Modules.OpenMedia.Model.Agents.IMediaOpenMedia
<ul style="list-style-type: none"> • CreateCollaboration 	<ul style="list-style-type: none"> • CommandParameter: IInteractionChat • System.String: Queue - Queue • UserData: KeyValueCollection • Media: IMediaOpenMedia - Genesyslab.Desktop.Modules.OpenMedia.Model.Agents.IMediaOpenMedia • CorrelatorData: String
<ul style="list-style-type: none"> • AddRecentCall 	<ul style="list-style-type: none"> • RecentIndex: ObjectTC

Default Commands	Parameters
Chain InteractionChatInviteAddInvitingParty	
<ul style="list-style-type: none"> AddParty 	<ul style="list-style-type: none"> CommandParameter: String Destination: String DisplayObject: IFormattedObject Type - Genesyslab.Desktop.Infrastructure.ObjectFormat.IFormattedObject Type
Chain InteractionChatInviteRemoveInvitingParty	
<ul style="list-style-type: none"> RemoveParty 	<ul style="list-style-type: none"> CommandParameter: String Destination: String
Chain InteractionChatIsPossibleToClose	
<ul style="list-style-type: none"> IsContactModified 	<ul style="list-style-type: none"> CasId: String- System.String
<ul style="list-style-type: none"> IsPossibleToClose 	<ul style="list-style-type: none"> CommandParameter: IInteractionChat
Chain InteractionChatJoinChat	
<ul style="list-style-type: none"> Join 	<ul style="list-style-type: none"> CommandParameter: IInteractionChat
Chain InteractionChatLeaveInteractionFromConference	
<ul style="list-style-type: none"> IsContactModified 	<ul style="list-style-type: none"> CasId: String - System.String
<ul style="list-style-type: none"> Leave 	<ul style="list-style-type: none"> CommandParameter: IInteractionChat

Default Commands	Parameters
	<ul style="list-style-type: none"> Reason: KeyValueCollection Extensions: KeyValueCollection
Chain InteractionChatPlaceInQueue	
<ul style="list-style-type: none"> IsContactModified 	<ul style="list-style-type: none"> Caseld: String - System.String
<ul style="list-style-type: none"> UpdateInteractionUCS 	<ul style="list-style-type: none"> CommandParameter: IInteraction NotepadViewModel: INotepadViewModel
<ul style="list-style-type: none"> SaveAttachedData 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia UserData: KeyValueCollection
<ul style="list-style-type: none"> PlaceInQueue 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia Destination: String Reasons: KeyValueCollection Extensions: KeyValueCollection
<ul style="list-style-type: none"> SetAutoClose 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia
<ul style="list-style-type: none"> ReleaseMyParty 	<ul style="list-style-type: none"> CommandParameter: IInteractionChat AfterActionMode: String

Default Commands	Parameters
<ul style="list-style-type: none"> AddRecentCall 	<ul style="list-style-type: none"> RecentIndex: ObjectTC
Chain InteractionChatPushUrl	
<ul style="list-style-type: none"> PushUrl 	<ul style="list-style-type: none"> CommandParameter: IInteractionChat Url: String Visibility: String
Chain InteractionChatReconnect	
<ul style="list-style-type: none"> Join 	<ul style="list-style-type: none"> CommandParameter: IInteractionChat
Chain InteractionChatReleasePartyFromConference	
<ul style="list-style-type: none"> ReleaseParty 	<ul style="list-style-type: none"> CommandParameter: IChatParty - Genesyslab.Desktop.Modules.OpenMedia.Model.Interactions.Chat.IChatParty
Chain InteractionChatReleasePartyFromConsult	
<ul style="list-style-type: none"> ReleaseParty 	<ul style="list-style-type: none"> CommandParameter: IChatParty
Chain InteractionChatReopenChannel	
<ul style="list-style-type: none"> Reopen 	<ul style="list-style-type: none"> CommandParameter: IInteractionChat
Chain InteractionChatSendMessage	
<ul style="list-style-type: none"> SendMessage 	<ul style="list-style-type: none"> CommandParameter: IInteractionChatCommon

Default Commands	Parameters
	<ul style="list-style-type: none"> • CaselId: String Added in: 8.5 • Visibility: String • Message: String
Chain InteractionChatTransferAgent	
<ul style="list-style-type: none"> • RequestSaveNotepad 	<ul style="list-style-type: none"> • CommandParameter: IInteraction • NotepadViewModel: INotepadViewModel
<ul style="list-style-type: none"> • TransferAgent 	<ul style="list-style-type: none"> • CommandParameter: IInteractionOpenMedia • Destination: String • Reasons: KeyValueCollection • Extensions: KeyValueCollection
<ul style="list-style-type: none"> • SetAutoClose 	<ul style="list-style-type: none"> • CommandParameter: IInteractionOpenMedia
<ul style="list-style-type: none"> • ReleaseMyParty 	<ul style="list-style-type: none"> • CommandParameter: IInteractionChat • AfterActionMode: String
<ul style="list-style-type: none"> • AddRecentCall 	<ul style="list-style-type: none"> • RecentIndex: ObjectTC
Chain InteractionChatTypingStarted	

Default Commands	Parameters
<ul style="list-style-type: none"> TypingStarted 	<ul style="list-style-type: none"> CommandParameter: IInteractionChatCommon Visibility: String
Chain InteractionChatTypingStopped	
<ul style="list-style-type: none"> TypingStopped 	<ul style="list-style-type: none"> CommandParameter: IInteractionChatCommon Visibility: String
Chain InteractionChatWorkflow	
<ul style="list-style-type: none"> CloseConsult 	<ul style="list-style-type: none"> CommandParameter: IInteractionChat
<ul style="list-style-type: none"> ExitJoinIssue Available since: 8.5 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia Reason: KeyValueCollection Extensions: KeyValueCollection
<ul style="list-style-type: none"> Workflow 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia Workflow: String Reasons: KeyValueCollection Extensions: KeyValueCollection
Commands common to all interactions	
Chain InteractionSetAttachedDataWithDelay	

Default Commands	Parameters
SetAttachedData	<ul style="list-style-type: none"> • CommandParameter: IInteraction • AttachedData: KeyValueCollection
Chain BundleClose	
GetAttachedDataInformationUCS	CommandParameter: IInteractionsBundle - Genesyslab.Desktop.Modules.Core.Model.Interactions.IInteractionsBundle
UpdateNotePadForVoice	<ul style="list-style-type: none"> • CommandParameter: IInteractionsBundle
ResetInteractionChatConsultation	CommandParameter: IInteractionsBundle
IsContactModified	<ul style="list-style-type: none"> • CaselId: String
IsPossibleToClose	CommandParameter: IInteractionsBundle
CompleteDispositionCodeOnBundle	CommandParameter: IInteractionsBundle
Close	CommandParameter: IInteractionsBundle
StopInteractionVoiceUCS	<ul style="list-style-type: none"> • CommandParameter: IInteractionsBundle
GetOutboundPreviewRecord	CommandParameter: ICampaign - Genesyslab.Enterprise.Model.Interaction.ICampaign

E-Mails

Purpose: To provide information about customizable commands for the e-mail interactions.

Learn about E-Mail Commands

The e-mail commands presented in this page are restricted to e-mail interactions. However, e-mail interactions are extended open media interactions and can be used as such in open media commands. In addition to the e-mail commands, you can use:

- [Open Media](#) to place interactions in a given queue, move them to workbins, and so on.
- [Commands for the E-Mail Media](#) to manage the creation of outbound e-mails.
- [Manage Toaster](#) which also deals with e-mail interactions.

Important

Read [Use Customizable Commands](#) to see code snippets which demonstrate the usage of the commands.

Managing E-Mail Interactions

The commands presented in this section are specific to e-mail interactions and cover the following features (this list is not exhaustive):

- Accept or decline a given e-mail;
- Load, send, reply, and save a given e-mail;
- Pull and place in queue;
- Close or stop the interactions.

Chains of Commands to Manage E-Mail Interactions

Default Commands	Parameters
Chain InteractionEmailActionFromWorkbinDeleteById	
<ul style="list-style-type: none"> GetInteractionState 	<ul style="list-style-type: none"> CommandParameter: String
<ul style="list-style-type: none"> CanPullInteraction 	<ul style="list-style-type: none"> CommandParameter: Dictionary
<ul style="list-style-type: none"> DisplayWarningPullNotPossible 	<ul style="list-style-type: none"> canPullInteractionresult: String
<ul style="list-style-type: none"> Confirm 	<ul style="list-style-type: none"> ActionToConfirm: String ConfirmParentWindow: Window
<ul style="list-style-type: none"> Delete 	<ul style="list-style-type: none"> CommandParameter: String WorkbinId: String
Chain InteractionEmailActionFromWorkbinMarkDoneById	
<ul style="list-style-type: none"> GetInteractionState 	<ul style="list-style-type: none"> CommandParameter: String
<ul style="list-style-type: none"> CanPullInteraction 	<ul style="list-style-type: none"> CommandParameter: Dictionary
<ul style="list-style-type: none"> DisplayWarningPullNotPossible 	<ul style="list-style-type: none"> canPullInteractionresult: String
<ul style="list-style-type: none"> NoChildrenOpened 	<ul style="list-style-type: none"> CommandParameter: String

Default Commands	Parameters
	<ul style="list-style-type: none"> WarningMessageTarget: String DoNotCheckForParentId: String
<ul style="list-style-type: none"> IsDispositionMissing Available since: 8.5 	<ul style="list-style-type: none"> EsdkOpenMediaInteraction: IIOpenMediaInteraction
<ul style="list-style-type: none"> Confirm 	<ul style="list-style-type: none"> ActionToConfirm: String ConfirmParentWindow: Window
<ul style="list-style-type: none"> MarkDone 	<ul style="list-style-type: none"> CommandParameter: String WorkbinId: String
Chain InteractionEmailActionFromWorkbinPullById	
<ul style="list-style-type: none"> GetInteractionState 	<ul style="list-style-type: none"> CommandParameter: String
<ul style="list-style-type: none"> CanPullInteraction 	<ul style="list-style-type: none"> CommandParameter: Dictionary
<ul style="list-style-type: none"> DisplayWarningPullNotPossible 	<ul style="list-style-type: none"> canPullInteractionresult: String
<ul style="list-style-type: none"> NoChildrenOpened 	<ul style="list-style-type: none"> CommandParameter: String WarningMessageTarget: String DoNotCheckForParentId: String

Default Commands	Parameters
<ul style="list-style-type: none"> NoDraftWithSameParentOpened 	<ul style="list-style-type: none"> ParentId: String
<ul style="list-style-type: none"> PullFromWorkbin 	<ul style="list-style-type: none"> CommandParameter: String WorkbinId: String
<ul style="list-style-type: none"> MoveToWorkbinOpenedParent 	<ul style="list-style-type: none"> ParentId: String
Chain InteractionEmailActionOnOpenedEmailById	
<ul style="list-style-type: none"> IsNotOpened 	<ul style="list-style-type: none"> CommandParameter: String - System.String OpenedInteractionCommand: String
Chain InteractionEmailClose	
<ul style="list-style-type: none"> IsContactModified 	<ul style="list-style-type: none"> CasId: String - System.String
<ul style="list-style-type: none"> IsPossibleToClose 	<ul style="list-style-type: none"> CommandParameter: IInteractionEmail - Genesyslab.Desktop.Modules.OpenMedia.Model.Interactions.Email.IInteractionEmail
<ul style="list-style-type: none"> Close 	<ul style="list-style-type: none"> CommandParameter: IInteractionEmail
Chain InteractionEmailDelete	
<ul style="list-style-type: none"> Confirm 	<ul style="list-style-type: none"> ActionToConfirm: String ConfirmParentWindow: Window System.Window.Window

Default Commands	Parameters
<ul style="list-style-type: none"> IsContactModified 	<ul style="list-style-type: none"> CasId: String - System.String
<ul style="list-style-type: none"> PullParentInboundEmail 	<ul style="list-style-type: none"> CommandParameter: IInteractionEmail DispositionCode: IDispositionCode - Genesyslab.Desktop.Modules.Core.Model.Interactions.IDispositionCode
<ul style="list-style-type: none"> Delete 	<ul style="list-style-type: none"> CommandParameter: IInteractionEmail
<ul style="list-style-type: none"> CleanConnectionInformation 	<ul style="list-style-type: none"> CommandParameter: String
Chain InteractionEmailDeleteById	
<ul style="list-style-type: none"> Confirm 	<ul style="list-style-type: none"> ActionToConfirm: String ConfirmParentWindow: Window
<ul style="list-style-type: none"> Delete 	<ul style="list-style-type: none"> CommandParameter: String WorkbinId: String
Chain InteractionEmailForward	
<ul style="list-style-type: none"> IsContactModified 	<ul style="list-style-type: none"> CasId: String - System.String
<ul style="list-style-type: none"> IsPossibleToForward 	<ul style="list-style-type: none"> CommandParameter: IInteractionInboundEmail - Genesyslab.Desktop.Modules.OpenMedia.Model.Interactions.Email.IInteractionInboundEmail

Default Commands	Parameters
<ul style="list-style-type: none"> AddForwardData 	<ul style="list-style-type: none"> CommandParameter: IInteractionInboundEmail Destination: String
<ul style="list-style-type: none"> SetAutoClose 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia
<ul style="list-style-type: none"> SaveAttachedData 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia UserData: KeyValueCollection
<ul style="list-style-type: none"> RequestSaveNotepad 	<ul style="list-style-type: none"> CommandParameter: IInteraction NotepadViewModel: INotepadViewModel
<ul style="list-style-type: none"> PlaceInQueue 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia Destination: String Reason: KeyValueCollection Extensions: KeyValueCollection
<ul style="list-style-type: none"> AddRecentCall 	<ul style="list-style-type: none"> RecentIndex: ObjectTC - Genesyslab.Desktop.Modules.TeamCommunicator.ObjectTC
Chain InteractionEmailsPossibleToClose	
<ul style="list-style-type: none"> IsContactModified 	<ul style="list-style-type: none"> CasId: String - System.String

Default Commands	Parameters
<ul style="list-style-type: none"> IsPossibleToClose 	<ul style="list-style-type: none"> CommandParameter: IInteractionEmail
Chain InteractionEmailLoad	
<ul style="list-style-type: none"> Load 	<ul style="list-style-type: none"> CommandParameter: IInteractionEmail
Chain InteractionEmailMarkDoneById	
<ul style="list-style-type: none"> NoChildrenOpened 	<ul style="list-style-type: none"> CommandParameter: String WarningMessageTarget: String DoNotCheckForParentId: String
<ul style="list-style-type: none"> Confirm 	<ul style="list-style-type: none"> ActionToConfirm: String ConfirmParentWindow: Window
<ul style="list-style-type: none"> MarkDone 	<ul style="list-style-type: none"> CommandParameter: String WorkbinId: String
Chain InteractionEmailMoveToWorkbin	
<ul style="list-style-type: none"> IsContactModified 	<ul style="list-style-type: none"> CasId: String - System.String
<ul style="list-style-type: none"> IsWorkbinDestinationDefined 	<ul style="list-style-type: none"> WorkbinId: String WorkbinOptionName: String CommandParameter: IInteractionOpenMedia

Default Commands	Parameters
	<ul style="list-style-type: none"> IgnoreInteractionFromWorkbinId: System Available since: 8.5.0
<ul style="list-style-type: none"> SetAttachedDadInformation 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia
<ul style="list-style-type: none"> Save 	<ul style="list-style-type: none"> CommandParameter: IInteractionEmail UserData: KeyValueCollection - Genesyslab.Enterprise.Commons.Collections.KeyValueCollection Available since: 8.5.0
<ul style="list-style-type: none"> MoveToWorkbin 	<ul style="list-style-type: none"> CommandParameter: String WorkbinId: String IgnoreInteractionFromWorkbinId: String Available since: 8.5.0
<ul style="list-style-type: none"> UpdateContactInformation 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia
Chain InteractionEmailPlaceInQueue	
<ul style="list-style-type: none"> UpdateInteractionUCS 	<ul style="list-style-type: none"> CommandParameter: IInteraction NotepadViewModel: INotepadViewModel
<ul style="list-style-type: none"> IsContactModified 	<ul style="list-style-type: none"> CasId: String

Default Commands	Parameters
<ul style="list-style-type: none"> Save 	<ul style="list-style-type: none"> CommandParameter: IInteractionEmail
<ul style="list-style-type: none"> SetAutoClose 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia - Genesyslab.Desktop.Modules.OpenMedia.Model.Interactions.IInteractionOpenMedia
<ul style="list-style-type: none"> SaveAttachedData 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia UserData: KeyValueCollection
<ul style="list-style-type: none"> PlaceInQueue 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia Destination: String Reasons: KeyValueCollection Extensions: KeyValueCollection
<ul style="list-style-type: none"> UpdateContactInformation 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia
<ul style="list-style-type: none"> AddRecentCall 	<ul style="list-style-type: none"> RecentIndex: ObjectTC
Chain InteractionEmailPrint	
<ul style="list-style-type: none"> Print 	<ul style="list-style-type: none"> CommandParameter: IInteractionEmail
Chain InteractionEmailPullById	
<ul style="list-style-type: none"> PullInteraction 	<ul style="list-style-type: none"> CommandParameter: IInteraction

Default Commands	Parameters
	<ul style="list-style-type: none"> • IW_WebcallbackInteractionId: String
Chain InteractionEmailPullFromWorkbinById	
<ul style="list-style-type: none"> • NoChildrenOpened 	<ul style="list-style-type: none"> • CommandParameter: String • WarningMessageTarget: String • DoNotCheckForParentId: String
<ul style="list-style-type: none"> • NoDraftWithSameParentOpened 	<ul style="list-style-type: none"> • ParentId: String
<ul style="list-style-type: none"> • PullFromWorkbin 	<ul style="list-style-type: none"> • CommandParameter: String • WorkbinId: String
<ul style="list-style-type: none"> • MoveToWorkbinOpenedParent 	<ul style="list-style-type: none"> • ParentId: String
Chain InteractionEmailReply	
<ul style="list-style-type: none"> • RequestSaveNotepad 	<ul style="list-style-type: none"> • CommandParameter: IInteraction • NotepadViewModel: INotepadViewModel
<ul style="list-style-type: none"> • IsContactModified 	<ul style="list-style-type: none"> • CaseId: String - System.String
<ul style="list-style-type: none"> • IsPossibleToCreate 	<ul style="list-style-type: none"> • CommandParameter: IInteractionInboundEmail - Genesyslab.Desktop.Modules.OpenMedia.Model.Interactions.Email.IInteractionInboundEmail

Default Commands	Parameters
<ul style="list-style-type: none"> IsWorkbinDestinationDefined 	<ul style="list-style-type: none"> WorkbinId: String WorkbinOptionName: String CommandParameter: IInteractionOpenMedia IgnoreInteractionFromWorkbinId: String Available since: 8.5.0
<ul style="list-style-type: none"> AddSignatureParameters 	<ul style="list-style-type: none"> CommandParameter: IInteractionInboundEmail
<ul style="list-style-type: none"> AddResponseSignatureParameters 	<ul style="list-style-type: none"> SignatureOptionValue: String ESDKInboundEmail: IInteraction - Genesyslab.Enterprise.Model.Interaction.IInteraction
<ul style="list-style-type: none"> Reply 	<ul style="list-style-type: none"> CommandParameter: IInteractionInboundEmail ReplyAll: String UserData: KeyValueCollection SignatureParameters: Dictionary - System.Collections.Generic.Dictionary<string, object>
<ul style="list-style-type: none"> InboundEmailSearchAndMoveToWorkbin 	<ul style="list-style-type: none"> CommandParameter: IInteractionEmail
Chain InteractionEmailReplyById	
<ul style="list-style-type: none"> IsPossibleToCreate 	<ul style="list-style-type: none"> CommandParameter: IInteractionInboundEmail

Default Commands	Parameters
<ul style="list-style-type: none"> NoChildrenOpened 	<ul style="list-style-type: none"> CommandParameter: String WarningMessageTarget: String DoNotCheckForParentId: String <p>Available since: 8.1.3</p>
<ul style="list-style-type: none"> IsNotOpened 	<ul style="list-style-type: none"> CommandParameter: String OpenedInteractionCommand: String
<ul style="list-style-type: none"> AddSignatureParameters 	<ul style="list-style-type: none"> CommandParameter: String
<ul style="list-style-type: none"> AddResponseSignatureParameters 	<ul style="list-style-type: none"> SignatureOptionValue: String ESDKInboundEmail: IInteraction
<ul style="list-style-type: none"> Reply 	<ul style="list-style-type: none"> CommandParameter: String WorkbinId: String ReplyAll: String UserData: KeyValueCollection SignatureParameters: Dictionary ESDKInboundEmail: IEmailInteraction - Genesyslab.Enterprise.Model.IEmailInteraction

Default Commands	Parameters
<ul style="list-style-type: none"> InboundEmailSearchAndMoveToWorkbin Available since: 8.5 	<ul style="list-style-type: none"> CommandParameter: System.String
Chain InteractionEmailResend	
<ul style="list-style-type: none"> IsPossibleToCreate 	<ul style="list-style-type: none"> CommandParameter: IInteractionInboundEmail
<ul style="list-style-type: none"> NoChildrenOpened Deprecated since: 8.5 	<ul style="list-style-type: none"> CommandParameter: String WarningMessageTarget: String DoNotCheckForParentId: String
<ul style="list-style-type: none"> AddSignatureParameters 	<ul style="list-style-type: none"> CommandParameter: String
<ul style="list-style-type: none"> AddResponseSignatureParameters 	<ul style="list-style-type: none"> SignatureOptionValue: String ESDKInboundEmail: IInteraction
<ul style="list-style-type: none"> Create 	<ul style="list-style-type: none"> CommandParameter: String ParentInteractionId: String InitialEmail: IMediaOpenMedia - Genesyslab.Desktop.Modules.OpenMedia.Model.Agents.IMediaOpenMedia InitialEmailId: String UserData: KeyValueCollection Extensions: KeyValueCollection

Default Commands	Parameters
	<ul style="list-style-type: none"> SignatureParameters: Dictionary<string, object>
Chain InteractionEmailSave	
<ul style="list-style-type: none"> Save 	<ul style="list-style-type: none"> CommandParameter: IInteractionEmail
Chain InteractionEmailSaveNewOutbound	
<ul style="list-style-type: none"> Save 	<ul style="list-style-type: none"> CommandParameter: IInteractionEmail
Chain InteractionEmailSend	
<ul style="list-style-type: none"> RequestSaveNotepad 	<ul style="list-style-type: none"> CommandParameter: IInteraction - Genesyslab.Desktop.Modules.Core.Model.Interactions.IInteraction NotepadViewModel: INotepadViewModel - Genesyslab.Desktop.Modules.Windows.Interactions.INotepadViewModel
<ul style="list-style-type: none"> IsPossibleToSend 	<ul style="list-style-type: none"> CommandParameter: IInteractionOutboundEmail - Genesyslab.Desktop.Modules.OpenMedia.Model.Interactions.Email.IInteractionOutbound
<ul style="list-style-type: none"> IsPossibleToClose 	<ul style="list-style-type: none"> CommandParameter: CommandParameter - CommandParameter
<ul style="list-style-type: none"> AddQAReviewData 	<ul style="list-style-type: none"> CommandParameter: IInteractionEmail
<ul style="list-style-type: none"> Save 	<ul style="list-style-type: none"> CommandParameter: IInteractionEmail UserData: KeyValueCollection

Default Commands	Parameters
	Available since: 8.5
<ul style="list-style-type: none"> • CheckEmailFieldsBeforeSend Available since: 8.5 	<ul style="list-style-type: none"> • CommandParameter:
<ul style="list-style-type: none"> • Send 	<ul style="list-style-type: none"> • CommandParameter: IInteractionOutboundEmail • Reasons: KeyValueCollection • Extensions: KeyValueCollection
Chain InteractionEmailSendInterimReply	
<ul style="list-style-type: none"> • IsPossibleToSend 	<ul style="list-style-type: none"> • CommandParameter: IInteractionOutboundEmail
<ul style="list-style-type: none"> • IsPossibleToClose 	<ul style="list-style-type: none"> • CommandParameter: IInteractionEmail
<ul style="list-style-type: none"> • AddQAReviewData 	<ul style="list-style-type: none"> • CommandParameter: IInteractionEmail
<ul style="list-style-type: none"> • SetAttachedDadaInformation 	<ul style="list-style-type: none"> • CommandParameter: IInteractionOpenMedia
<ul style="list-style-type: none"> • Save 	<ul style="list-style-type: none"> • CommandParameter: IInteractionEmail • UserData: KeyValueCollection Available since: 8.5

Default Commands	Parameters
<ul style="list-style-type: none"> • CheckEmailFieldsBeforeSend Available since: 8.5 	<ul style="list-style-type: none"> • CommandParameter:
<ul style="list-style-type: none"> • PullParentInboundEmail 	<ul style="list-style-type: none"> • CommandParameter: IInteractionEmail • DispositionCode: IDispositionCode
<ul style="list-style-type: none"> • Send 	<ul style="list-style-type: none"> • CommandParameter: IInteractionOutboundEmail • Reasons: KeyValueCollection • Extensions: KeyValueCollection
Chain InteractionEmailTransferAgent	
<ul style="list-style-type: none"> • Save 	<ul style="list-style-type: none"> • CommandParameter: IInteractionEmail
<ul style="list-style-type: none"> • SetAutoClose 	<ul style="list-style-type: none"> • CommandParameter: IInteractionOpenMedia
<ul style="list-style-type: none"> • RequestSaveNotepad 	<ul style="list-style-type: none"> • CommandParameter: IInteraction • NotepadViewModel: INotepadViewModel
<ul style="list-style-type: none"> • TransferAgent 	<ul style="list-style-type: none"> • CommandParameter: IInteractionOpenMedia • Destination: String • Reasons: KeyValueCollection • Extensions: KeyValueCollection

Default Commands	Parameters
<ul style="list-style-type: none"> AddRecentCall 	<ul style="list-style-type: none"> RecentIndex: ObjectTC
Chain InteractionEmailWorkflow	
<ul style="list-style-type: none"> PromptDone 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia ConfirmParentWindow: WindowSystem.Window.Window
<ul style="list-style-type: none"> Workflow 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia Workflow: String Reasons: KeyValueCollection Extensions: KeyValueCollection
Chain InteractionEmailWorkflowFromInteractionESDK	
<ul style="list-style-type: none"> ESDKWorkflow 	<ul style="list-style-type: none"> CommandParameter: IOpenMediaInteraction - Genesyslab.Enterprise.Model.Interaction.IOpenMediaInteraction Workflow: String Reasons: KeyValueCollection Extensions: KeyValueCollection AttachedDataInformation: IDispositionCode DispositionCode: KeyValueCollection
Chain InteractionInboundEmailPossibleToMoveToWorkbin	
<ul style="list-style-type: none"> IsWorkbinDestinationDefined 	<ul style="list-style-type: none"> WorkbinId: String

Default Commands	Parameters
	<ul style="list-style-type: none"> • WorkbinOptionName: String • CommandParameter: IInteractionOpenMedia • IgnoreInteractionFromWorkbinId: String Available since: 8.5
Chain InteractionInboundEmailSearchAndMoveToWorkbin	
<ul style="list-style-type: none"> • IsContactModified 	<ul style="list-style-type: none"> • CaseId: String
<ul style="list-style-type: none"> • InboundEmailSearchAndMoveToWorkbin 	<ul style="list-style-type: none"> • CommandParameter: IInteractionEmail
Chain InteractionOutboundEmailPossibleToMoveToWorkbin	
<ul style="list-style-type: none"> • IsWorkbinDestinationDefined 	<ul style="list-style-type: none"> • WorkbinId: String • WorkbinOptionName: String • CommandParameter: IInteractionOpenMedia • IgnoreInteractionFromWorkbinId: Added in 8.5.0
Chain InteractionOutboundEmailSearchAndMoveToWorkbin	
<ul style="list-style-type: none"> • IsContactModified 	<ul style="list-style-type: none"> • CaseId: String
<ul style="list-style-type: none"> • OutboundEmailSearchAndMoveToWorkbin 	<ul style="list-style-type: none"> • CommandParameter: IInteractionEmail
Commands common to all interactions	

Default Commands	Parameters
Chain InteractionSetAttachedDataWithDelay	
SetAttachedData	<ul style="list-style-type: none"> • CommandParameter: IInteraction • AttachedData: KeyValueCollection
Chain BundleClose	
GetAttachedDataInformationUCS	CommandParameter: IInteractionsBundle - Genesyslab.Desktop.Modules.Core.Model.Interactions.IInteractionsBundle
UpdateNotePadForVoice	<ul style="list-style-type: none"> • CommandParameter: IInteractionsBundle
ResetInteractionChatConsultation	CommandParameter: IInteractionsBundle
IsContactModified	<ul style="list-style-type: none"> • CaseId: String
IsPossibleToClose	CommandParameter: IInteractionsBundle
CompleteDispositionCodeOnBundle	CommandParameter: IInteractionsBundle
Close	CommandParameter: IInteractionsBundle
StopInteractionVoiceUCS	<ul style="list-style-type: none"> • CommandParameter: IInteractionsBundle
GetOutboundPreviewRecord	CommandParameter: ICampaign - Genesyslab.Enterprise.Model.Interaction.ICampaign

IM



Purpose: To provide information about customizable commands for the windows.

Learn about IM Commands

IM stands for Instant Messaging and is a feature provided by underlying SIP T-Servers and Switches. As a consequence, IM is not part of the open media offer. IM interactions are related to Voice interactions and voice media. In addition to the commands listed below, you can also read the following pages:

- [Commands for the Voice Media](#) to manage the creation of IM sessions.
- [Manage Toaster](#) which also accepts and rejects IM interactions.

Important

Read [Use Customizable Commands](#) to see code snippets which demonstrate the usage of the commands.

IM Commands

Chains of Commands to Manage IM Interactions

Default Commands	Parameters
Chain InteractionIMClose	
<ul style="list-style-type: none"> IsPossibleToClose 	<ul style="list-style-type: none"> CommandParameter: IInteractionIM -Genesyslab.Desktop.Modules.Voice.Model.Interactions.IInteractionIM
<ul style="list-style-type: none"> Close 	<ul style="list-style-type: none"> CommandParameter: IInteractionIM
Chain InteractionIMIsPossibleToClose	
<ul style="list-style-type: none"> IsPossibleToClose 	<ul style="list-style-type: none"> CommandParameter: IInteractionIM
Chain InteractionIMReleaseIM	
<ul style="list-style-type: none"> PromptEnd 	<ul style="list-style-type: none"> CommandParameter: IInteraction - IInteraction ConfirmParentWindow: Window
<ul style="list-style-type: none"> Release 	<ul style="list-style-type: none"> CommandParameter: IInteractionIM Reasons: KeyValueCollection - Genesyslab.Enterprise.Commons.Collections.KeyValueCollection Extensions: KeyValueCollection
Chain InteractionIMSendMessage	
<ul style="list-style-type: none"> SendMessage 	<ul style="list-style-type: none"> CommandParameter: IInteractionIM Content: string - System.String

Default Commands	Parameters
	<ul style="list-style-type: none"> • ContentType: string
Commands common to all interactions	
Chain InteractionSetAttachedDataWithDelay	
SetAttachedData	<ul style="list-style-type: none"> • CommandParameter: IInteraction • AttachedData: KeyValueCollection
Chain BundleClose	
GetAttachedDataInformationUCS	CommandParameter: IInteractionsBundle - Genesyslab.Desktop.Modules.Core.Model.Interactions.IInteractionsBundle
UpdateNotePadForVoice	<ul style="list-style-type: none"> • CommandParameter: IInteractionsBundle
ResetInteractionChatConsultation	CommandParameter: IInteractionsBundle
IsContactModified	<ul style="list-style-type: none"> • CaseId: String
IsPossibleToClose	CommandParameter: IInteractionsBundle
CompleteDispositionCodeOnBundle	CommandParameter: IInteractionsBundle
Close	CommandParameter: IInteractionsBundle

Default Commands	Parameters
StopInteractionVoiceUCS	<ul style="list-style-type: none">• CommandParameter: IInteractionsBundle
GetOutboundPreviewRecord	CommandParameter: ICampaign - Genesyslab.Enterprise.Model.Interaction.ICampaign

Open Media

Purpose: To provide information about customizable commands.

Learn about Open Media Commands

The open media interactions are top level interfaces implemented by e-mail, chat, and work item interactions. The open media commands presented in this page are basic operations which apply to open media interactions by inheritance: e-mail, chat, and workitem interactions. [E-Mails](#), [Chat](#), and [Work Items](#) are specific to e-mail, chat, and work items respectively, and present an addition to the commands detailed in this page. You can also use specific [Windows](#) which apply to open media interactions in some GUI components.

Important

Read [Use Customizable Commands](#) to see code snippets which demonstrate the usage of the commands.

Managing Open Media Interactions

Open media commands cover the following features (this list is not exhaustive):

- Accept and decline;
- Place in queue;
- Transfer;
- Move to workbins.

Chains of Commands to Manage Open Media Interactions

Default commands	Parameters
Chain InteractionOpenMediaCanPullInteraction	

Default commands	Parameters
<ul style="list-style-type: none"> CanPullInteraction 	<ul style="list-style-type: none"> CommandParameter: Dictionary - Dictionary<string, object>
Chain InteractionOpenMediaClose	
<ul style="list-style-type: none"> IsContactModified 	<ul style="list-style-type: none"> CasId: String
<ul style="list-style-type: none"> Close 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia
Chain InteractionOpenMediaCloseOnOutOfService	
<ul style="list-style-type: none"> UpdateInteractionUCS 	<ul style="list-style-type: none"> CommandParameter: IInteraction - Genesyslab.Desktop.Modules.Core.Model.Interactions.IInteraction
<ul style="list-style-type: none"> SmsReleaseParty 	<ul style="list-style-type: none"> CommandParameter: IInteractionSms - Genesyslab.Desktop.Modules.OpenMedia.Model.Interactions.Sms.IInteractionSms
<ul style="list-style-type: none"> ReleaseMyParty 	<ul style="list-style-type: none"> CommandParameter: IInteractionChat - Genesyslab.Desktop.Modules.OpenMedia.Model.Interactions.Chat.IInteractionChat AfterActionMode: String
<ul style="list-style-type: none"> CloseConsult 	<ul style="list-style-type: none"> CommandParameter: IInteractionChat
<ul style="list-style-type: none"> Close 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia - Genesyslab.Desktop.Modules.OpenMedia.Model.Interactions.IInteractionOpenMedia
Chain InteractionOpenMediaConferenceAgent	

Default commands	Parameters
<ul style="list-style-type: none"> ConferenceAgent 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia Visibility: String Destination: String Extensions: KeyValueCollection - Genesyslab.Enterprise.Commons.Collections.KeyValueCollection
Chain InteractionOpenMediaGetInteractionStateById	
<ul style="list-style-type: none"> GetInteractionState 	<ul style="list-style-type: none"> CommandParameter: String
Chain InteractionOpenMediaHandlingTimeout	
<ul style="list-style-type: none"> UpdateInteractionUCS 	<ul style="list-style-type: none"> CommandParameter: IInteraction
<ul style="list-style-type: none"> SmsReleaseParty 	<ul style="list-style-type: none"> CommandParameter: IInteractionSms
<ul style="list-style-type: none"> ReleaseMyParty 	<ul style="list-style-type: none"> CommandParameter: IInteractionChat AfterActionMode: String
<ul style="list-style-type: none"> CloseConsult 	<ul style="list-style-type: none"> CommandParameter: IInteractionChat
<ul style="list-style-type: none"> Close 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia
Chain InteractionOpenMediaMoveToQueue	

Default commands	Parameters
<ul style="list-style-type: none"> PlaceInQueue 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia Destination: String Reason: KeyValueCollection Extensions: KeyValueCollection
Chain InteractionOpenMediaMoveToWorkbin	
<ul style="list-style-type: none"> MoveToWorkbin 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia WorkbinId: String WorkbinParameter: String WorkbinType: WorkbinType - Genesyslab.Platform.OpenMedia.Protocols.OpenMediaAttributes.WorkbinType Deprecated in: 8.5
Chain InteractionOpenMediaPlaceInQueue	
<ul style="list-style-type: none"> UpdateInteractionUCS 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia NotepadViewModel: INotepadViewModel
<ul style="list-style-type: none"> IsContactModified 	<ul style="list-style-type: none"> CasId: String
<ul style="list-style-type: none"> SaveAttachedData 	<ul style="list-style-type: none"> CommandParameter: IInteraction UserData: KeyValueCollection

Default commands	Parameters
<ul style="list-style-type: none"> SetAutoClose 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia
<ul style="list-style-type: none"> PlaceInQueue 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia Destination: String Reasons: KeyValueCollection Extensions: KeyValueCollection
<ul style="list-style-type: none"> UpdateContactInformation 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia
<ul style="list-style-type: none"> AddRecentCall 	<ul style="list-style-type: none"> RecentIndex: ObjectTC - Genesyslab.Desktop.Modules.TeamCommunicator.ObjectTC
Chain InteractionOpenMediaPullById	
<ul style="list-style-type: none"> PullInteraction 	<ul style="list-style-type: none"> CommandParameter: String queue: String sourceType: String agentId: String
Chain InteractionOpenMediaPullFromWorkbinById	
<ul style="list-style-type: none"> GetInteractionState 	<ul style="list-style-type: none"> CommandParameter: String
<ul style="list-style-type: none"> CanPullInteraction 	<ul style="list-style-type: none"> CommandParameter: Dictionary

Default commands	Parameters
<ul style="list-style-type: none"> DisplayWarningPullNotPossible 	<ul style="list-style-type: none"> canPullInteraction (result: String)
<ul style="list-style-type: none"> PullFromWorkbin 	<ul style="list-style-type: none"> CommandParameter: String WorkbinId: String
Chain InteractionOpenMediaUCSStopProcessing	
<ul style="list-style-type: none"> StopProcessing 	<ul style="list-style-type: none"> CommandParameter: String UserData: KeyValueCollection MediaType: String
Chain InteractionOpenMediaUCSUpdate	
<ul style="list-style-type: none"> UpdateInteraction 	<ul style="list-style-type: none"> CommandParameter: IInteraction
Chain InteractionOpenMediaPullFromWorkbinById	
<ul style="list-style-type: none"> GetInteractionState 	<ul style="list-style-type: none"> CommandParameter: String
<ul style="list-style-type: none"> CanPullInteraction 	<ul style="list-style-type: none"> CommandParameter: Dictionary<string, object>
<ul style="list-style-type: none"> DisplayWarningPullNotPossible 	<ul style="list-style-type: none"> canPullInteraction (result: String)
<ul style="list-style-type: none"> PullFromWorkbin 	<ul style="list-style-type: none"> CommandParameter: String QueueId: String

Default commands	Parameters
Chain InteractionOpenMediaTransferAgent	
<ul style="list-style-type: none"> SetAutoClose 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia
<ul style="list-style-type: none"> RequestSaveNotepad 	<ul style="list-style-type: none"> CommandParameter: IInteraction NotepadViewModel: INotepadViewModel - Genesyslab.Desktop.Modules.Windows.Interactions.INotepadViewModel
<ul style="list-style-type: none"> TransferAgent 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia Destination: String Reasons: KeyValueCollection Extensions: KeyValueCollection
<ul style="list-style-type: none"> AddRecentCall 	<ul style="list-style-type: none"> RecentIndex: ObjectTC
Chain InteractionOpenMediaWorkflow	
<ul style="list-style-type: none"> PromptDone 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia ConfirmParentWindow: Window - System.Window.Window
<ul style="list-style-type: none"> Workflow 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia Workflow: String Reasons: KeyValueCollection Extensions: KeyValueCollection

Default commands	Parameters
Chain InteractionPossibleToMoveToWorkbin	
<ul style="list-style-type: none"> IsWorkbinDestinationDefined Available since: 8.1.3	<ul style="list-style-type: none"> WorkbinId: String WorkbinOptionName: String CommandParameter: IInteractionOpenMedia
Chain InteractionInternalConferenceInviteAccept	
<ul style="list-style-type: none"> Accept 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia Extensions: KeyValueCollection
Chain InteractionInternalConferenceInviteConferenceAgent	
<ul style="list-style-type: none"> ConferenceAgent 	<ul style="list-style-type: none"> CommandParameter: IInteractionInternalConferenceInvite - Genesyslab.Desktop.Modules.OpenMedia.Model.Interactions.Invite.IInteractionInternalC Visibility: String Destination: String Extensions: KeyValueCollection
Chain InteractionInternalConferenceInviteDecline	
<ul style="list-style-type: none"> Decline 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia Reasons: KeyValueCollection Extensions: KeyValueCollection
<ul style="list-style-type: none"> Close 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia

Default commands	Parameters
Chain InteractionInternalConferenceInvitePull	
<ul style="list-style-type: none"> • GetInteractionProperties 	<ul style="list-style-type: none"> • CommandParameter: String
<ul style="list-style-type: none"> • Pull 	<ul style="list-style-type: none"> • CommandParameter: String • Source: String Added in 8.5 • OwnerInteractionId: String Added in 8.5
Chain InteractionInternalConferenceInviteSendInvite	
<ul style="list-style-type: none"> • SetCollaborationQueueContext 	<ul style="list-style-type: none"> • CommandParameter: IInteractionInternalConferenceInvite • System.String: Queue, output parameter
<ul style="list-style-type: none"> • SetOwner 	<ul style="list-style-type: none"> • CommandParameter: IInteractionInternalConferenceInvite
<ul style="list-style-type: none"> • PlaceInQueueCollaboration 	<ul style="list-style-type: none"> • CommandParameter: IInteractionInternalConferenceInvite • Queue: String • Reason: KeyValueCollection • Extensions: KeyValueCollection • CorrelatorData: String
Chain InteractionInternalConferenceInviteStopProcessing	

Default commands	Parameters
<ul style="list-style-type: none"> StopProcessing 	<ul style="list-style-type: none"> CommandParameter: IOpenMediaInteraction - Genesyslab.Enterprise.Model.Interaction.IOpenMediaInteraction Reason: KeyValueCollection Extensions: KeyValueCollection CorrelatorData: String
Chain OpenMediaEsdkInteractionPutBackInOriginalLocation Added in 8.5	
<ul style="list-style-type: none"> PutBack 	
Commands common to all interactions	
Chain InteractionSetAttachedDataWithDelay	
SetAttachedData	<ul style="list-style-type: none"> CommandParameter: IInteraction AttachedData: KeyValueCollection
Chain BundleClose	
GetAttachedDataInformationUCS	CommandParameter: IInteractionsBundle - Genesyslab.Desktop.Modules.Core.Model.Interactions.IInteractionsBundle
UpdateNotePadForVoice	<ul style="list-style-type: none"> CommandParameter: IInteractionsBundle
ResetInteractionChatConsultation	CommandParameter: IInteractionsBundle
IsContactModified	<ul style="list-style-type: none"> Caseld: String

Default commands	Parameters
IsPossibleToClose	CommandParameter: IInteractionsBundle
CompleteDispositionCodeOnBundle	CommandParameter: IInteractionsBundle
Close	CommandParameter: IInteractionsBundle
StopInteractionVoiceUCS	<ul style="list-style-type: none">• CommandParameter: IInteractionsBundle
GetOutboundPreviewRecord	CommandParameter: ICampaign - Genesyslab.Enterprise.Model.Interaction.ICampaign

Outbound

Purpose: To provide information about customizable commands.

Learn about Outbound Commands

These commands manage outbound operations on campaigns, interactions, chains and records.

Important

Read [Use Customizable Commands](#) to see code snippets which demonstrate the usage of the commands.

Managing Chains and Records

The following commands cover:

Default Commands	Parameters
Chain CampaignGetPreviewRecord	
<ul style="list-style-type: none"> GetOutboundPreviewRecord 	CommandParameter: ICampaign - Genesyslab.Enterprise.Model.Interaction.ICampaign
Chain CancelOutboundChain	
<ul style="list-style-type: none"> UpdateOutboundRecord 	CommandParameter: IInteraction - Genesyslab.Desktop.Modules.Core.Model.Interactions.IInteraction
<ul style="list-style-type: none"> CancelOutboundChain 	
Chain ClosePullPreviewRecord	
<ul style="list-style-type: none"> RescheduleOutboundRecord 	<ul style="list-style-type: none"> CommandParameter: IInteraction
<ul style="list-style-type: none"> CancelOutboundChain 	
<ul style="list-style-type: none"> RejectOutboundRecord 	

Default Commands	Parameters
<ul style="list-style-type: none"> MarkProcessedOutboundChain 	<ul style="list-style-type: none"> CommandParameter: IInteraction
<ul style="list-style-type: none"> ClosePullPreviewRecord 	<ul style="list-style-type: none"> CommandParameter: IInteractionPullPreview - Genesyslab.Desktop.Modules.Outbound.Model.Interactions.IInt
Chain MarkProcessedOutboundChain	
<ul style="list-style-type: none"> MarkProcessedOutboundChain 	<ul style="list-style-type: none"> CommandParameter: IInteraction
Chain PossibleToClosePullPreview	
<ul style="list-style-type: none"> PossibleToClosePullPreview 	
Chain PossibleToClosePushPreview	
<ul style="list-style-type: none"> PossibleToClosePushPreview 	
Chain RejectOutboundChain	
<ul style="list-style-type: none"> UpdateOutboundRecord 	<ul style="list-style-type: none"> CommandParameter: IInteraction
<ul style="list-style-type: none"> RejectOutboundChain 	
Chain StartPreviewRecordOutbound	
<ul style="list-style-type: none"> StartPreviewRecord 	
Chain UpdateOutboundRecord	
<ul style="list-style-type: none"> UpdateOutboundRecord 	<ul style="list-style-type: none"> CommandParameter: IInteraction

Managing Outbound Interactions

Default Commands	Parameters
Chain InteractionPushPreviewAccept	
<ul style="list-style-type: none"> AcceptPushPreview 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia - Genesyslab.Desktop.Modules.OpenMedia.Model.Interactions.IIn Extensions: KeyValueCollection - Genesyslab.Enterprise.Commons.Collections.KeyValueCollectio

Default Commands	Parameters
Chain InteractionPushPreviewDecline	
<ul style="list-style-type: none"> DeclinePushPreview 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia Reasons: KeyValueCollection - Genesyslab.Enterprise.Commons.Collections.KeyValueCollection Extensions: KeyValueCollection
Chain InteractionPushPreviewWorkflow	
<ul style="list-style-type: none"> Workflow 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia Workflow: String Reasons: KeyValueCollection Extensions: KeyValueCollection
<ul style="list-style-type: none"> RescheduleOutboundRecord 	<ul style="list-style-type: none"> CommandParameter: IInteraction
<ul style="list-style-type: none"> CancelOutboundChain 	
<ul style="list-style-type: none"> RejectOutboundRecord 	
<ul style="list-style-type: none"> MarkProcessedOutboundChain 	<ul style="list-style-type: none"> CommandParameter: IInteraction
<ul style="list-style-type: none"> ClosePushPreviewRecord 	<ul style="list-style-type: none"> CommandParameter: IInteractionPushPreview - Genesyslab.Desktop.Modules.Outbound.Model.Interactions.IInt
Commands common to all interactions	
Chain InteractionSetAttachedDataWithDelay	
SetAttachedData	<ul style="list-style-type: none"> CommandParameter: IInteraction AttachedData: KeyValueCollection
Chain BundleClose	
GetAttachedDataInformationUCS	CommandParameter: IInteractionsBundle - Genesyslab.Desktop.Modules.Core.Model.Interactions.IInteractionsBundle
UpdateNotePadForVoice	<ul style="list-style-type: none"> CommandParameter: IInteractionsBundle

Default Commands	Parameters
ResetInteractionChatConsultation	CommandParameter: IInteractionsBundle
IsContactModified	• CaselId: String
IsPossibleToClose	CommandParameter: IInteractionsBundle
CompleteDispositionCodeOnBundle	CommandParameter: IInteractionsBundle
Close	CommandParameter: IInteractionsBundle
StopInteractionVoiceUCS	• CommandParameter: IInteractionsBundle
GetOutboundPreviewRecord	CommandParameter: ICampaign - Genesyslab.Enterprise.Model.Interaction.ICampaign

Place and Media

Purpose: To provide information about customizable commands which modify the media of a given place.

Learn about Place and Media Commands

The commands presented in this page enable your application to manage the agent activity (login, ready, not ready, log off) on the media of a given place. Commands for the media of type Open Media apply to Chat, E-Mail, and Work Item media.

Important

Read [Use Customizable Commands](#) to see code snippets which demonstrate the usage of the commands.

Manage Open Media

The following commands let you manage media of type open media, including work items, chat and e-mail (which inherit from open media):

- Change the media status (ready, not ready, login, log off);
- Activate or deactivate *Do Not Disturb* (DND) features;
- Modify the state reason for a given media.

Important

These commands do not apply to DNs (voice media).

Default Commands	Parameters
Chain MediaOpenMediaChangeStateReason	
<ul style="list-style-type: none"> ChangeStateReason 	<ul style="list-style-type: none"> EnterpriseAgent: IAgent - Genesyslab.Enterprise.Agent.IAgent Device: IDevice - Genesyslab.Enterprise.Model.Device.IDevice Reason: KeyValueCollection - Genesyslab.Enterprise.Commons.Collections.KeyValueCollection Extensions: KeyValueCollection
Chain MediaOpenMediaDNDOff	
<ul style="list-style-type: none"> DNDOff 	<ul style="list-style-type: none"> EnterpriseAgent: IAgent Channel: IClientChanel - Genesyslab.Enterprise.Model.Channel.IClientChanel Reason: KeyValueCollection { reasonName } - Genesyslab.Enterprise.Commons.Collections.KeyValueCollection Extensions: KeyValueCollection
Chain MediaOpenMediaDNDOn	
<ul style="list-style-type: none"> DNDOn 	<ul style="list-style-type: none"> EnterpriseAgent: IAgent Channel: IClientChanel Reason: KeyValueCollection Extensions: KeyValueCollection
Chain MediaOpenMediaLogOff	
<ul style="list-style-type: none"> LogOff 	<ul style="list-style-type: none"> Channel: IClientChanel

Default Commands	Parameters
	<ul style="list-style-type: none"> • DevicePlace: IDevice • Device: IDevice • Reason: KeyValueCollection • Extensions: KeyValueCollection
Chain MediaOpenMediaLogOffPlace	
<ul style="list-style-type: none"> • LogOffPlace 	<ul style="list-style-type: none"> • Channel: IClientChanel • EnterpriseAgent: IAgent • Reason: KeyValueCollection { reasonName } • Extensions: KeyValueCollection
Chain MediaOpenMediaLogOn	
<ul style="list-style-type: none"> • UpdateInteractionSignature 	
<ul style="list-style-type: none"> • LogOn 	<ul style="list-style-type: none"> • Channel: IClientChanel • DevicePlace: IDevice • Device: IDevice • Reason: KeyValueCollection { reasonName } • Extensions: KeyValueCollection
Chain MediaOpenMediaLogOnPlace	
<ul style="list-style-type: none"> • UpdateInteractionSignature 	

Default Commands	Parameters
<ul style="list-style-type: none"> LogOnPlace 	<ul style="list-style-type: none"> EnterpriseAgent: IAgent Channel: IClientChanel Reason: KeyValueCollection { reasonName} Extensions: KeyValueCollection
Chain MediaOpenMediaNotReady	
<ul style="list-style-type: none"> UpdateInteractionSignature 	
<ul style="list-style-type: none"> NotReady 	<ul style="list-style-type: none"> Channel: IClientChanel DevicePlace: IDevice Device: IDevice Reason: KeyValueCollection { reasonName } Extensions: KeyValueCollection
Chain MediaOpenMediaReady	
<ul style="list-style-type: none"> UpdateInteractionSignature 	
<ul style="list-style-type: none"> Ready 	<ul style="list-style-type: none"> Channel: IClientChanel DevicePlace: IDevice Device: IDevice Reason: KeyValueCollection { reasonName }

Default Commands	Parameters
	<ul style="list-style-type: none"><li data-bbox="1144 320 1541 344">• Extensions: KeyValueCollection

Manage SMS Media

You can also create new outbound sms through the following commands:

Default Commands	Parameters
Chain MediaSmsCreateNewOutboundPageMode	
<ul style="list-style-type: none"> IsPossibleToCreate 	<ul style="list-style-type: none"> SourcePhoneNumber: String
<ul style="list-style-type: none"> ThreadingOpenMedia 	<ul style="list-style-type: none"> ParentInteractionId: String
<ul style="list-style-type: none"> Create 	<ul style="list-style-type: none"> CommandParameter: IMediaOpenMedia - Genesyslab.Desktop.Modules.OpenMedia.Model.Agents.IMediaOpenMedia Subtype: String ParentInteractionId: String (ESDK Id) SourcePhoneNumber: String DestinationPhoneNumber: String InputQueues: ICollection<string> - System.Collections.Generic.ICollection OutputQueues: ICollection<string> UserData: KeyValueCollection Extensions: KeyValueCollection CorrelatorData: String
<ul style="list-style-type: none"> AddRecentCall 	<ul style="list-style-type: none"> RecentIndex: ObjectTC

Manage E-Mail Media

In addition to [the Open Media commands](#), you can also create new outbound e-mails through the following commands:

Default Commands	Parameters
Chain MediaEmailCreateNewOutboundEmail	
<ul style="list-style-type: none"> IsPossibleToCreate 	<ul style="list-style-type: none"> CommandParameter: IInteractionInboundEmail

Default Commands	Parameters
<ul style="list-style-type: none"> AddSignatureParameters 	<ul style="list-style-type: none"> CommandParameter: IMediaOpenMedia
<ul style="list-style-type: none"> AddResponseSignatureParameters 	<ul style="list-style-type: none"> SignatureOptionValue: String ESDKInboundEmail: Interaction
<ul style="list-style-type: none"> ThreadingOpenMedia 	<ul style="list-style-type: none"> ParentInteractionId: String
<ul style="list-style-type: none"> Create 	<ul style="list-style-type: none"> CommandParameter: IMediaOpenMedia ParentInteractionId: String InitialEmail: String UserData: KeyValueCollection Extensions: KeyValueCollection SignatureParameters: Dictionary<string, object>
<ul style="list-style-type: none"> AddRecentCall 	<ul style="list-style-type: none"> RecentIndex: ObjectTC

Manage Voice Media

The following commands apply to the voice media only. The voice media is composed of Directory Numbers (DNs) available on the underlying switches. Through the below commands, you can:

- Change the media status (ready, not ready, login, log off);
- Activate or deactivate the *Do Not Disturb* (DND) features;
- Start a new call;
- Manage a new Instant Messaging session.

Default Commands	Parameters
Chain MediaVoiceDNDOff	
<ul style="list-style-type: none"> • DNDOff 	<ul style="list-style-type: none"> • EnterpriseAgent: IAgent • Channel: IClientChanel • Reasons: KeyValueCollection • Extensions: KeyValueCollection
Chain MediaVoiceDNDOOn	
<ul style="list-style-type: none"> • DNDOOn 	<ul style="list-style-type: none"> • EnterpriseAgent: IAgent • Channel: IClientChanel • Reasons: KeyValueCollection • Extensions: KeyValueCollection
Chain MediaVoiceLogOff	
<ul style="list-style-type: none"> • NegotiatedLogout 	<ul style="list-style-type: none"> • Media: IMedia - Genesyslab.Desktop.Modules.Core.Model.Agents.IMedia
<ul style="list-style-type: none"> • TLibUpdateSignature 	
<ul style="list-style-type: none"> • SipEndpointPossibleToLogOff Available since: 8.5 	
<ul style="list-style-type: none"> • LogOff 	<ul style="list-style-type: none"> • EnterpriseAgent: IAgent • Channel: IClientChanel

Default Commands	Parameters
	<ul style="list-style-type: none"> • Queue: String • Password: String • Reasons: KeyValueCollection • Extensions: KeyValueCollection
<ul style="list-style-type: none"> • SipEndpointLogOff Available since: 8.5 	
Chain MediaVoiceLogOn	
<ul style="list-style-type: none"> • UpdateTlibSignature 	
<ul style="list-style-type: none"> • SipEndpointLogOn Available since: 8.5 	
<ul style="list-style-type: none"> • LogOn 	<ul style="list-style-type: none"> • EnterpriseAgent: IAgent • Channel: IClientChanel • Queue: String • WorkMode: Identity - Genesyslab.Enterprise.Model.Identity.workmode • Password: String • Reasons: KeyValueCollection • Extensions: KeyValueCollection
Chain MediaVoiceMakeCall	

Default Commands	Parameters
<ul style="list-style-type: none"> CleanUpInteractionCode 	<ul style="list-style-type: none"> CommandParameter: IMediaVoice - Genesyslab.Desktop.Modules.Core.Model.Agents.IMediaVoice UserData: KeyValueCollection
<ul style="list-style-type: none"> ThreadingVoice 	<ul style="list-style-type: none"> UserData: String
<ul style="list-style-type: none"> MakeCall 	<ul style="list-style-type: none"> CommandParameter: IMediaVoice Destination: String Location: String MakeCallType: Genesyslab.Enterprise.Model.Interaction.MakeCallType UserData: KeyValueCollection Reasons: KeyValueCollection Extensions: KeyValueCollection
<ul style="list-style-type: none"> AddRecentCall 	<ul style="list-style-type: none"> RecentIndex: ObjectTC
Chain MediaVoiceMakeIM	
<ul style="list-style-type: none"> MakeIM 	<ul style="list-style-type: none"> CommandParameter: IMediaVoice Destination: String Location: String MakeCallType: Genesyslab.Enterprise.Model.Interaction.MakeCallType UserData: KeyValueCollection

Default Commands	Parameters
	<ul style="list-style-type: none"> • Reasons: KeyValueCollection • Extensions: KeyValueCollection
<ul style="list-style-type: none"> • AddRecentCall 	<ul style="list-style-type: none"> • RecentIndex: ObjectTC
Chain MediaVoiceNotReady	
<ul style="list-style-type: none"> • UpdateTLibSignature 	
<ul style="list-style-type: none"> • NotReady 	<ul style="list-style-type: none"> • EnterpriseAgent: IAgent • Channel: IClientChanel • Queue: String • WorkMode: Identity • Reasons: KeyValueCollection • Extensions: KeyValueCollection
Chain MediaVoiceReady	
<ul style="list-style-type: none"> • UpdateTLibSignature 	
<ul style="list-style-type: none"> • Ready 	<ul style="list-style-type: none"> • EnterpriseAgent: IAgent • Channel: IClientChanel • Queue: String

Default Commands	Parameters
	<ul style="list-style-type: none">• WorkMode: Identity• Reasons: KeyValueCollection• Extensions: KeyValueCollection
Chain MediaVoiceStartIMSession	
<ul style="list-style-type: none">• Start	

SMS

Purpose: To provide information about customizable commands.

Learn about SMS Commands

These commands manage operations on SMS interactions.

Important

Read [Use Customizable Commands](#) to see code snippets which demonstrate the usage of the commands.

Managing SMS Interactions

Chains of Commands to Manage SMS Interactions

Default Commands	Parameters
Chain InteractionSmsAccept	
<ul style="list-style-type: none"> Accept 	CommandParameter: InteractionSms - Genesyslab.Desktop.Modules.OpenMedia.Model.Interactions.Sms.IInteractionSms
Chain InteractionSmsCreateReplyMessage	
<ul style="list-style-type: none"> IsPossibleToCreate 	SourcePhoneNumber: String - System.String
<ul style="list-style-type: none"> IsItPossibleToSendPageSms 	OutboundQueue: String
<ul style="list-style-type: none"> SetContext 	CorrelatorData: String

Default Commands	Parameters
<ul style="list-style-type: none"> • Create 	<ul style="list-style-type: none"> • CommandParameter: IMediaOpenMedia - Genesyslab.Desktop.Modules.OpenMedia.Model.Agents.IMediaOpenMedia • Subtype: String • ParentInteractionId: String(Enterprise Service ID) • SourcePhoneNumber: String • DestinationPhoneNumber: String • InputQueues: ICollection - System.Collections.Generic.ICollection • OutputQueues: ICollection - • UserData: KeyValueCollection - Genesyslab.Enterprise.Commons.Collections.KeyValueCollection • Extensions: KeyValueCollection • CorrelatorData: String
<ul style="list-style-type: none"> • CreateUCS 	<ul style="list-style-type: none"> • EsdkInteraction: IInteraction - Genesyslab.Enterprise.Model.Interaction.IInteraction • ParentInteractionId: String
Chain InteractionSmsDecline	
<ul style="list-style-type: none"> • Decline 	<ul style="list-style-type: none"> • CommandParameter: IInteractionOpenMedia - Genesyslab.Desktop.Modules.OpenMedia.Model.Interactions.IInteractionOpenMedia • Reasons: KeyValueCollection • Extensions: KeyValueCollection

Default Commands	Parameters
<ul style="list-style-type: none"> Close 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia
Chain InteractionSmsDelete	
<ul style="list-style-type: none"> Confirm 	<ul style="list-style-type: none"> ActionToConfirm: String ConfirmParentWindow: Window - System.Window.Window
<ul style="list-style-type: none"> IsContactModified 	<ul style="list-style-type: none"> Caseld: String
<ul style="list-style-type: none"> Delete 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia
Chain InteractionSmsDisconnectSmsSessionChannel	
<ul style="list-style-type: none"> Disconnect 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia
Chain InteractionSmsGetHistoricMessage	
<ul style="list-style-type: none"> ExtractInteractions 	<ul style="list-style-type: none"> CommandParameter: String (contactId) ListAttributes: StringList - Genesyslab.Platform.Contacts.Protocols.ContactServer.StringList MediaTypes: String (list of media type seperated by comma.) TimeFrame: System (time frame in hour) ExcludedInteractions: IList<T> - System.Collections.Generic.IList<T> (interaction id) InteractionsUCS: IList<T> (output parameter).

Default Commands	Parameters
<ul style="list-style-type: none"> DecodeInteractions 	<ul style="list-style-type: none"> InteractionsUCS: IList<T> SmsMessages: ArrayList - System.Collections.ArrayList</ref>; the collection contains [ISmsMessage - Genesyslab.Desktop.Modules.Contacts.Decoder.Sms.ISmsMessage, output parameter].
<ul style="list-style-type: none"> Processing 	<ul style="list-style-type: none"> ModelView: ISmsViewModel - Genesyslab.Desktop.Modules.OpenMedia.Windows.Interactions.MediaView.Sms.Interacti Interaction: Interaction - Genesyslab.Desktop.Modules.Core.Model.Interactions.Interaction SmsMessages: ArrayList - System.Collections.ArrayList; the collection contains [ISmsMessage- Genesyslab.Desktop.Modules.Contacts.Decoder.Sms.ISmsMessage, output parameter].
<p>Chain InteractionSmsIsPossibleToClose</p>	
<ul style="list-style-type: none"> IsContactModified 	<ul style="list-style-type: none"> Caseld: String
<ul style="list-style-type: none"> IsPossibleToClose 	<ul style="list-style-type: none"> CommandParameter: IInteractionSms- Genesyslab.Desktop.Modules.OpenMedia.Model.Interactions.Sms.IInteractionSms
<p>Chain InteractionSmsPlaceInQueue</p>	
<ul style="list-style-type: none"> UpdateInteractionUCS 	<ul style="list-style-type: none"> CommandParameter: IInteraction NotepadViewModel: INotepadViewModel

Default Commands	Parameters
<ul style="list-style-type: none"> IsContactModified 	<ul style="list-style-type: none"> CasId: String
<ul style="list-style-type: none"> SaveAttachedData 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia UserData: KeyValueCollection
<ul style="list-style-type: none"> SetAutoClosePageMode 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia
<ul style="list-style-type: none"> PlaceInQueue 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia Destination: String Reason: KeyValueCollection Extensions: KeyValueCollection
<ul style="list-style-type: none"> SetAutoCloseSessionMode 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia
<ul style="list-style-type: none"> ReleaseParty 	<ul style="list-style-type: none"> CommandParameter: IInteractionSms
<ul style="list-style-type: none"> AddRecentCall 	<ul style="list-style-type: none"> RecentIndex: ObjectTC - Genesyslab.Desktop.Modules.TeamCommunicator.ObjectTC
Chain InteractionSmsSendMessage	
<ul style="list-style-type: none"> IsContactModified 	<ul style="list-style-type: none"> CasId: String

Default Commands	Parameters
<ul style="list-style-type: none"> IsDispositionCodeMandatory 	<ul style="list-style-type: none"> CommandParameter: IInteractionSms
<ul style="list-style-type: none"> SendMessage 	<ul style="list-style-type: none"> CommandParameter: IInteractionSms Visibility: String Message: String
<ul style="list-style-type: none"> MimizeWindow 	<ul style="list-style-type: none"> CommandParameter: IInteractionSms Window: Window InteractionContainerType: Genesyslab.Desktop.Modules.Windows.Views.Interactions.Container
Chain InteractionSmsSendMessage	
<ul style="list-style-type: none"> IsContactModified 	<ul style="list-style-type: none"> CasId: String
<ul style="list-style-type: none"> IsItPossibleToSendPageSms 	<ul style="list-style-type: none"> OutboundQueue: String
<ul style="list-style-type: none"> Send 	<ul style="list-style-type: none"> CommandParameter: IInteractionSms OutboundQueue: String SourcePhoneNumber: String Reason: KeyValueCollection Extensions: KeyValueCollection CorrelatorData: String

Default Commands	Parameters
<ul style="list-style-type: none"> UpdateInteractionUCS 	<ul style="list-style-type: none"> CommandParameter: IInteraction
Chain InteractionSmsSendReplyMessage	
<ul style="list-style-type: none"> IsContactModified 	<ul style="list-style-type: none"> CasId: String
<ul style="list-style-type: none"> IsItPossibleToSendPageSms 	<ul style="list-style-type: none"> OutboundQueue: String
<ul style="list-style-type: none"> Send 	<ul style="list-style-type: none"> CommandParameter: ISMSInteraction - Genesyslab.Enterprise.Model.Interaction.ISMSInteraction OutboundQueue: String Reason: KeyValueCollection Extensions: KeyValueCollection CorrelatorData: String
<ul style="list-style-type: none"> CloseParentInteraction 	<ul style="list-style-type: none"> ParentInteraction: CommandParameter - CommandParameter
Chain InteractionSmsTransferAgent	
<ul style="list-style-type: none"> SetAutoClosePageMode 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia
<ul style="list-style-type: none"> RequestSaveNotepad 	<ul style="list-style-type: none"> CommandParameter: IInteraction NotepadViewModel: INotepadViewModel - Genesyslab.Desktop.Modules.Windows.Interactions.INotepadViewModel

Default Commands	Parameters
<ul style="list-style-type: none"> TransferAgent 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia Destination: String Reasons: KeyValueCollection Extensions: KeyValueCollection
<ul style="list-style-type: none"> SetAutoCloseSessionMode 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia
<ul style="list-style-type: none"> ReleaseParty 	<ul style="list-style-type: none"> CommandParameter: IInteractionSms
<ul style="list-style-type: none"> AddRecentCall 	<ul style="list-style-type: none"> RecentIndex: ObjectTC
Chain InteractionSmsWorkflow	
<ul style="list-style-type: none"> PromptDone 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia ConfirmParentWindow: Window - System.Window.Window
<ul style="list-style-type: none"> ExitJoinIssue Available since: 8.5 	<ul style="list-style-type: none"> CommandParameter: Genesyslab.Desktop.Modules.OpenMedia.Model.Interactions.Sms.IInteractionSms Reason: KeyValueCollection Extensions: KeyValueCollection
<ul style="list-style-type: none"> ReleaseParty 	<ul style="list-style-type: none"> CommandParameter: IInteractionSms

Default Commands	Parameters
<ul style="list-style-type: none"> UpdateInteractionUCS 	<ul style="list-style-type: none"> CommandParameter: IInteraction
<ul style="list-style-type: none"> Workflow 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia Workflow: String Reasons: KeyValueCollection Extensions: KeyValueCollection
Commands common to all interactions	
Chain InteractionSetAttachedDataWithDelay	
SetAttachedData	<ul style="list-style-type: none"> CommandParameter: IInteraction AttachedData: KeyValueCollection
Chain BundleClose	
GetAttachedDataInformationUCS	CommandParameter: IInteractionsBundle - Genesyslab.Desktop.Modules.Core.Model.Interactions.IInteractionsBundle
UpdateNotePadForVoice	<ul style="list-style-type: none"> CommandParameter: IInteractionsBundle
ResetInteractionChatConsultation	CommandParameter: IInteractionsBundle
IsContactModified	<ul style="list-style-type: none"> Caseld: String

Default Commands	Parameters
IsPossibleToClose	CommandParameter: IInteractionsBundle
CompleteDispositionCodeOnBundle	CommandParameter: IInteractionsBundle
Close	CommandParameter: IInteractionsBundle
StopInteractionVoiceUCS	<ul style="list-style-type: none">• CommandParameter: IInteractionsBundle
GetOutboundPreviewRecord	CommandParameter: ICampaign - Genesyslab.Enterprise.Model.Interaction.ICampaign

Supervision

Purpose: To provide information about customizable commands for the supervision features.

Learn about Supervision

The following commands apply to supervisor buttons available in toolbars and toaster components that your application can customize. For further information, see [Views and Regions](#), which lists the type of available components.

Important

Read [Use Customizable Commands](#) to see code snippets which demonstrate the usage of the commands.

Manage Monitoring

The following commands are use to manage supervision.

Chains of Commands to Manage Monitoring

Default Commands	Parameters
Chain TeamLeadEndMonitoring	
<ul style="list-style-type: none"> EndMonitoring 	
Chain TeamLeadEndMonitoringChat	
<ul style="list-style-type: none"> EndMonitoringChat 	
Chain TeamLeadEndMonitoringVoice	
<ul style="list-style-type: none"> EndMonitoringVoice 	
Chain TeamLeadIntrudeCurrentChat	
<ul style="list-style-type: none"> IntrudeCurrentChat 	
<ul style="list-style-type: none"> AddRecentCall 	<ul style="list-style-type: none"> RecentIndex: ObjectTC
Chain TeamLeadIntrudeCurrentVoice	
<ul style="list-style-type: none"> IntrudeCurrentVoice 	
<ul style="list-style-type: none"> AddRecentCall 	<ul style="list-style-type: none"> RecentIndex: ObjectTC
Chain TeamLeadIsItPossibleToMonitor	
<ul style="list-style-type: none"> IsItPossibleToMonitor 	<ul style="list-style-type: none"> CommandParameter: String (employeeID)

Default Commands	Parameters
	<ul style="list-style-type: none"> IsItPossibleToMonitor: Boolean - System.Boolean, output parameter
Chain TeamLeadMonitorAll	
<ul style="list-style-type: none"> MonitorAllVoiceCalls 	
<ul style="list-style-type: none"> MonitorAllChatInteractions 	
<ul style="list-style-type: none"> AddRecentCall 	<ul style="list-style-type: none"> RecentIndex: ObjectTC - Genesyslab.Desktop.Modules.TeamCommunicator.ObjectTC
<ul style="list-style-type: none"> AddToTrackingList 	
Chain TeamLeadSetSupervisorBargelnChat	
<ul style="list-style-type: none"> SetSupervisorBargelnChat 	
Chain TeamLeadSetSupervisorBargelnVoice	
<ul style="list-style-type: none"> SetSupervisorBargelnVoice 	
Chain TeamLeadSetSupervisorCoachChat	
<ul style="list-style-type: none"> SetSupervisorCoachChat 	
Chain TeamLeadSetSupervisorCoachVoice	

Default Commands	Parameters
• SetSupervisorCoachVoice	
Chain TeamLeadSetSupervisorMuteVoice	
• SetSupervisorMuteVoice	
Chain TeamLeadStopMonitor	
• StopMonitorVoiceCalls	
• StopMonitorChatInteractions	
• RemoveFromTrackingList	
• AddRecentCall	• RecentIndex: ObjectTC

UCS



Purpose: To present for each UCS and contact command, the list of default subcommands available and the associated parameters.

Learn about the UCS Commands

The UCS commands presented in this page can apply to any type of interaction.

Important

Read [Use Customizable Commands](#) to see code snippets which demonstrate the usage of the commands.

Managing Interactions in UCS

The commands presented in this section are specific to UCS management.

Chains of Commands to UCS Operations for Interactions

Default Commands	Parameters
Chain AskSaveModifiedContact	
<ul style="list-style-type: none"> IsContactModified 	<ul style="list-style-type: none"> CaselId: String - System.String
Chain ContactAssignInteractionToContact	
<ul style="list-style-type: none"> Assign 	<ul style="list-style-type: none"> DestinationContactId: String ContactIdBeforeMerge: String UCSInteractionId: String AlertSectionName: String AssignInteractionToContactError: String, output parameter
Chain ContactAssignInteractionToContactForCustom	
<ul style="list-style-type: none"> AssignInteractionToContact 	<ul style="list-style-type: none"> DestinationContactId: String

Default Commands	Parameters
	<ul style="list-style-type: none"> • ContactIdBeforeMerge: String • UCSInteractionId: String • AlertSectionName: String • AssignInteractionToContactError: String, output parameter
<ul style="list-style-type: none"> • GetContact 	<ul style="list-style-type: none"> • CommandParameter: String (ContactId) • ContactAttributes: IList - System.Collections.Generic.IList, if key is missing get ContactAttributes defined in option 'contact.displayed-attributes' • IWContact: IIWContact - Genesyslab.Desktop.Modules.Contacts.IWContact.IIWContact, output parameter
<ul style="list-style-type: none"> • RefreshInformation 	<ul style="list-style-type: none"> • IWContact: IIWContact - Genesyslab.Desktop.Modules.Contacts.IWContact.IIWContact, output parameter • CaseId: String • InteractionId : String
Chain ContactAttributeInfoOrder	
<ul style="list-style-type: none"> • GetContactAttributeInfoOrder 	<ul style="list-style-type: none"> • AttributesListOrdered:IContactAttributeField[], output parameter
Chain ContactCreateUCSInteraction	
<ul style="list-style-type: none"> • Create 	<ul style="list-style-type: none"> • ContactId: String • InteractionToCreate: IInteraction • InteractionUCSId: String, output parameter
Chain CreateContactUCS Available since: 8.5.153.05	
<ul style="list-style-type: none"> • CreateContact 	<ul style="list-style-type: none"> • ContactAttributes: Genesyslab.Platform.Contacts.Protocols.ContactServer.Attributes • ContactHandlingResponse: Genesyslab.Desktop.Modules.Contacts.Requests.IContactHandlingResponse
Chain CreateInteractionUCSFromESDKInteraction	

Default Commands	Parameters
<ul style="list-style-type: none"> • CreateUCS 	<ul style="list-style-type: none"> • EsdkInteraction: IInteraction - Genesyslab.Enterprise.Model.Interaction.IInteraction • ParentInteractionId: String - System.String • StartDate: DateTime - System.DateTime
Chain DeleteInteractionUCS	
<ul style="list-style-type: none"> • DeleteUCS 	<ul style="list-style-type: none"> • CommandParameter: String
Chain ForceToCloseCase Available since: 8.5.0	
<ul style="list-style-type: none"> • CloseCase 	<ul style="list-style-type: none"> • System.string (CaseId)
Chain GetContactUCS	
<ul style="list-style-type: none"> • GetContact 	<ul style="list-style-type: none"> • CommandParameter: String (ContactId) • ContactAttributes: IList, if key is missing get ContactAttributes defined in option 'contact.displayed-attributes' • IWContact: IIWContact - Genesyslab.Desktop.Modules.Contacts.IWContact.IWContact, output parameter
Chain InteractionUCSInQueue	
<ul style="list-style-type: none"> • GetInteraction 	<ul style="list-style-type: none"> • CommandParameter: String (contactId) • ListAttributes: StringList - Genesyslab.Platform.Contacts.Protocols.ContactServer.StringList • ExcludedInteractions: IList <String > (list of String interaction id) • CurrentInteraction : IInteraction • InteractionsUCS: GenericList<IDictionary<string,object>>, output parameter - System.Collections.GenericList
Chain SetLastCalledAgentUCS	
<ul style="list-style-type: none"> • SetLCA 	
Chain ContactNotification	
<ul style="list-style-type: none"> • GetInteraction 	<ul style="list-style-type: none"> • CommandParameter: String (contactId) -

Default Commands	Parameters
	<ul style="list-style-type: none"> System.String (contactId) • TimeFrame: System (time frame in day) - System (time frame in day) • ListAttributes: StringList - Genesyslab.Platform.Contacts.Protocols.ContactServer.StringList • ExcludedInteractions: IList<String>(interaction id) • CurrentInteraction : IInteraction • InteractionsUCS: GenericList<IDictionary<string,object>> - System.Collections.GenericList<IDictionary<string,object>>, output parameter
Chain ContactOpenSearchWindow	
<ul style="list-style-type: none"> • Open 	<ul style="list-style-type: none"> • Parent: UserControl - System.Windows.Controls.UserControl
Chain ContactOpenToDialog	
<ul style="list-style-type: none"> • Open 	<ul style="list-style-type: none"> • ActionTarget: ActionTarget - Genesyslab.Desktop.Modules.Windows.Interactions.ActionTarget
Chain UpdateContactUCS Available since: 8.5.153.05	
<ul style="list-style-type: none"> • UpdateContact 	<ul style="list-style-type: none"> • CommandParameter: System.String (ContactId) • DeleteAttributes: Genesyslab.Platform.Contacts.Protocols.ContactServer.AttributeList • InsertAttributes: Genesyslab.Platform.Contacts.Protocols.ContactServer.AttributeList • UpdateAttributes: Genesyslab.Platform.Contacts.Protocols.ContactServer.AttributeList • ContactHandlingResponse: Genesyslab.Desktop.Modules.Contacts.Requests.IContactHandlingResponse

Voice

Purpose: To provide information about customizable commands.

Learn about Voice Commands

The commands presented in this page enable your application to manage voice interactions. In addition, to create a new voice interaction, see [Manage Voice Media](#).

Important

Read [Use Customizable Commands](#) to see code snippets which demonstrate the usage of the commands.

Managing Voice Interactions

The commands presented in this section are specific to voice interactions and cover the following features (this list is not exhaustive):

- Answer a call;
- Transfer;
- Conference;
- Mute;
- Send DTMF;
- Listen reconnections and disconnections;
- Retrieve and hold calls.

Important

To create a new voice interaction, see the [Manage Voice Media](#).

Chains of Commands to Manage Voice Interactions

Default Commands	Parameters
Chain InteractionVoiceAlternateCall	
<ul style="list-style-type: none"> AlternateCall 	<ul style="list-style-type: none"> CommandParameter: IInteractionVoice - Genesyslab.Desktop.Modules.Core.Model.Interactions.IInteractionVoice Reasons: KeyValueCollection - Genesyslab.Enterprise.Commons.Collections.KeyValueCollection Extensions: KeyValueCollection
Chain InteractionVoiceAnswerCall	
<ul style="list-style-type: none"> AnswerCall 	<ul style="list-style-type: none"> CommandParameter: IInteractionVoice Reasons: KeyValueCollection Extensions: KeyValueCollection
Chain InteractionVoiceAutoCloseConsult	
<ul style="list-style-type: none"> AutoCloseConsult 	<ul style="list-style-type: none"> CommandParameter: IInteractionVoice
Chain InteractionVoiceBeforeClose	
<ul style="list-style-type: none"> DoNotCallOutboundChain 	<ul style="list-style-type: none"> CommandParameter: IInteraction
<ul style="list-style-type: none"> SetCallResultOutboundRecord 	<ul style="list-style-type: none"> CommandParameter: IInteraction
<ul style="list-style-type: none"> RescheduleOutboundRecord 	<ul style="list-style-type: none"> CommandParameter: IInteraction

Default Commands	Parameters
<ul style="list-style-type: none"> UpdateOutboundRecord 	<ul style="list-style-type: none"> CommandParameter: IInteraction
<ul style="list-style-type: none"> MarkProcessedOutboundChain 	<ul style="list-style-type: none"> CommandParameter: IInteraction
<ul style="list-style-type: none"> IsContactModified 	<ul style="list-style-type: none"> Caseld: String
<ul style="list-style-type: none"> Close 	<ul style="list-style-type: none"> CommandParameter: IInteractionVoice
Chain InteractionVoiceClose	
<ul style="list-style-type: none"> DoNotCallOutboundChain 	<ul style="list-style-type: none"> CommandParameter: IInteraction
<ul style="list-style-type: none"> SetCallResultOutboundRecord 	<ul style="list-style-type: none"> CommandParameter: IInteraction
<ul style="list-style-type: none"> RescheduleOutboundRecord 	<ul style="list-style-type: none"> CommandParameter: IInteraction
<ul style="list-style-type: none"> UpdateOutboundRecord 	<ul style="list-style-type: none"> CommandParameter: IInteraction
<ul style="list-style-type: none"> MarkProcessedOutboundChain 	<ul style="list-style-type: none"> CommandParameter: IInteraction
<ul style="list-style-type: none"> GetAttachedDataInformationUCS <p>Available since: 8.5</p>	<ul style="list-style-type: none"> CommandParameter: IInteractionsBundle - Genesyslab.Desktop.Modules.Core.Model.Interactions.IInteractionsBundle

Default Commands	Parameters
<ul style="list-style-type: none"> UpdateNotePad Available since: 8.5 	<ul style="list-style-type: none"> CommandParameter: IInteractionsBundle
<ul style="list-style-type: none"> StopInteractionVoiceUCS Available since: 8.5 	<ul style="list-style-type: none"> CommandParameter: IInteractionsBundle
<ul style="list-style-type: none"> IsContactModified 	<ul style="list-style-type: none"> Caseld: String
<ul style="list-style-type: none"> Close 	<ul style="list-style-type: none"> CommandParameter: IInteractionVoice
Chain InteractionVoiceCompleteConference	
<ul style="list-style-type: none"> CompleteNotePad Available since: 8.5 	<ul style="list-style-type: none"> CommandParameter: IInteraction UserData: KeyValueCollection
<ul style="list-style-type: none"> CompleteConference 	<ul style="list-style-type: none"> CommandParameter: IInteractionVoice Reasons: KeyValueCollection Extensions: KeyValueCollection
Chain InteractionVoiceCompleteTransfer	
<ul style="list-style-type: none"> CompleteNotePad Available since: 8.5 	<ul style="list-style-type: none"> CommandParameter: IInteraction UserData: KeyValueCollection

Default Commands	Parameters
<ul style="list-style-type: none"> ParkCallback 	<ul style="list-style-type: none"> CommandParameter: IInteraction
<ul style="list-style-type: none"> CompleteTransfer 	<ul style="list-style-type: none"> CommandParameter: IInteractionVoice Reasons: KeyValueCollection Extensions: KeyValueCollection
Chain InteractionVoiceDeleteFromConference	
<ul style="list-style-type: none"> DeleteFromConference 	<ul style="list-style-type: none"> CommandParameter: Party - Genesyslab.Desktop.Modules.Core.Model.Interactions.Party Reasons: KeyValueCollection Extensions: KeyValueCollection
Chain InteractionVoiceHoldCall	
<ul style="list-style-type: none"> HoldCall 	<ul style="list-style-type: none"> CommandParameter: IInteractionVoice Reasons: KeyValueCollection Extensions: KeyValueCollection
Chain InteractionVoiceIfPossibleCloseInteraction	
<ul style="list-style-type: none"> DoNotCallOutboundChain 	<ul style="list-style-type: none"> CommandParameter: IInteraction
<ul style="list-style-type: none"> SetCallResultOutboundRecord 	<ul style="list-style-type: none"> CommandParameter: IInteraction

Default Commands	Parameters
<ul style="list-style-type: none"> RescheduleOutboundRecord 	<ul style="list-style-type: none"> CommandParameter: IInteraction
<ul style="list-style-type: none"> UpdateOutboundRecord 	<ul style="list-style-type: none"> CommandParameter: IInteraction
<ul style="list-style-type: none"> MarkProcessedOutboundChain 	<ul style="list-style-type: none"> CommandParameter: IInteraction
<ul style="list-style-type: none"> GetAttachedDataInformationUCS 	<ul style="list-style-type: none"> CommandParameter: IInteractionsBundle - Genesyslab.Desktop.Modules.Core.Model.Interactions.IInteractionsBundle
<ul style="list-style-type: none"> UpdateNotePad 	<ul style="list-style-type: none"> CommandParameter: IInteractionsBundle
<ul style="list-style-type: none"> IsContactModified CasId: System.String 	
<ul style="list-style-type: none"> IsPossibleToMarkDone 	<ul style="list-style-type: none"> CommandParameter: IInteractionVoice
<ul style="list-style-type: none"> CompleteDispositionCode 	<ul style="list-style-type: none"> CommandParameter: IInteraction - Genesyslab.Desktop.Modules.Core.Model.Interactions.IInteraction
<ul style="list-style-type: none"> Close 	<ul style="list-style-type: none"> CommandParameter: IInteractionVoice
<ul style="list-style-type: none"> StopInteractionVoiceUCS 	<ul style="list-style-type: none"> CommandParameter: IInteractionsBundle

Default Commands	Parameters
Chain InteractionVoiceInitConference	
<ul style="list-style-type: none"> ConferenceNotepad 	<ul style="list-style-type: none"> CommandParameter: IInteraction UserData: KeyValueCollection
<ul style="list-style-type: none"> RequestSaveNotepad 	<ul style="list-style-type: none"> CommandParameter: IInteraction NotepadViewModel: INotepadViewModel - Genesyslab.Desktop.Modules.Windows.Interactions.INotepadViewModel
<ul style="list-style-type: none"> InitConference 	<ul style="list-style-type: none"> CommandParameter: IInteractionVoice Destination: - System.String Location: String UserData: KeyValueCollection Reasons: KeyValueCollection Extensions: KeyValueCollection
<ul style="list-style-type: none"> AddRecentCall 	<ul style="list-style-type: none"> RecentIndex: ObjectTC
Chain InteractionVoiceInitTransfer	
<ul style="list-style-type: none"> TransferNotepadInitiateNotepad Modified in: 8.5 	<ul style="list-style-type: none"> CommandParameter: IInteraction UserData: KeyValueCollection
<ul style="list-style-type: none"> RequestSaveNotepad 	<ul style="list-style-type: none"> CommandParameter: IInteraction

Default Commands	Parameters
	<ul style="list-style-type: none"> • NotepadViewModel: INotepadViewModel
<ul style="list-style-type: none"> • InitTransfer 	<ul style="list-style-type: none"> • CommandParameter: IInteractionVoice • Destination: String • Location: String • UserData: KeyValueCollection • Reasons: KeyValueCollection • Extensions: KeyValueCollection • CorrelatorData: String
<ul style="list-style-type: none"> • AddRecentCall 	<ul style="list-style-type: none"> • RecentIndex: ObjectTC - Genesyslab.Desktop.Modules.TeamCommunicator.ObjectTC
Chain InteractionVoicesPossibleToClose	
<ul style="list-style-type: none"> • IsContactModified 	<ul style="list-style-type: none"> • CaseId: String
<ul style="list-style-type: none"> • IsPossibleToMarkDone 	<ul style="list-style-type: none"> • CommandParameter: IInteractionVoice
<ul style="list-style-type: none"> • CompleteDispositionCode 	<ul style="list-style-type: none"> • CommandParameter: IInteraction
Chain InteractionVoiceListenDisconnect	
<ul style="list-style-type: none"> • ListenDisconnect 	<ul style="list-style-type: none"> • CommandParameter: Party

Default Commands	Parameters
	<ul style="list-style-type: none"> • Reasons: KeyValueCollection • Extensions: KeyValueCollection
Chain InteractionVoiceListenReconnect	
<ul style="list-style-type: none"> • ListenReconnect 	<ul style="list-style-type: none"> • CommandParameter: Party • Reasons: KeyValueCollection • Extensions: KeyValueCollection
Chain InteractionVoicePauseRecordCall	
<ul style="list-style-type: none"> • PauseRecordCall 	<ul style="list-style-type: none"> • CommandParameter: IInteractionVoice
Chain InteractionVoiceReconnectCall	
<ul style="list-style-type: none"> • PromptEnd 	<ul style="list-style-type: none"> • CommandParameter: IInteraction • ConfirmParentWindow: Window
<ul style="list-style-type: none"> • ReconnectCall 	<ul style="list-style-type: none"> • CommandParameter: IInteractionVoice • Reasons: KeyValueCollection • Extensions: KeyValueCollection
Chain InteractionVoiceRecordCall	
<ul style="list-style-type: none"> • RecordCall 	<ul style="list-style-type: none"> • CommandParameter: IInteractionVoice
Chain InteractionVoiceRedirectCall	

Default Commands	Parameters
<ul style="list-style-type: none"> RedirectCall 	<ul style="list-style-type: none"> CommandParameter: IInteractionVoice OtherDn: String OtherDnLocation: String Reasons: KeyValueCollection Extensions: KeyValueCollection
Chain InteractionVoiceReleaseCall	
<ul style="list-style-type: none"> SetSupervisorSilentCommand 	
<ul style="list-style-type: none"> PromptEnd 	<ul style="list-style-type: none"> CommandParameter: IInteraction ConfirmParentWindow: Window
<ul style="list-style-type: none"> ReleaseCall 	<ul style="list-style-type: none"> CommandParameter: IInteractionVoice Reasons: KeyValueCollection Extensions: KeyValueCollection
Chain InteractionVoiceReportBadQualityOfMedia	
<ul style="list-style-type: none"> ReportBadQualityOfMedia 	<ul style="list-style-type: none"> CommandParameter: IInteractionVoice
Chain InteractionVoiceResume	
<ul style="list-style-type: none"> Resume 	<ul style="list-style-type: none"> CommandParameter: IInteractionVoice Reasons: KeyValueCollection

Default Commands	Parameters
	<ul style="list-style-type: none"> • Extensions: KeyValueCollection
Chain InteractionVoiceResumeRecordCall	
<ul style="list-style-type: none"> • ResumeRecordCall 	<ul style="list-style-type: none"> • CommandParameter: IInteractionVoice
Chain InteractionVoiceRetrieveCall	
<ul style="list-style-type: none"> • RetrieveCall 	<ul style="list-style-type: none"> • CommandParameter: IInteractionVoice • Reasons: KeyValueCollection • Extensions: KeyValueCollection
Chain InteractionVoiceSendDTMF	
<ul style="list-style-type: none"> • SendDTMF 	<ul style="list-style-type: none"> • CommandParameter: IInteractionVoice • DtmfDigits: String • Reasons: KeyValueCollection • Extensions: KeyValueCollection
Chain InteractionVoiceSingleStepConference	
<ul style="list-style-type: none"> • ConferenceSingleStepNotepad 	<ul style="list-style-type: none"> • CommandParameter: IInteraction • UserData: KeyValueCollection
<ul style="list-style-type: none"> • RequestSaveNotepad 	<ul style="list-style-type: none"> • CommandParameter: IInteraction • NotepadViewModel: INotepadViewModel

Default Commands	Parameters
<ul style="list-style-type: none"> SingleStepConference 	<ul style="list-style-type: none"> CommandParameter: IInteractionVoice Destination: String Location: String UserData: KeyValueCollection Reasons: KeyValueCollection Extensions: KeyValueCollection
<ul style="list-style-type: none"> AddRecentCall 	<ul style="list-style-type: none"> RecentIndex: ObjectTC
Chain InteractionVoiceSingleStepTransfer	
<ul style="list-style-type: none"> ParkCallback 	<ul style="list-style-type: none"> CommandParameter: IInteraction
<ul style="list-style-type: none"> TransferSingleStepNotepad 	<ul style="list-style-type: none"> CommandParameter: IInteraction UserData: KeyValueCollection
<ul style="list-style-type: none"> RequestSaveNotepad 	<ul style="list-style-type: none"> CommandParameter: IInteraction NotepadViewModel: INotepadViewModel
<ul style="list-style-type: none"> SingleStepTransfer 	<ul style="list-style-type: none"> CommandParameter: IInteractionVoice Destination: String Location: String

Default Commands	Parameters
	<ul style="list-style-type: none"> • UserData: KeyValueCollection • Reasons: KeyValueCollection • Extensions: KeyValueCollection
<ul style="list-style-type: none"> • AddRecentCall 	<ul style="list-style-type: none"> • RecentIndex: ObjectTC
Chain InteractionVoiceStopRecordCall	
<ul style="list-style-type: none"> • StopRecordCall 	<ul style="list-style-type: none"> • CommandParameter: IInteractionVoice
Chain UpFrontPauseRecordingVoiceCallCommand Added in 8.5	
<ul style="list-style-type: none"> • Pause 	<ul style="list-style-type: none"> • CommandParameter: IInteractionVoice
Chain UpFrontPauseRecordingVoiceCallCommandImplicit Added in 8.5	
<ul style="list-style-type: none"> • Pause 	<ul style="list-style-type: none"> • CommandParameter: IInteractionVoice
Chain UpFrontResumeRecordingVoiceCallCommand Added in 8.5	
<ul style="list-style-type: none"> • Resume 	<ul style="list-style-type: none"> • CommandParameter: IInteractionVoice
Chain UpFrontResumeRecordingVoiceCallCommandImplicit Added in 8.5	

Default Commands	Parameters
<ul style="list-style-type: none"> Resume 	<ul style="list-style-type: none"> CommandParameter: IInteractionVoice
Commands common to all interactions	
Chain InteractionSetAttachedDataWithDelay	
SetAttachedData	<ul style="list-style-type: none"> CommandParameter: IInteraction AttachedData: KeyValueCollection
Chain BundleClose	
GetAttachedDataInformationUCS	CommandParameter: IInteractionsBundle - Genesyslab.Desktop.Modules.Core.Model.Interactions.IInteractionsBundle
UpdateNotePadForVoice	<ul style="list-style-type: none"> CommandParameter: IInteractionsBundle
ResetInteractionChatConsultation	CommandParameter: IInteractionsBundle
IsContactModified	<ul style="list-style-type: none"> Caseld: String
IsPossibleToClose	CommandParameter: IInteractionsBundle
CompleteDispositionCodeOnBundle	CommandParameter: IInteractionsBundle
Close	CommandParameter: IInteractionsBundle

Default Commands	Parameters
StopInteractionVoiceUCS	<ul style="list-style-type: none"><li data-bbox="1144 320 1666 347">• CommandParameter: IInteractionsBundle
GetOutboundPreviewRecord	CommandParameter: ICampaign - Genesyslab.Enterprise.Model.Interaction.ICampaign

Windows



Purpose: To provide information about customizable commands for the windows.

Learn about the Windows

The following commands apply to GUI components that your application can customize. For further information, see [Views and Regions](#), which lists the type of available components.

Important

Read [Use Customizable Commands](#) to see code snippets which demonstrate the usage of the commands.

Close Windows

The following commands are used to close windows.

Chains of Commands to Close Windows

Default Commands	Parameters
Chain WindowClose	
<ul style="list-style-type: none"> • IsPossibleToClose 	<ul style="list-style-type: none"> • CommandParameter: IInteractionsWindow - Genesyslab.Desktop.Modules.Windows.Interactions.IInteractionsWindow
<ul style="list-style-type: none"> • Close 	<ul style="list-style-type: none"> • CommandParameter: IInteractionsWindow
Chain WindowInteractionClose	
<ul style="list-style-type: none"> • IsPossibleToClose 	<ul style="list-style-type: none"> • CommandParameter: IInteractionsWindow
<ul style="list-style-type: none"> • Close 	<ul style="list-style-type: none"> • CommandParameter: IInteractionsWindow

Open and Restore Interaction Windows

The following commands are used to screen-pop interaction windows.

Chains of Commands to Open and Restore Interaction Windows

Default Commands	Parameters
Chain RestoreInteractionsWindow	
<ul style="list-style-type: none"> RestoreInteractionsWindow 	<ul style="list-style-type: none"> CommandParameter: IInteractionsWindow - Genesyslab.Desktop.Modules.Windows.Interactions.IInteractionsWindow Interaction: IInteraction - Genesyslab.Desktop.Modules.Core.Model.Interactions.IInteraction Case: ICase - Genesyslab.Desktop.Modules.Core.Model.Interactions.ICase
Chain ShowInteractionsWindow	
<ul style="list-style-type: none"> ShowInteractionsWindow 	<ul style="list-style-type: none"> CommandParameter: IInteractionsWindow Interaction: IInteraction Case: ICase

Manage Toaster

The following commands are available for the **Toaster Window**.

Chain of Commands to Manage Toaster

Default Commands	Parameters
Chain ShowToasterWindow	
<ul style="list-style-type: none"> ShowToasterWindow 	<ul style="list-style-type: none"> CommandParameter: IToasterWindow - Genesyslab.Desktop.Modules.Windows.Views.Toaster.IToasterWindow Interaction: IInteraction
Chain ToasterAcceptPreview	
<ul style="list-style-type: none"> Accept 	<ul style="list-style-type: none"> CommandParameter: IInteraction - Genesyslab.Desktop.Modules.Core.Model.Interactions.IInteraction
Chain ToasterDeclinePreview	
<ul style="list-style-type: none"> Decline 	<ul style="list-style-type: none"> CommandParameter: IInteraction
Chain ToasterInteractionAccept	
<ul style="list-style-type: none"> Accept 	<ul style="list-style-type: none"> CommandParameter: IInteraction
Chain ToasterInteractionDecline	
<ul style="list-style-type: none"> Decline 	<ul style="list-style-type: none"> CommandParameter: IInteraction OtherDn: String - System.String OtherDnLocation: String
<ul style="list-style-type: none"> RemoveOutboundChain 	<ul style="list-style-type: none"> CommandParameter: IInteraction
Chain ToasterInteractionEmailAccept	

Default Commands	Parameters
<ul style="list-style-type: none"> Accept 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia - Genesyslab.Desktop.Modules.OpenMedia.Model.Interactions.IInteractionOpenMedia Extensions: KeyValueCollection - Genesyslab.Enterprise.Commons.Collections.KeyValueCollection
Chain ToasterInteractionEmailDecline	
<ul style="list-style-type: none"> Decline 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia Reasons: KeyValueCollection Extensions: KeyValueCollection
<ul style="list-style-type: none"> Close 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia
Chain ToasterInteractionIMAccept	
<ul style="list-style-type: none"> Accept 	<ul style="list-style-type: none"> CommandParameter: IInteractionIM - Genesyslab.Desktop.Modules.Voice.Model.Interactions.IInteractionIM
Chain ToasterInteractionIMReject	
<ul style="list-style-type: none"> Reject 	<ul style="list-style-type: none"> CommandParameter: IInteractionIM<ref name="IInteractionIM" /> RejectReason: String Reasons: KeyValueCollection Extensions: KeyValueCollection
<ul style="list-style-type: none"> Reject 	<ul style="list-style-type: none"> CommandParameter: IInteraction

Default Commands	Parameters
Chain ToasterInteractionOpenMediaAccept	
<ul style="list-style-type: none"> Accept 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia Extensions: KeyValueCollection
Chain ToasterInteractionOpenMediaDecline	
<ul style="list-style-type: none"> Decline 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia Reasons: KeyValueCollection Extensions: KeyValueCollection
<ul style="list-style-type: none"> Close 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia
Chain ToasterInteractionPreviewAccept	
<ul style="list-style-type: none"> Accept 	<ul style="list-style-type: none"> CommandParameter: IInteraction
Chain ToasterInteractionPreviewDecline	
<ul style="list-style-type: none"> Decline 	<ul style="list-style-type: none"> CommandParameter: IInteraction
Chain ToasterShowBroadcastMessage	
<ul style="list-style-type: none"> ShowToaster 	<ul style="list-style-type: none"> CommandParameter: IBroadcastMessage

Broadcast Window

These commands enable you to manage a broadcast message displayed to the agents.

Chain of Commands for the Broadcast Window

Default Commands	Parameters
Chain BroadcastMessageClose	
<ul style="list-style-type: none"> Close 	<ul style="list-style-type: none"> CommandParameter: String - System.String
Chain BroadcastMessageShow	
<ul style="list-style-type: none"> Show 	<ul style="list-style-type: none"> CommandParameter: IBroadcastMessage - Genesyslab.Desktop.Modules.Core.Model.Broadcast.IBroadcastMessage Source: BroadcastMessageWindowSource - Genesyslab.Desktop.Modules.Windows.Views.Broadcast.BroadcastMessageWindowSource
Chain ShowMessageToasterWindow	
<ul style="list-style-type: none"> ShowMessageToasterWindow 	<ul style="list-style-type: none"> CommandParameter: IToasterWindow BroadcastMessage: IBroadcastMessage

Additional GUI Commands

Additionally, you can use the following commands:

Additional Chains of Commands

Default Commands	Parameters
Chain ApplicationClose	
<ul style="list-style-type: none"> • IsPossibleToClose 	
<ul style="list-style-type: none"> • CloseAllInteractions 	
<ul style="list-style-type: none"> • StopAllMonitoringCommand 	
<ul style="list-style-type: none"> • ChannelsLogOff 	
<ul style="list-style-type: none"> • NegotiatedLogout 	
<ul style="list-style-type: none"> • Close 	
Chain CancelConnectionCommand	
Chain NotepadInsertTimeStamp	
<ul style="list-style-type: none"> • InsertTimeStamp 	<ul style="list-style-type: none"> • CommandParameter: IInteraction • TextBoxBase: TextBoxBase - System.Windows.Controls.TextBoxBase
Chain PossibleToCloseAllInteractions	
<ul style="list-style-type: none"> • IsPossibleToClose 	
Chain TeamCommunicatorAddToFavorites	

Default Commands	Parameters
<ul style="list-style-type: none"> AddObjectTCToFavorite 	
Chain TeamCommunicatorModifyFavorite	
<ul style="list-style-type: none"> ModifyFavorite 	
Chain TeamCommunicatorOpenInteraction	
<ul style="list-style-type: none"> Open 	<ul style="list-style-type: none"> ActionTarget: ActionTarget - Genesyslab.Desktop.Modules.Windows.Interactions.ActionTarget
Chain ContactIsModified	
<ul style="list-style-type: none"> IsModified 	Caseld: String

Workbins

Purpose: To provide information about customizable commands.

Learn about Workbin Commands

These commands manage selections in workbins.

Important

Read [Use Customizable Commands](#) to see code snippets which demonstrate the usage of the commands.

Managing Workbin Selections

The following commands cover:

- Selections in workbins;
- Push and Pull interactions.

Chains of Commands to Manage Workbin Selections

Default Commands	Parameters
Chain InteractionFilterSelected	
<ul style="list-style-type: none"> InteractionFilterSelected 	<ul style="list-style-type: none"> WorkbinName: string - System.string WorkbinNameDisplay: string WorkbinGroupName: string WorkbinGroupNameDisplay: string
Chain InteractionPossibleToMoveToWorkbin	
<ul style="list-style-type: none"> IsContactModified 	<ul style="list-style-type: none"> CasId: String - System.String
<ul style="list-style-type: none"> IsWorkbinDestinationDefined 	<ul style="list-style-type: none"> WorkbinId: String WorkbinOptionName: String CommandParameter: IInteractionOpenMedia - Genesyslab.Desktop.Modules.OpenMedia.Model.Interaction.IInteractionOpenMedia IgnoreInteractionFromWorkbinId: String Added in 8.5.0
Chain InteractionManagementMarkDone	
<ul style="list-style-type: none"> MediaTypeMarkDone 	<ul style="list-style-type: none"> mediaType: String
Chain InteractionManagementMoveToQueueById	
<ul style="list-style-type: none"> Confirm 	<ul style="list-style-type: none"> ActionToConfirm: String

Default Commands	Parameters
	<ul style="list-style-type: none"> ConfirmParentWindow: Window - System.Window.Window
<ul style="list-style-type: none"> PlaceInQueue 	<ul style="list-style-type: none"> CommandParameter: String Destination: String
<ul style="list-style-type: none"> AddRecentCall 	<ul style="list-style-type: none"> RecentIndex: ObjectTC - Genesyslab.Desktop.Modules.TeamCommunicator.ObjectTC
Chain InteractionManagementMoveToWorkbinById	
<ul style="list-style-type: none"> Confirm 	<ul style="list-style-type: none"> ActionToConfirm: String ConfirmParentWindow: Window
<ul style="list-style-type: none"> MoveToWorkbin 	<ul style="list-style-type: none"> CommandParameter: String DestinationId: String DestinationParameter: String
<ul style="list-style-type: none"> AddRecentCall 	<ul style="list-style-type: none"> RecentIndex: ObjectTC
Chain InteractionManagementSetAttachedData	
<ul style="list-style-type: none"> TeamLeadSetAttachedData 	<ul style="list-style-type: none"> CommandParameter: String
Chain MultipleInteractionsActionFromWorkbinById	

Default Commands	Parameters
<ul style="list-style-type: none"> Confirm 	<ul style="list-style-type: none"> ConfirmMessageText: String ConfirmParentWindow: Window
<ul style="list-style-type: none"> MultipleInteractionsCommand 	<ul style="list-style-type: none"> nbInteractions: Int32 - System.Int32 interactionId + i: String mediaType + i: String BasicActionNameBasicActionNameDicold: String Renamed in 8.5 SingleActionChainOfCommandName: String
Chain SetAttachedDataAndPutBack	
<ul style="list-style-type: none"> EsdkSetAttachedData 	<ul style="list-style-type: none"> CommandParameter: IOpenMediaInteraction - Genesyslab.Enterprise.Model.Interaction.IOpenMediaInteraction UserData: KeyValueCollection - Genesyslab.Enterprise.Commons.Collections.KeyValueCollection
<ul style="list-style-type: none"> PutBack 	
<ul style="list-style-type: none"> UpdateCaseDataInteractionsFilter 	<ul style="list-style-type: none"> CommandParameter: IOpenMediaInteraction UserData: KeyValueCollection ParentView: IWorkbinsView - Genesyslab.Desktop.Modules.OpenMedia.View.WorkbinsView.IWorkbinsView
Chain WorkbinPreSelectedByName	

Default Commands	Parameters
<ul style="list-style-type: none">ShowSelectedDetail	
Chain WorkbinPullInteraction	
<ul style="list-style-type: none">ShowSelectedDetail	
Chain WorkbinPushInteraction	
<ul style="list-style-type: none">ShowSelectedDetail	
Chain WorkbinSelected	
<ul style="list-style-type: none">NewWorkbinSelected	<ul style="list-style-type: none">WorkbinName: string - System.stringWorkbinNameDisplay: stringWorkbinGroupName: stringWorkbinGroupNameDisplay: string

Work Items

Purpose: To provide information about customizable commands.

Learn about Work Items

Work items are extended interactions that your application can configure through the Configuration Manager. Basically, they implement the open media interactions. The commands presented in this page are restricted to the work items. In addition to the work-items commands, you can use:

- [Open Media](#) to place interactions in a given queue, move them to workbins, and so on.

Important

Read [Use Customizable Commands](#) to see code snippets which demonstrate the usage of the commands.

Managing Work Items

The commands presented in this section are specific to work item interactions and cover the following features (this list is not exhaustive):

- Accept or decline a given work item;
- Pull and place in workbins;
- Close or stop the interactions.

Chains of Commands to Manage Workitems

Default Commands	Parameters
Chain InteractionWorkItemAccept	
<ul style="list-style-type: none"> Accept 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia - Genesyslab.Desktop.Modules.OpenMedia.Model.Interactions.IInteractionOpenMedia Extensions: KeyValueCollection
Chain InteractionWorkItemDecline	
<ul style="list-style-type: none"> Decline 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia Reasons: KeyValueCollection - Genesyslab.Enterprise.Commons.Collections.KeyValueCollection Extensions: KeyValueCollection
<ul style="list-style-type: none"> Close 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia
Chain InteractionWorkitemMarkDoneById	
<ul style="list-style-type: none"> GetEsdkInteraction Available since: 8.5 	<ul style="list-style-type: none"> CommandParameter: String
<ul style="list-style-type: none"> IsDispositionMissing Available since: 8.5 	<ul style="list-style-type: none"> EsdkOpenMediaInteraction: IOpenMediaInteraction - Genesys.Enterprise.Model.Interaction.IOpenMediaInteraction
<ul style="list-style-type: none"> ConfirmWorkitem 	<ul style="list-style-type: none"> ActionToConfirm: String - System.String ConfirmParentWindow: Window - System.Window.Window

Default Commands	Parameters
<ul style="list-style-type: none"> MarkDoneldWorkitem 	<ul style="list-style-type: none"> CommandParameter: String WorkbinId: String
Chain InteractionWorkItemMoveToWorkbin	
<ul style="list-style-type: none"> IsContactModified 	<ul style="list-style-type: none"> Caseld: String
<ul style="list-style-type: none"> IsWorkbinDestinationDefined 	<ul style="list-style-type: none"> WorkbinId: String WorkbinOptionName: String CommandParameter: IInteractionOpenMedia IgnoreInteractionFromWorkbinId: String Added in 8.5
<ul style="list-style-type: none"> SetAttachedDadaInformation 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia
<ul style="list-style-type: none"> UpdateInteractionUCS 	<ul style="list-style-type: none"> CommandParameter: IInteraction - Genesyslab.Desktop.Modules.Core.Model.Interactions.IInteraction
<ul style="list-style-type: none"> MoveToWorkbin 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia WorkbinId: String WorkbinParameter: String WorkbinType: WorkbinType Deprecated in 8.5


Default Commands	Parameters
<ul style="list-style-type: none"> UpdateContactInformation 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia
<p>Chain InteractionWorkItemPossibleToMoveToWorkbin</p>	
<ul style="list-style-type: none"> SetWorkbinDefinition 	<ul style="list-style-type: none"> WorkbinId: String output parameter - System.String output parameter WorkbinOptionName: String, output parameter - System.String, output parameter CommandParameter: IInteractionOpenMedia
<ul style="list-style-type: none"> IsWorkbinDestinationDefined 	<ul style="list-style-type: none"> WorkbinId: String WorkbinOptionName: String CommandParameter: IInteractionOpenMedia IgnoreInteractionFromWorkbinId: String Added in 8.5
<p>Chain InteractionWorkitemPutBackToOriginalSource</p>	
<ul style="list-style-type: none"> IsContactModified 	<ul style="list-style-type: none"> Caseld: String
<ul style="list-style-type: none"> PutBackWorkitem 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia Destination: String Reason: KeyValueCollection Extensions: KeyValueCollection

Default Commands	Parameters
<ul style="list-style-type: none"> Close 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia
<ul style="list-style-type: none"> UpdateContactInformation 	<ul style="list-style-type: none"> CommandParameter: IInteractionOpenMedia
Chain InteractionWorkItemSearchAndMoveToWorkbin	
<ul style="list-style-type: none"> IsContactModified 	<ul style="list-style-type: none"> CasId: String
<ul style="list-style-type: none"> SearchAndMoveToWorkbin 	
Chain InteractionWorkitemWorkflowFromInteractionESDK	
<ul style="list-style-type: none"> ESDKWorkflow 	<ul style="list-style-type: none"> CommandParameter: IIOpenMediaInteraction - Genesyslab.Enterprise.Model.Interaction.IIOpenMediaInteraction Workflow: String Reasons: KeyValueCollection Extensions: KeyValueCollection AttachedDataInformation: IDispositionCode - Genesyslab.Desktop.Modules.Core.Model.Interactions.IDispositionCode DispositionCode: KeyValueCollection
Commands common to all interactions	
Chain InteractionSetAttachedDataWithDelay	
SetAttachedData	<ul style="list-style-type: none"> CommandParameter: IInteraction

Default Commands	Parameters
	<ul style="list-style-type: none"> AttachedData: KeyValueCollection
Chain BundleClose	
GetAttachedDataInformationUCS	CommandParameter: IInteractionsBundle - Genesyslab.Desktop.Modules.Core.Model.Interactions.IInteractionsBundle
UpdateNotePadForVoice	<ul style="list-style-type: none"> CommandParameter: IInteractionsBundle
ResetInteractionChatConsultation	CommandParameter: IInteractionsBundle
IsContactModified	<ul style="list-style-type: none"> Caseld: String
IsPossibleToClose	CommandParameter: IInteractionsBundle
CompleteDispositionCodeOnBundle	CommandParameter: IInteractionsBundle
Close	CommandParameter: IInteractionsBundle
StopInteractionVoiceUCS	<ul style="list-style-type: none"> CommandParameter: IInteractionsBundle
GetOutboundPreviewRecord	CommandParameter: ICampaign - Genesyslab.Enterprise.Model.Interaction.ICampaign

References

Customize Views and Regions

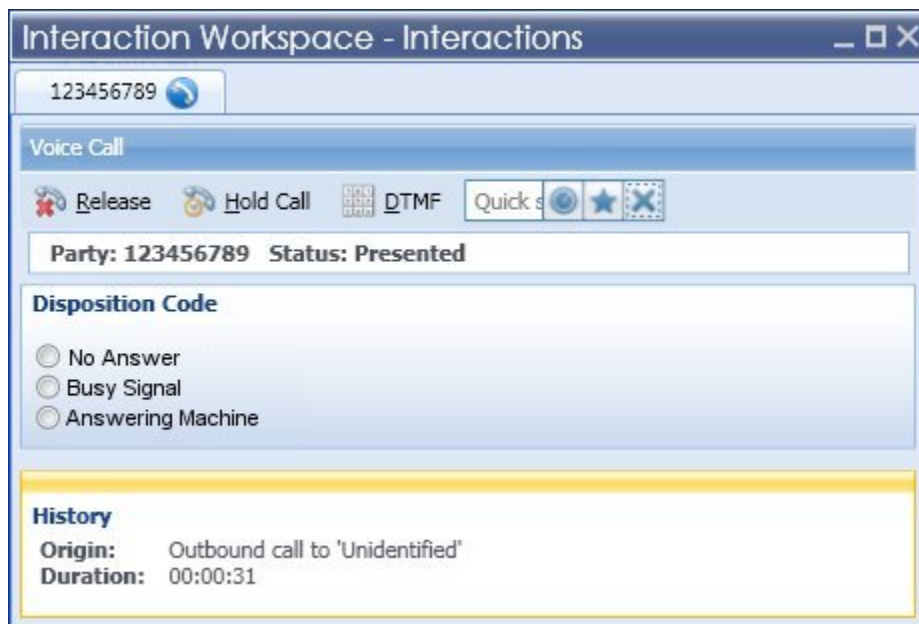
	<p>Purpose: To provide information about customizable views and their regions.</p>
---	---

Before You Start

- All the code snippets in this page are extracted from the Genesyslab.Desktop.Modules.ExtensionSample source files. See [About the Extension Samples](#) for additional information about the samples.
- In addition to this page, read:
 - [Creating a New Module](#)
 - [Deploying Your Custom Module into the Genesys Out-Of-The-Box Application](#)

Replacing an Existing View

There are several ways to customize Workspace Desktop Edition. The most basic way is to change an existing behavior or appearance by changing the implementation of an existing interface. The code that is displayed after the figure demonstrates how to replace an existing view, `DispositionCodeView`, with the new view, `DispositionCodeExView`. You can replace the existing view with another by associating the existing `IDispositionCodeView` interface with the new `DispositionCodeExView` implementation.



Voice Interactions View Before Customization. The out-of-the-box application uses radio buttons in the Disposition Code View. The code sample that is displayed after the figure modifies this view.

1. Start by creating a new Windows Presentation Foundation (WPF) UserControl that is composed of the following two files:
 - DispositionCodeExView.xaml
 - DispositionCodeExView.xaml.cs
2. Ensure that your custom-built view named, DispositionCodeExView implements the genuine IDispositionCodeView interface:

[C#]

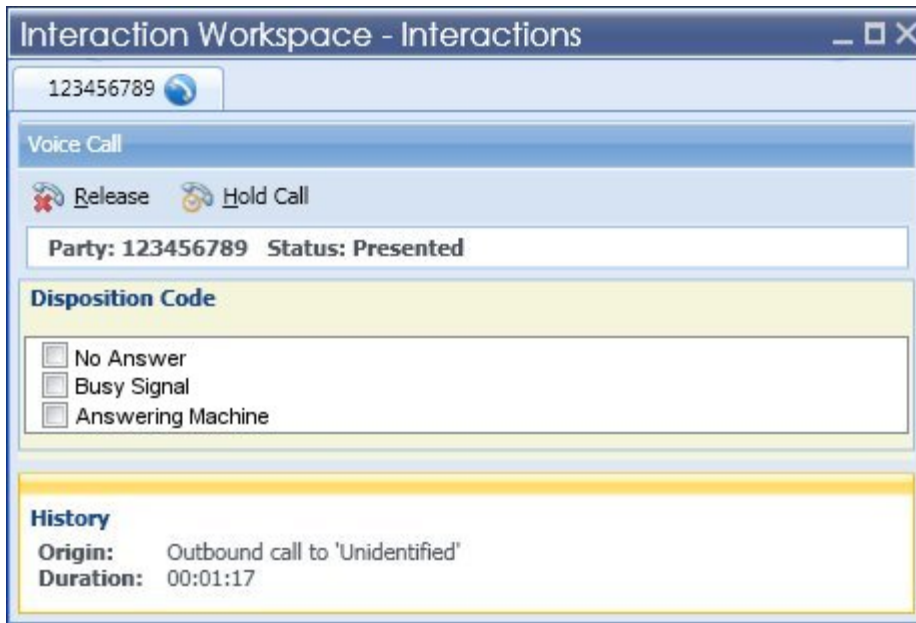
```
// File: DispositionCodeExView.cs
public partial class DispositionCodeExView : UserControl, IDispositionCodeView
{
    public DispositionCodeExView(IDispositionCodeViewModel dispositionCodeViewModel)
    {
        this.viewManager = viewManager;
        this.Model = dispositionCodePresentationModel;
        InitializeComponent();
        Width = Double.NaN;
        Height = Double.NaN;
    }
    #region IDispositionCodeView Members
    public IDispositionCodeViewModel Model
    {
        get { return this.DataContext as IDispositionCodeViewModel; }
        set { this.DataContext = value; }
    }
    #endregion
    ...
}
```

3. Register the new view in your module to make it replace the former view when the module is loaded. Do this by calling the `IObjectContainer.RegisterType` method must be used to register the new implementation in the initialization method of the `ExtensionSampleModule`:

[C#]

```
// File: ExtensionSampleModule.cs
public class ExtensionSampleModule : IModule
{
    readonly IObjectContainer container;
    ...
    public void Initialize()
    {
        container.RegisterType<IDispositionCodeView, DispositionCodeExView>();
    }
}
```

You can replace any view with your own custom-built view by using the preceding examples. The figure below shows the view for Workspace Desktop Edition Voice Interactions after customization. In the customized view, the radio buttons for disposition codes are replaced with check boxes.



Disposition Code View After Customization

Important

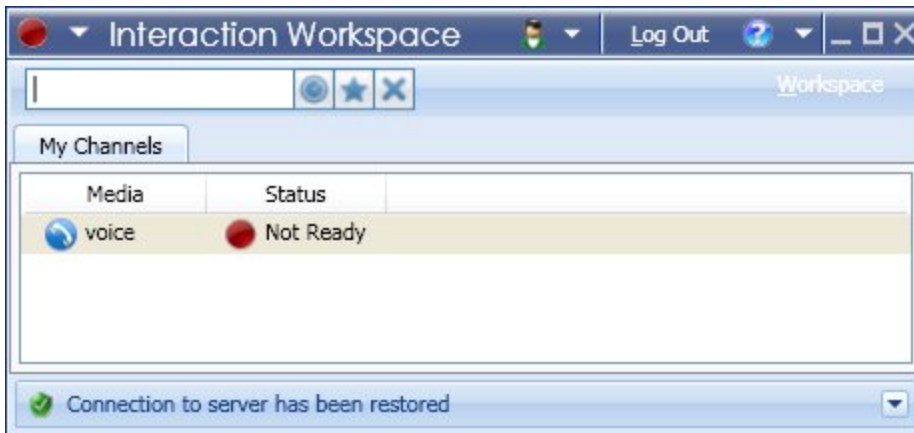
Although the application has a different appearance, it retains its previous behavior.

Creating a New View

Advanced customization provides the `IViewManager` interface to add a new view to an existing region (which is embedded in an existing view). Regions which embed multiple views, tabs, or buttons, can be enriched with new views. To use the Model-View-ViewModel (MVVM) pattern, you must create both the view (for instance, `MySampleView`) which extends the `IView` interface and the view-model, for instance `MySampleViewModel`. The following subsections detail the creation for two new views through the customization samples.

Adding a Tab to the `ToolBarWorkplaceRegion`

The `Genesyslab.Desktop.Modules.ExtensionSample` example creates a new view in the Voice Interaction panel. In the following figure, the out-of-the-box application has a single tab called `My Channels`, which is part of the `ToolBarWorkplaceRegion` region. The customization adds a new tab called `My Sample Header` which contains a button and a time counter.



Voice Interactions View before customization. A single tab 'My Channels' is available in the `ToolBarWorkspaceRegion`

The following steps details the content of the `Genesyslab.Desktop.Modules.ExtensionSample`.

1. To create the view-model, create a new interface named `IMySampleViewModel` which manages a time counter and the header label for the new tab:

[C#]

```
// File: IMySamplePresentationModel.cs
public interface IMySampleViewModel
{
    string Header { get; set; }
    TimeSpan Counter { get; set; }
    void ResetCounter();
}
```

2. Implement this interface by creating the `MySampleViewModel` class, as follows:

[C#]

```
// File: MySamplePresentationModel.cs
public class MySampleViewModel : IMySampleViewModel, INotifyPropertyChanged
{
    // Field variables
    string header = "My Sample Header";
    TimeSpan counter = TimeSpan.Zero;
    public MySampleViewModel()
    {
        // Start the counter timer
        DispatcherTimer dispatcherTimer = new DispatcherTimer();
        dispatcherTimer.Interval = new TimeSpan(0, 0, 1);
        dispatcherTimer.Tick += new EventHandler(delegate(object sender, EventArgs e)
        {
            Counter += TimeSpan.FromSeconds(1.0);
        });
        dispatcherTimer.Start();
    }
    #region IMySamplePresentationModel Members
    public string Header
    {
        get { return header; }
        set { if (header != value) { header = value; OnPropertyChanged("Header"); } }
    }
    public TimeSpan Counter
    {
```

```

    get { return counter; }
    set { if (counter != value) { counter = value; OnPropertyChanged("Counter"); } }
}
public void ResetCounter()
{
    Counter = TimeSpan.Zero;
}
#endregion
#region INotifyPropertyChanged Members
public event PropertyChangedEventHandler PropertyChanged;
protected void OnPropertyChanged(string propertyName)
{
    if (PropertyChanged != null)
        PropertyChanged(this, new PropertyChangedEventArgs(propertyName));
}
#endregion
}

```

3. Create the the view interface you want to implement, named `IMySampleView`, for your built-in component:

[C#]

```

// File: IMySampleView.cs
public interface IMySampleView : IView
{
    IMySampleViewModel Model { get; set; }
}

```

4. Create a content for the new view with a new WPF `UserControl` that is composed of the following two files:

- `MySampleView.xaml`

[XAML]

```

<UserControl x:Class="Genesyslab.Desktop.Modules.ExtensionSample.MySample.MySampleView"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    Height="220" Width="279" MinHeight="90">
    <Grid>
        <Ellipse Margin="12" Name="ellipse1" Stroke="Black" />
        <StackPanel HorizontalAlignment="Center" VerticalAlignment="Center">
            <Button Click="Button_Click">Button</Button>
            <TextBlock Text="{Binding Counter}" />
        </StackPanel>
    </Grid>
</UserControl>

```

- `MySampleView.xaml.cs` which contains your custom-built class named `MySampleView` implementing the `IMySampleView` interface created previously:

[C#]

```

// File: MySampleView.xaml.cs
public partial class MySampleView : UserControl, IMySampleView
{
    public MySampleView(IMySampleViewModel mySampleViewModel)
    {
        this.Model = mySampleViewModel;
        InitializeComponent();
        Width = Double.NaN;
        Height = Double.NaN;
    }
}

```

```

#region IMySampleView Members

public IMySampleViewModel Model
{
    get { return this.DataContext as IMySampleViewModel; }
    set { this.DataContext = value; }
}
#endregion
#region IView Members
public object Context { get; set; }
public void Create()
{
}
public void Destroy()
{
}
#endregion
private void Button_Click(object sender, System.Windows.RoutedEventArgs e)
{
    Model.ResetCounter();
}
}

```

5. In the Initialize() method of your module (see [Creating a New Module](#)):

- Register your new view and model with the `IObjectContainer.RegisterType` method.
- Insert the view in the appropriate view or region.

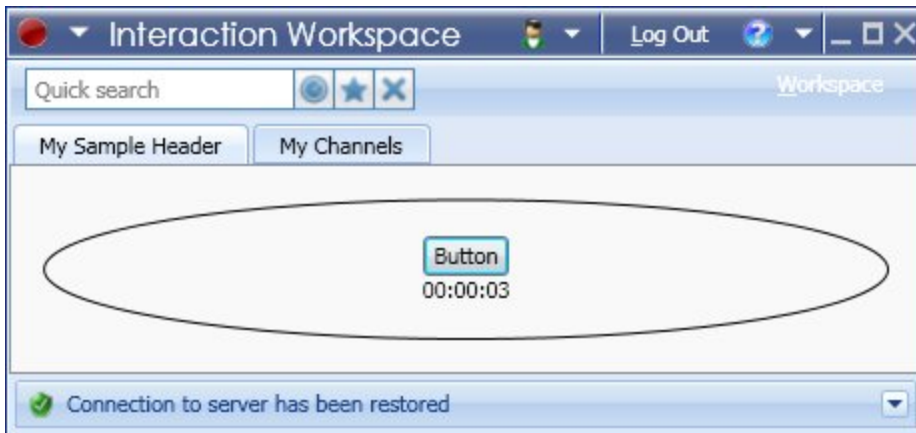
[C#]

```

// File: ExtensionSampleModule.cs
public class ExtensionSampleModule : IModule
{
    public void Initialize()
    {
        container.RegisterType<IMySampleView, MySampleView>();
        container.RegisterType<IMySampleViewModel, MySampleViewModel>();
        // Add the MySample view to the region "ToolbarWorkplaceRegion" (The TabControl in
the main toolbar)
        viewManager.ViewsByRegionName["ToolbarWorkplaceRegion"].Insert(0,
            new ViewActivator() { ViewType = typeof(IMySampleView), ViewName =
"MySample" });
        ...
    }
}

```

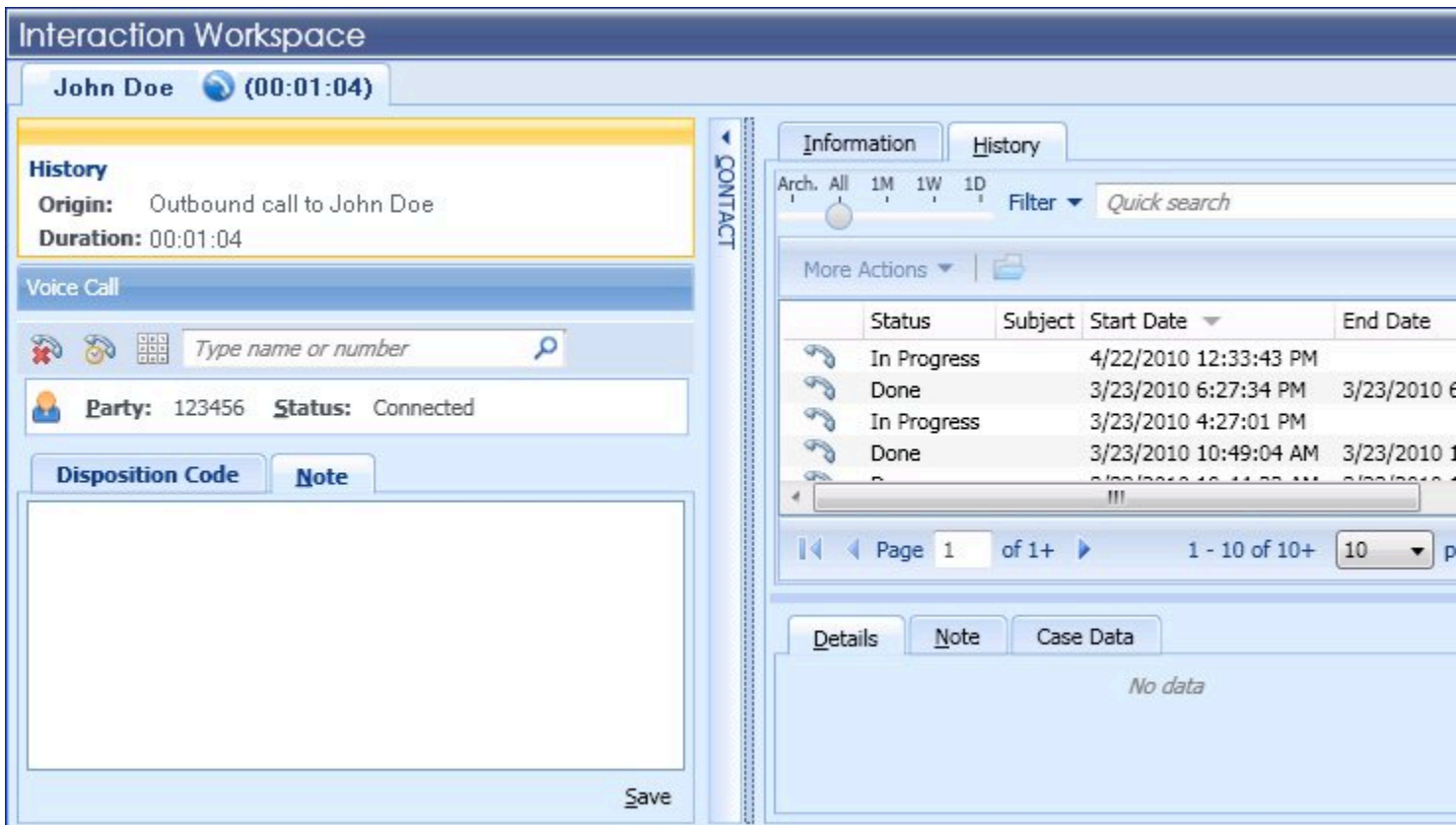
The figure below depicts the `MySampleView` after customization with the second tab (`My Sample Header`) included in the view. In the following example, the `ToolbarWorkplaceRegion` of the view is modified. For a complete list of views and regions, see [How to Customize Views and Their Regions Reference for Windows](#).



Voice Interactions View After Customization: A new tab 'MySampleHeader' is available.

Adding a View to the Interaction Window

Similar to the Genesyslab.Desktop.Modules.ExtensionSample is the Genesyslab.Desktop.Modules.InteractionExtensionSample which adds a view to the right panel of the Interaction Window.



The Interaction Window before customization. In the middle bar which separates the windows in two parts, a "CONTACT" expand button displays the current contact view to the right side of the window.

1. Create a new interface named `IMySampleViewModel`.

The view model implemented includes a case for the interaction management, in addition to the counter and the header.

[C#]

```
// File: IMySamplePresentationModel.cs
namespace Genesyslab.Desktop.Modules.InteractionExtensionSample.MySample
{
    public interface IMySampleViewModel
    {
        string Header { get; set; }
        TimeSpan Counter { get; set; }
        ICase Case { get; set; }
        void ResetCounter();
    }
}
```

2. the implementation of the interface includes the management of the case.

[C#]

```
// File: MySamplePresentationModel.cs
namespace Genesyslab.Desktop.Modules.InteractionExtensionSample.MySample
{
    public class MySampleViewModel : IMySampleViewModel, INotifyPropertyChanged
    {
        // Field variables
        string header = "My Sample Header";
        TimeSpan counter = TimeSpan.Zero;
        ICase @case;
        public MySampleViewModel()
        {
            // Start the counter timer
            DispatcherTimer dispatcherTimer = new DispatcherTimer();
            dispatcherTimer.Interval = new TimeSpan(0, 0, 1);
            dispatcherTimer.Tick += new EventHandler(delegate(object sender, EventArgs e)
            {
                Counter += TimeSpan.FromSeconds(1.0);
            });
            dispatcherTimer.Start();
        }
        #region IMySamplePresentationModel Members
        ...
        public ICase Case
        {
            get { return @case; }
            set { if (@case != value) { @case = value; OnPropertyChanged("Case"); } }
        }
        #endregion
        #region INotifyPropertyChanged Members
        ...
        #endregion
    }
}
```

3. Then, you create the view interfaces you want to implement, named `IMySampleView` and `IMySampleButtonView.cs`, for your built-in components:

[C#]

```
// File: IMySampleButtonView.cs
namespace Genesyslab.Desktop.Modules.InteractionExtensionSample.MySample
```

```

public interface IMySampleButtonView : IView
{
    IMySampleViewModel Model { get; set; }
}
// File: IMySampleView.cs
namespace Genesyslab.Desktop.Modules.InteractionExtensionSample.MySample
{
    // Interface matching the MySampleView view
    public interface IMySampleView : IView
    {
        // Gets or sets the model.
        IMySampleViewModel Model { get; set; }
    }
}

```

4. Create a content for the new view with a new WPF UserControl that is composed of the following two files:

- MySampleButtonView.xaml
- MySampleButtonView.xaml.cs

See the files in the InteractionExtensionSample.

5. In the Initialize() method of your module (see [Creating a New Module](#)):
 - Register your views and models with the `IObjectContainer.RegisterType` method.
 - Insert the views in the appropriate view or region, as shown here:

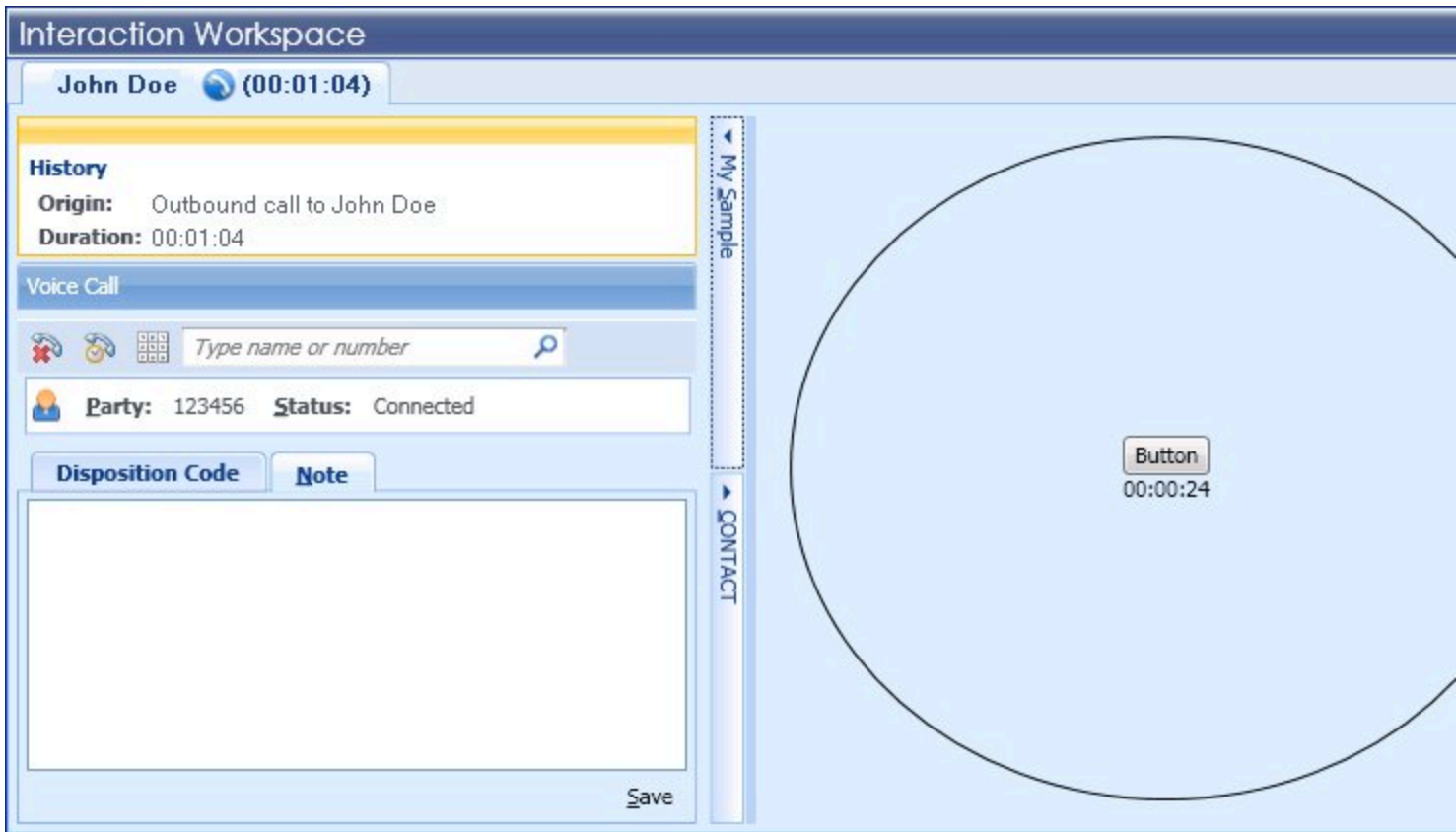
[C#]

```

// File: InteractionExtensionSampleModule.cs
public void Initialize()
{
    // Add a view in the right panel in the interaction window
    // Here we register the view (GUI) "IMySampleView" and its behavior counterpart
    "IMySampleViewModel"
    container.RegisterType<IMySampleView, MySampleView>();
    container.RegisterType<IMySampleViewModel, MySampleViewModel>();
    // Put the MySample view in the region "InteractionWorksheetRegion"
    viewManager.ViewsByRegionName["InteractionWorksheetRegion"].Add(
        new ViewActivator() { ViewType = typeof(IMySampleView),
            ViewName = "MyInteractionSample", ActivateView = true }
    );
    // Here we register the view (GUI) "IMySampleButtonView"
    container.RegisterType<IMySampleButtonView, MySampleButtonView>();
    // Put the MySampleMenuView view in the region "CaseViewSideButtonRegion"
    // (The case toggle button in the interaction windows)
    viewManager.ViewsByRegionName["CaseViewSideButtonRegion"].Add(
        new ViewActivator() { ViewType = typeof(IMySampleButtonView),
            ViewName = "MySampleButtonView", ActivateView = true }
    );
}

```

The figure below depicts the Interaction window after customization.



The Interaction Window after customization. In the middle bar which separates the windows in two parts, a "MySample" expand button displays the sample view.

Hiding and Showing Custom Views

Available since: 8.1.100.14

You can display a custom view according to specific parameters by using the concept of conditions. To do this, include a Condition when you register your view with the IViewManager. This condition will be executed each time that the application framework instantiates the region. For instance, if you wish to change the displayed tabs in the interaction window based on the context of the displayed interaction, then you can include a condition when adding your customized IMySampleView to the InteractionDetailsRegion region:

[C#]

```
viewManager.ViewsByRegionName["InteractionDetailsRegion"].Add(new ViewActivator() {
    ViewType = typeof(IMySampleView), ViewName = "MyInteractionSample", ActivateView =
true,
    Condition = MySampleViewModel.MySampleViewModelCondition});
```

Next, implement this condition somewhere in your code. In the following example, this method returns true to show the custom view or false to hide it:

[C#]

```
public static bool MySampleViewModelCondition(ref object context)
{
    IDictionary<string, object> contextDictionary = context as IDictionary<string,
object>;
    object caseView;
    contextDictionary.TryGetValue("CaseView", out caseView);
    object caseObject;
    contextDictionary.TryGetValue("Case", out caseObject);
    ICase @case = caseObject as ICase;
    if (@case != null)
    {
        if (@case.MainInteraction != null)
        {
            IInteraction i = @case.MainInteraction;
            return (i.HasBeenPresentedIn);
        }
    }
    return false;
}
```

Best Practices for Views

Purpose: To provide a set of recommendations that are required in order to implement a typical view within Workspace Desktop Edition.

Keyboard Navigation

TAB Key--Every control in a window has the ability to have focus. Use the TAB key to move from one control to the next, or use SHIFT+TAB to move the previous control. The TAB order is determined by the order in which the controls are defined in the Extensible Application Markup Language (XAML) page.

Access Keys--A labeled control can obtain focus by pressing the ALT key and then typing the control's associated letter (label). To add this functionality, include an underscore character (_) in the content of a control. See the following sample XAML file:

[XAML]

```
<Label Content="_AcctNumber" />
```

Focus can also be given to a specific GUI control by typing a single character. Use the WPF control AccessText (the counterpart of the TextBlock control) to modify your application for this functionality. For example, you can use the code in the following XAML sample to eliminate having to press the ALT key:

[XAML]

```
<AccessText Text="_AcctNumber" />
```

Shortcut Keys--Trigger a command by typing a key combination on the keyboard. For example, press CTRL+C to copy selected text. **Alarm Notification**--Workspace Desktop Edition can be configured to emit a sound when an unsolicited event occurs.

Branding

To replace trademark logos, icon images and text, you must create two files, a .module-config file and a Rebranding.xml file. The RebrandingTheme.xml file is similar to a language dictionary and enables you customize the appearance of your application. The .module-config file links to the RebrandingTheme.xml file. For example, you could make a Rebranding.module-config file with the following content:

[.module-config]

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<configuration>
<configSections>
<section name="themes" type="Genesyslab.Desktop.Infrastructure.Theming.ThemesSection, Genesyslab.Desktop.Infrastructure" />
</configSections>
<themes>
<theme name="Default">
<xmlDictionaries>
<xmlDictionary name="rebranding" path=".\\RebrandingTheme.xml"></xmlDictionary>
</xmlDictionaries>
</theme>
</themes>
</configuration>
```

The second file, which here is named `RebrandingTheme.xml`, is file where the new images for logos, Splash Screen, Copyrights, about window text, and so on, are defined:

[XML]

```
<?xml version="1.0" encoding="utf-8" ?>
<Dictionary>
<Value Id="Application.SplashScreen" Source="pack://application:,,,/Genesyslab.Desktop.WPFCommon;component/Images/Splash.png"/>
<!--Value Id="Application.SplashScreen" Source="./Splash.png"/-->
<Value Id="Common.Images.CompanyLogo" Source="pack://siteoforigin:,,,/CompanyLogo.png" ResourceKey="" />
<Value Id="Windows.Common.Copyright" Text="2009-2014 My New Copyright."/>
<Value Id="Windows.AboutWindow.TextBlockWarning" Text="Warning: "/>
<Value Id="Windows.Common.Text.InteractionWorkspace" Text="NewCO"/>
</Dictionary>
```

For information about URIs in Windows Presentation Foundation (WPF), see: [\[1\]](#)

Localization

To dynamically change the language in your view, modify the XAML by using the following sample:

[XAML]

```
<UserControl xmlns:loc="http://schemas.tomer.com/winfx/2006/xaml/presentation">
```

```
<Expander>
  <Expander.Header>
    <TextBlock loc:Translate.Uid="DispositionCodeView.TextBlockDisposition"
      Text="{loc:Translate Default=The Disposition}" />
  </Expander.Header>
  <Button/>
</Expander>
</UserControl>
```

Refer to `DispositionCodeView.TextBlockDisposition` in the language XML file. For English, modify the `Genesyslab.Desktop.Modules.Windows.en-US.xml` file as shown in the following example:

[XML]

```
<Dictionary EnglishName="English" CultureName="English" Culture="en-US">
  <Value Id="DispositionCodeView.TextBlockDisposition" Text="The Disposition"/>
</Dictionary>
```

For French, modify the `Genesyslab.Desktop.Modules.Windows.fr-FR.xml` file as shown in the following example:

[XML]

```
<Dictionary EnglishName="French" CultureName="France" Culture="fr-FR">
  <Value Id="DispositionCodeView.TextBlockDisposition" Text="La Disposition"/>
</Dictionary>
```

The language can also be changed within the code itself, as shown in the following example:

[C#]

```
string text = LanguageDictionary.Current.Translate("DispositionCodeView.TextBlockDisposition", "Text");
```

Parameterization

Workspace Desktop Edition is configured as a role-based application. For example, if an agent is assigned the task of `TeamCommunicator`, the Click-Once group file that is related to this task is downloaded when the application starts up and the associated module is loaded in RAM. The GUI that is specific to this task is then displayed only to the agents that are assigned the `TeamCommunicator` task. The task section in the following example enables you to download and execute a custom module extension. If the task name (`InteractionWorkspace.TeamCommunicator.canUse`) is configured in Configuration Manager, the required group of files (`TeamCommunicator`) is downloaded, and the module (`TeamCommunicatorModule`) are

executed. This parameterization functionality is configured in the `InteractionWorkspace.exe.config` file, as shown in the following example:

[XML]

```
<configuration>
  ...
  <tasks>
    ...
    <task name="InteractionWorkspace.Features.TeamCommunicator"
      clickOnceGroupsToDownload="TeamCommunicator"
      modulesToLoad="TeamCommunicatorModule" />
    ...
  </tasks>

  <modules>
    ...
    <module assemblyFile="Genesyslab.Desktop.Modules.TeamCommunicator.dll"
      moduleType="Genesyslab.Desktop.Modules.TeamCommunicator.TeamCommunicatorModule"
      moduleName="TeamCommunicatorModule"
      startupLoaded="false"/>
    ...
  </modules>
  ...
</configuration>
```

Parameterization functionality can also be accomplished by loading a custom module conditioned with a task. In the Configuration Manager, a role must be configured with the name of the task. In this example, the task is named `InteractionWorkspace.ExtensionSample.canUse` and assigned to the agent. This custom parameterization functionality is configured in the `ExtensionSample.module-config` file, as shown in the following example:

[XML]

```
<configuration>
  <configSections>
    <section name="tasks"
      type="Genesyslab.Desktop.Infrastructure.Config.TasksSection, Genesyslab.Desktop.Infrastructure" />
    <section name="modules"
      type="Microsoft.Practices.Composite.Modularity.ModulesConfigurationSection, Microsoft.Practices.Composite" />
  </configSections>
  <tasks>
    <task name="InteractionWorkspace.ExtensionSample.canUse"
      clickOnceGroupsToDownload="ExtensionSample"
      modulesToLoad="ExtensionSampleModule" />
  </tasks>
</configuration>
```

```
</tasks>
<modules>
  <module assemblyFile="Genesyslab.Desktop.Modules.ExtensionSample.dll"
    moduleType="Genesyslab.Desktop.Modules.ExtensionSample.ExtensionSampleModule"
    moduleName="ExtensionSampleModule"
    startupLoaded="false" />
</modules>
</configuration>
```

Internationalization

WPF and .NET work with Unicode strings, so internationalization does not normally require extra coding. However, there are some potential issues to consider when creating your custom code, such as:

- Strings coming from the server might not be in true Unicode.
- The language might not be read/written from left to right (for example, Arabic languages).
- The correct font must be installed on the agents system.

Screen Reader Compatibility

The Microsoft UI Automation API is used for WPF applications that require accessibility functionality. The following two tools are available to assist you in developing applications that are compliant with accessibility software, such as Job Access With Speech (JAWS):

- UISpy.exe (Microsoft Windows SDK)--Displays the GUI controls tree along with the UIAutomation properties of the controls (such as AccessKey, Name, and others)
- Narrator (Microsoft Windows)--Reads the content of a window

Use the following code sample to add a name to a GUI control in the XAML file:

[XAML]

```
<TextBox Name="textBoxUserName" AutomationProperties.Name="UserName" />
```

The `AutomationProperties.Name` of the `TextBox` control is automatically set with the content value of a `Label` control. If a GUI control already has a `Label` control the XAML file looks similar to the following example:

[XAML]

```
<Label Target="{Binding ElementName=textBoxUserName}" Content="_UserName" />
<TextBox Name="textBoxUserName" />
```

Note: The `AutomationProperties.Name` must be localized.

Themes

Genesys recommends that you place the control styles and color resources that are used in the application into an XAML file containing a WPF `ResourceDictionary`. This enables you to modify and extend an existing theme. To make the themes extensible, use `ThemeManager` to register all the available themes in the application. When a theme is changed, `ThemeManager` copies this `ResourceDictionary` to the global application `ResourceDictionary`. All previously copied styles and brushes are overwritten with the new ones. **Note:** The XAML file that you create to contain the control styles and color resources is not a Microsoft Composite Application Library (CAL) module.

To add a new theme, you must first create a new theme in a `.module-config` file, for example:

[.module-config]

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
<configSections>
<section name="themes" type="Genesyslab.Desktop.Infrastructure.Theming.ThemesSection, Genesyslab.Desktop.Infrastructure" />
</configSections>
<themes>
<theme name="CustomTheme" displayNameKey="Theme.Custom.DisplayName"
mainResourceDictionary="/Genesyslab.Desktop.Modules.CustomThemeSample;component/Resources/themes/CustomTheme.xaml">
<xmlDictionaries>
<xmlDictionary name="NewTheme" path=".\\Resources\\ResourcesDefinitionCustom.xml"></xmlDictionary>
</xmlDictionaries>
</theme>
```



```
</themes>
</configuration>
```

The CustomTheme.xaml file must declare the main resource dictionary of the new style and Custom Color dictionary, for example:

[XAML]

```
<ResourceDictionary xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
  <ResourceDictionary.MergedDictionaries>
    <!-- New IW Style -->
    <ResourceDictionary Source="/Genesyslab.Desktop.WPFCommon;component/Resources/NewStyles/NewStylesResourceLibrary.xaml"/>
    <ResourceDictionary Source="/Genesyslab.Desktop.Modules.CustomThemeSample;component/Resources/ColorBrushes/CustomDefaultColorTheme.xaml"/>
  </ResourceDictionary.MergedDictionaries>
</ResourceDictionary>
```

Use the `gui.themes` option to add the new theme name.

Loosely-coupled Application Library and Standard Controls

Workspace Desktop Edition is a modular Windows Presentation Foundation (WPF) client application and uses the standard WPF controls. This section provides information about these controls. The [Loosely-coupled Application Library](#) is part of the Composite Application Guidance which aims to produce a flexible WPF client application that is loosely coupled. The following graphical tree shows a typical composite application built with loosely-coupled applications:

```
Shell
  Region1
    View11
    View12
  Region2
    View21
      Region21
        View211
        View212
Shell
```

The typical GUI is composed of a shell, region(s), and view(s). The shell is the main window of the application where the primary user interface (UI) content is contained. The shell is usually a single main window that contains multiple views. The shell can contain named regions where modules can add views. A region is a rectangular graphical area that is embedded in a shell or a view and can contain one or more views. Views are the composite portions of the user interface that are contained in the window(s) of the shell. Views are the elementary pieces of UI, such as a user control that defines a rectangular portion of the client area in the main window.

Views

A view contains controls that display data. The logic that is used to retrieve the data, handle user events, and submit the changes to the data is often included in the view. When this functionality is included in the View, the class becomes complex, and is difficult to maintain and test. You can resolve these issues by using [Presentation Patterns](#) and [Data Binding](#).

Presentation Patterns

Use patterns to separate the responsibilities of the display and the behavior of the application into different classes, named the View and the View Model. Genesys suggests the following presentation patterns:

- Model-View-ViewModel (MVVM)
- Model-View-PresentationModel (Presentation Model)

The MVVM pattern is used in Genesys samples.

- The Model is similar to having several data sources (InteractionService from Enterprise Services, Statistics from the Platform SDK, or any other data).
- The View is a stateless UserControl; a graphical interface with no behavior.
- The ViewModel is an adaptation layer between the Model and the View. It offers the Model data to the View. The behavior of the View is defined in this layer. For instance, the View launches the commands, but the commands are implemented in the ViewModel.

Each view consists of several classes. The VoiceView is described in the following table:

Roles	Classes/Interfaces	Files	Description
View	IVoiceView	IVoiceView.cs	The interface
View	VoiceView	VoiceView.xaml VoiceView.xaml.cs	The implementation of the IVoiceView. VoiceView.xaml is the XAML file that describes the view and VoiceView.xaml.cs contains the code behind.
ViewModel	IVoiceViewModel	IVoiceViewModel.cs	The interface
ViewModel	VoiceViewModel	VoiceViewModel.cs	The implementation of the IVoiceViewModel.

Data Binding

When you use presentation patterns in application development you have the option of using the data-binding capabilities that are provided by the WPF. Data-binding is used to bind elements to application data. The bound elements automatically reflect changes when the data changes its value. For example, if the DataContext property of the VoiceView class is set to an instance of the VoiceViewModel class, then the Text property of a TextBlock control can have a DataBinding toward the PhoneNumber property of the VoiceViewModel class. By default it is a two-way binding. If the value of either the VoiceViewModel.PhoneNumber or the TextBlock display changes then the other changes as well. The following example also shows how the command VoiceViewModel.AnswerCallCommand can be initiated from the VoiceView:

```
<TextBlock Text="{Binding PhoneNumber}"/>
<Button Command="{Binding AnswerCallCommand}">Answer Call</Button>
```

Note: **Modularity** requires that each interface is registered in the module initialization. See [Customize Views and Regions](#) for details on how to register an interface.

Tips and Tricks


When you need to control several Views, you can use a Controller class to coordinate the activities of multiple Views (and others controllers). The ViewModel is created by the View, and the Views are created and managed by the Controllers. The following logical tree is a depiction of the relationship between the instantiated classes:

```
Controller1
  Controller11
```

```
View111
  ViewModel111
View112
  ViewModel112
View12
  ViewModel12
Controller2
View21
  ViewModel21
View22
  ViewModel22
```

Use the information provided in this section along with the information in the Customizing Workspace Desktop Edition topic [to create your own view](#).

Reference for Windows

	<p>Purpose: To present the list of windows that your application can customize.</p>
---	--

Windows and Views

Available windows and views	Related pages
<p>Added in 8.5:</p> <ul style="list-style-type: none"> Window InsertTableView Window TablePropertiesView 	<p>Windows Utilities</p>
<ul style="list-style-type: none"> Window: ToolbarWindow <ul style="list-style-type: none"> Multi-view: InteractionBarCasesRegion Multi-view: MyWorkplaceContainerView Multi-view: MainToolbarContainerView Multi-view: MainToolbarWorkbinsContainerView 	<p>Toolbar Window</p>
<ul style="list-style-type: none"> Window: GadgetDialerWindow Deprecated since: 8.5.0 Window: PopupInteractionContainerView Window: SystemTrayView Window: GadgetMyWorkspaceWindow <ul style="list-style-type: none"> View: GadgetMyChannelsWindow View: GadgetMyHistoryWindow View: GadgetWorkbinsContainerView View: GadgetMyStatisticsWindow View: GadgetCallCenterStatisticsWindow View: GadgetMyMessages View: GadgetMyCampaignsWindow View: GadgetContactContainerView Window: GadgetStatisticsWindow 	<p>Gadget Windows</p>
<ul style="list-style-type: none"> InteractionsWindow 	<p>Interactions Window</p>

Available windows and views	Related pages
<ul style="list-style-type: none"> • Window: ToasterMessageWindow • Window: ToasterWindow 	<p style="color: #c00000;">Toaster Window</p>
<ul style="list-style-type: none"> • ContactSearchWindow • ContactMergeWindow • ContainerView 	<p style="color: #c00000;">Windows Utilities, in Contacts and Favorites</p>
<ul style="list-style-type: none"> • ForwardView • CancelForwardView 	<p style="color: #c00000;">Windows Utilities, in Forward and Cancel Forward</p>
<ul style="list-style-type: none"> • IWMessageBoxView • AboutWindow • AuthenticationWindow • AuthenticationPasswordWindow • BroadcastMessageWindow • MediaInformationView • PrintPreviewWindow • EndWorkflowOpenMediaView • CalendarWindowView 	<p style="color: #c00000;">Window Utilities, in Additional Accessories</p>

Interactions Window

Tip

If you want to highlight the functional areas in the Workspace UI with the red boxes and descriptions shown in the screen shots in this topic, on a *non-production* developer workstation, either configure the value of the `keyboard.shortcut.action.test.hidden` developer option in the `interactionworkspace.exe.config` file or in Configuration Server using GAX, to a keyboard short cut, such as: `Ctrl+P`. When you press the keyboard shortcut, the functional areas are highlighted.

Genesys recommends that you set this option at the agent level for testing purposes before setting it higher up in the option hierarchy or in the `interactionworkspace.exe.config` file.

Upon execution of the file, the following files are generated in the execution folder:

- `DumpChainOfCommandsInHTML.htm` containing all chains of commands available in the product in html format.
- `DumpChainOfCommandsInWikiMarkup.txt` containing all chains of commands available in the product in wiki format.
- `DumpKeyboardShortcut.htm` containing the list of all commands that can be triggered by a shortcut, and the shortcut configured if any.
- `DumpViewTree.htm` containing all regions.
- `DumpFullTreeView.htm` (when full tree view is specified) containing view tree names.



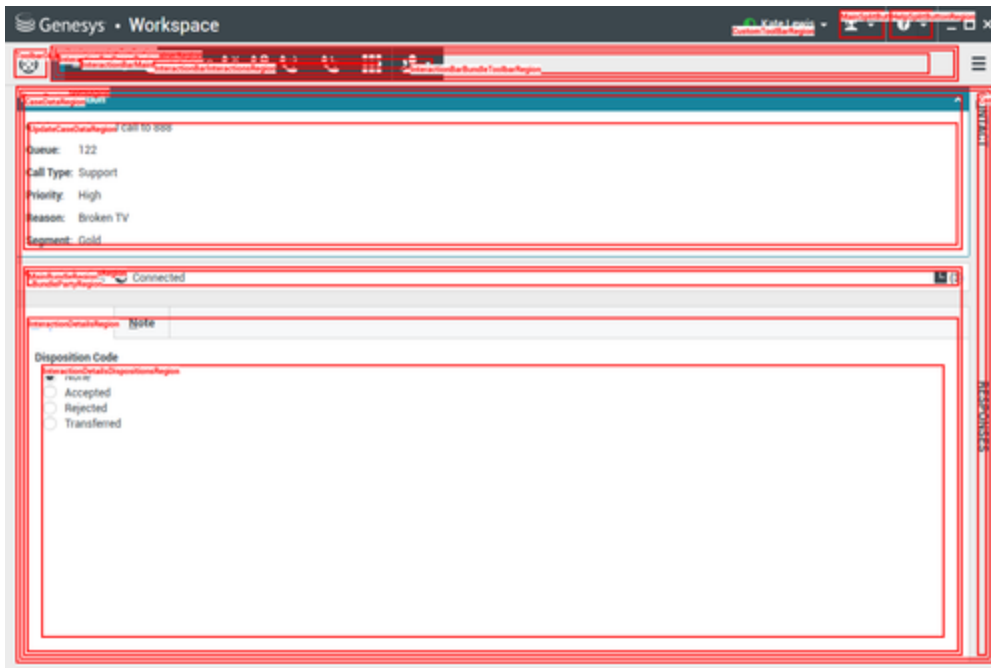
Purpose: Presents for the Interactions window, the list of default views and regions that your application can customize.

Window: InteractionsWindow

Available since: 8.0.1; updated in: 8.1.1, 8.1.2, 8.1.3, 8.1.4, 8.5

Pinned Interaction View

The `CaseView` root container of the Pinned Interaction View includes the same regions and views than the `CasesRegion` root container of the 8.1 Interaction Window View. If your customization requires the `CasesRegion` root container (not displayed in Pinned mode), Genesys recommends that you work in the legacy interaction window presentation mode.



Displaying a new case in Pinned mode. (Full resolution)

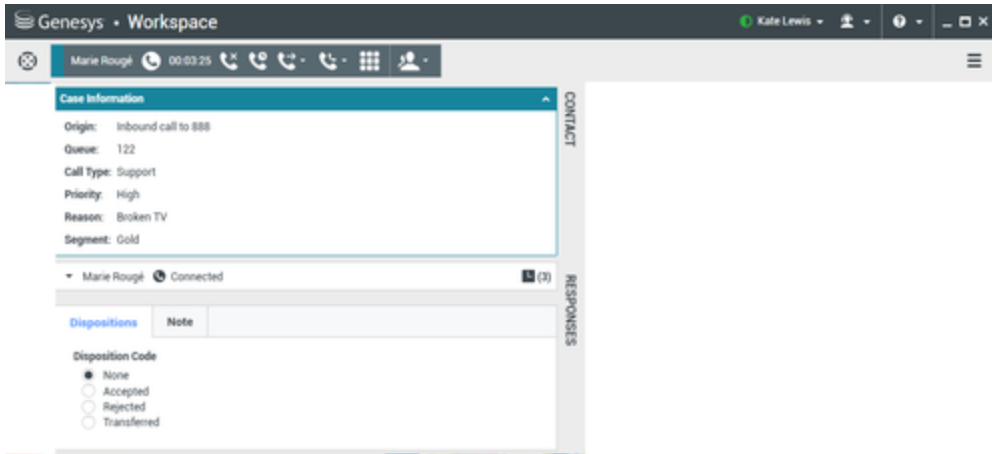
Important

All other sub-regions and sub-views of the former Interaction Window View are still available in the pinned interaction mode.

Floating Interaction View

The Region structure of the Floating Interaction View is a subset of the Pinned Interaction View described above:

- Its root container is CaseView;
- To add buttons to the side button menu, you must implement buttons in the CaseViewSideButtonRegion region; all the cascading material that they contain are not instantiated;
- This mode displays the left part of the Interaction View;
- The CaseViewSideButtonRegion region is replaced with the new CaseViewPopupSideButtonRegion region which enables you to insert side buttons in floating mode.
- Only one case can be displayed at a time.



Displaying a new case in floating mode.

If you implement side buttons in `CaseViewPopupSideButtonRegion` region, your custom code must trigger a programmatic switch to display the associated views when the buttons are clicked. You can do this by calling the following method:

```
IManagerInteractionContainer.ShowCaseViewInMainToolBar(<case_id>, <side_view_name>);
```

InteractionsWindowView

The main window displays the given interaction and the associated cases; it contains the following views and regions:

- **Window:** `InteractionsWindowView (IInteractionsWindow)` <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>
 - **Region:** `CasesRegion (Multi-view)`
 - **View:** `CaseView (ICaseView)` <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>
 - **Region:** `CaseViewSideButtonRegion (Multi-view)`
 - **View:** `InteractionContainerSideButtonView (IInteractionContainerSideButtonView)` <ref name="Contacts">Genesyslab.Desktop.Modules.Contacts</ref>
 - **View:** `SRLSideButtonView (ISRLSideButtonView)` <ref name="StandarResponse">Genesyslab.Desktop.Modules.StandarResponse</ref>
 - **Region:** `CaseViewPopupSideButtonRegion (Multi-view)`
 - **View:** `InteractionContainerPopupSideButtonView (IInteractionContainerPopupSideButtonView)` <ref name="Contacts"/>
 - **View:** `SRLPopupSideButtonView (ISRLPopupSideButtonView)` <ref name="StandarResponse"/>
 - **Region:** `InteractionsErrorRegion`
 - **View:** `InteractionsErrorView (ILastAlertErrorView)` <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>
 - **Region:** `CaseDataRegion`

- **View:** CaseDataView (ICaseDataView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>
 - **Region:** UpdateCaseDataRegion
 - **View:** UpdateCaseDataView (IGenericDataView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>
- **Region:** ConsultationBundlesRegion (Multi-view)
 - **View:** MainBundleView (IMainBundleView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>
 - **Region:** MainBundleRegion
 - **View:** BundleView (IBundleView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>
 - **Region:** BundlePartyRegion
 - **View:** BundlePartyView (IBundlePartyView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>
 - **Region:** PartyRegion (Multi-view)
 - **View:** PartyView (IPartyView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>
 - **Region:** CustomBundlePartyRegion (Multi-view)
 - **View:** IScreenRecordingMonitorView (IScreenRecordingMonitorView) <ref name="ActiveRecording">Genesyslab.Desktop.Modules.ActiveRecording</ref>
 - **View:** InteractionQueueView (IInteractionQueueView) <ref name="Contacts">Genesyslab.Desktop.Modules.Contacts</ref>
 - **View:** ContactNotificationView (IContactNotificationView) <ref name="Contacts" />
 - **Region:** BundleToolbarContainerRegion
 - **View:** BundleToolbarView (IBundleToolbarView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>
 - **Region:** BundleCustomButtonRegion (Multi-view)
 - **View:** DonePullPreviewToolbarView (IDonePullPreviewToolbarView) <ref name="Outbound">Genesyslab.Desktop.Modules.Outbound</ref>
 - **View:** RescheduleToolbarView (IRescheduleToolbarView) <ref name="Outbound">Genesyslab.Desktop.Modules.Outbound</ref>
 - **View:** IxnMonitor (IIxnMonitor) <ref name="ActiveRecording">Genesyslab.Desktop.Modules.ActiveRecording</ref>
Deprecated in: 8.1.4
 - **View:** InteractionCallbackRescheduleToolbarView

```
(InteractionCallbackRescheduleToolBarView) <ref
name=
"Callback">Genesyslab.Desktop.Modules.Callback</ref>
```

- **View:** InteractionCallbackToolBarView
(InteractionCallbackToolBarView) <ref name="Callback" />
- **View:** TeamLeadToolBarView (ITeamLeadToolBarView) <ref name="TeamLead">Genesyslab.Desktop.Modules.TeamLead</ref>
- **View:** InteractionSmsToolBarView
(InteractionSmsToolBarView) <ref name="OpenMedia">Genesyslab.Desktop.Modules.OpenMedia</ref>
- **View:** InteractionInboundEmailToolBarView
(InteractionInboundEmailToolBarView) <ref name="OpenMedia">Genesyslab.Desktop.Modules.OpenMedia</ref>
- **View:** InteractionOutboundEmailToolBarView
(InteractionOutboundEmailToolBarView) <ref name="OpenMedia">Genesyslab.Desktop.Modules.OpenMedia</ref>
- **View:** InteractionChatToolBarView
(InteractionChatToolBarView) <ref name="OpenMedia">Genesyslab.Desktop.Modules.OpenMedia</ref>
- **View:** InteractionChatConsultToolBarView
(InteractionChatConsultToolBarView) <ref name="OpenMedia">Genesyslab.Desktop.Modules.OpenMedia</ref>
- **View:** InteractionIMToolBarView (InteractionIMToolBarView) <ref name="Voice">Genesyslab.Desktop.Modules.Voice</ref>
- **View:** InteractionPreviewToolBarView
(InteractionPreviewToolBarView) <ref name="Outbound">Genesyslab.Desktop.Modules.Outbound</ref>
- **View:** InteractionVoiceToolBarView
(InteractionVoiceToolBarView) <ref name="Voice">Genesyslab.Desktop.Modules.Voice</ref>
 - **Region:** InteractionVoiceCustomButtonRegion (Multi-view)
Updated in 8.5
 - **View:** SIPEndpointVideoView
(SIPEndpointVideoView) <ref name="SIPEndpointCommunication">Genesyslab.Desktop.Modules.SIPEndpointCommunication</ref>
Added in: 8.5.0
 - **View:** SIPEndpointSoundView
(SIPEndpointSoundView) <ref name="SIPEndpointCommunication" />
 - **View:** SIPMonitor (SIPMonitor) <ref name="ActiveRecording">Genesyslab.Desktop.Modules.ActiveRecording</ref>
 - **View:** IxnMonitor (IxnMonitor) <ref name="ActiveRecording" />

```
"OpenMedia">Genesyslab.Desktop.Modules.OpenMedia</ref>
```

- **Region:** InteractionDetailsRegion (Multi-view)
 - **View:** DispositionsView (IDispositionsView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>
 - **Region:** InteractionDetailsDispositionsRegion (Multi-view)
 - **View:** OutboundDispositionView (IOutboundDispositionView) <ref name="Outbound">Genesyslab.Desktop.Modules.Outbound</ref>
 - **View:** DispositionCodeView (IDispositionCodeView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>
 - **View:** NotepadViewNotepadVoiceNotepadView (INotepadView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>
- **View:** BundleView (IBundleView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>
 - **Region:** BundlePartyRegion
Added in: 8.1.3
 - **View:** BundlePartyView (IBundlePartyView) <ref name="Windows" />
 - **Region:** PartyRegion (Multi-view)
 - **View:** PartyView (IPartyView) <ref name="Windows" />
 - **Region:** CustomBundlePartyRegion (Multi-view)
 - **View:** IScreenRecordingMonitorView (IScreenRecordingMonitorView) <ref name="ActiveRecording" />
 - **View:** InteractionQueueView (IInteractionQueueView) <ref name="Contacts" />
 - **View:** ContactNotificationView (IContactNotificationView) <ref name="Contacts" />
 - **Region:** BundleToolbarContainerRegion
 - **View:** TeamLeadToolBarView (ITeamLeadToolBarView) <ref name="TeamLead" />
 - **View:** InteractionPreviewToolBarView (IInteractionPreviewToolBarView) <ref name="Outbound" />
 - **View:** InteractionCallbackToolBarView (IInteractionCallbackToolBarView) <ref name="Callback" />
 - **View:** InteractionSmsToolBarView (IInteractionSmsToolBarView) <ref name="OpenMedia" />
 - **View:** InteractionWorkItemToolBarView (IInteractionWorkItemToolBarView) <ref name="OpenMedia" />
 - **Region:** CustomWorkItemToolBarRegion (Multi-view)

-
- **View:** InteractionOutboundEmailToolBarView
(InteractionOutboundEmailToolBarView) <ref name="OpenMedia" />
 - **View:** InteractionInboundEmailToolBarView
(InteractionInboundEmailToolBarView) <ref name="OpenMedia" />
 - **View:** BundleToolBarView (IBundleToolBarView) <ref name="Windows" />
 - **Region:** BundleCustomButtonRegion (Multi-view)
 - **View:** InteractionCallbackRescheduleToolBarView
(InteractionCallbackRescheduleToolBarView) <ref name="Callback" />
 - **View:** DonePullPreviewToolBarView
(IDonePullPreviewToolBarView) <ref name="Outbound" />
 - **View:** RescheduleToolBarView (IRescheduleToolBarView) <ref name="Outbound" />
 - **View:** InteractionVoiceToolBarView (InteractionVoiceToolBarView) <ref name="Voice" />
 - **Region:** InteractionVoiceCustomButtonRegion (Multi-view)
Updated in 8.5
 - **View:** SIPEndpointVideoView (ISIPEndpointVideoView) <ref name="SIPEndpointCommunication">Genesyslab.Desktop.Modules.SIPEndpointCommunication</ref>
Added in: 8.5.0
 - **View:** SIPEndpointSoundView (ISIPEndpointSoundView) <ref name="SIPEndpointCommunication" />
 - **View:** SIPMonitor (ISIPMonitor) <ref name="ActiveRecording">Genesyslab.Desktop.Modules.ActiveRecording</ref>
 - **View:** IxnMonitor (IIxnMonitor) <ref name="ActiveRecording" />
 - **View:** InteractionIMToolBarView (InteractionIMToolBarView) <ref name="Voice" />
 - **View:** InteractionChatToolBarView (InteractionChatToolBarView) <ref name="OpenMedia" />
 - **View:** InteractionChatConsultToolBarView
(InteractionChatConsultToolBarView) <ref name="OpenMedia" />
 - **Region:** CustomInfoOnInteractionsBundleRegion (Multi-view)
 - **View:** InteractionCallbackInformationView
(InteractionCallbackInformationView) <ref name="Callback" />
 - **Region:** CallbackInformationRegion
 - **View:** CallbackInformationView (IGenericDataView) <ref name="Windows" />
 - **View:** OutboundRecordView (IOutboundRecordView) <ref name="Outbound" />
-

- **Region:** UpdateOutboundFieldsRegion
 - (IGenericDataView) <ref name="Windows" />
- **Region:** InteractionsBundleRegion (Multi-view)
 - **View:** IMView (IIMCallView) <ref name="Voice" />
 - **View:** ChatView (IChatView) <ref name="OpenMedia" />
 - **View:** InboundEmailView (IInboundEmailView) <ref name="OpenMedia" />
 - **Region:** EmailAttachmentsContentControlRegion
 - **View:** EmailAttachmentsContentControlView (IAttachmentView) <ref name="Windows" />
 - **View:** OutboundEmailView (IOutboundEmailView) <ref name="OpenMedia" />
 - **Region:** OutboundEmailAttachmentsContentControlRegion
 - **View:** OutboundEmailAttachmentsContentControlView (IAttachmentView) <ref name="Windows" />
 - **View:** WorkItemView (IWorkItemView) <ref name="OpenMedia" />
 - **View:** SmsView (ISmsView) <ref name="OpenMedia" />
- **Region:** InteractionDetailsRegion (Multi-view)
 - **View:** NotepadVoiceNotepadView (INotepadView) <ref name="Windows" />
 - **View:** DispositionsView (IDispositionsView) <ref name="Windows" />
 - **Region:** InteractionDetailsDispositionsRegion (Multi-view)
 - **View:** OutboundDispositionView (IOutboundDispositionView) <ref name="Outbound" />
 - **View:** DispositionCodeView (IDispositionCodeView) <ref name="Windows" />
- **View:** BundleView (IBundleView) <ref name="Windows" />
 - **Region:** BundlePartyRegion
 - **View:** BundlePartyView (IBundlePartyView) <ref name="Windows" />
 - **Region:** PartyRegion (Multi-view)
 - **View:** PartyView (IPartyView) <ref name="Windows" />
 - **Region:** CustomBundlePartyRegion (Multi-view)
 - **View:** IScreenRecordingMonitorView (IScreenRecordingMonitorView) <ref name="ActiveRecording" />
 - **View:** InteractionQueueView (IInteractionQueueView) <ref name="Contacts" />
 - **View:** ContactNotificationView (IContactNotificationView)

```
<ref name="Contacts" />
```

- **Region:** BundleToolBarContainerRegion
 - **View:** TeamLeadToolBarView (ITeamLeadToolBarView) <ref name="TeamLead" />
 - **View:** InteractionPreviewToolBarView (IInteractionPreviewToolBarView) <ref name="Outbound" />
 - **View:** InteractionCallbackToolBarView (IInteractionCallbackToolBarView) <ref name="Callback" />
 - **View:** InteractionSmsToolBarView (IInteractionSmsToolBarView) <ref name="OpenMedia" />
 - **View:** InteractionWorkItemToolBarView (IInteractionWorkItemToolBarView) <ref name="OpenMedia" />
 - **Region:** CustomWorkItemToolBarRegion (Multi-view)
 - **View:** InteractionOutboundEmailToolBarView (IInteractionOutboundEmailToolBarView) <ref name="OpenMedia" />
 - **View:** InteractionInboundEmailToolBarView (IInteractionInboundEmailToolBarView) <ref name="OpenMedia" />
 - **View:** BundleToolBarView (IBundleToolBarView) <ref name="Windows" />
 - **Region:** BundleCustomButtonRegion (Multi-view)
 - **View:** InteractionCallbackRescheduleToolBarView (IInteractionCallbackRescheduleToolBarView) <ref name="Callback" />
 - **View:** DonePullPreviewToolBarView (IDonePullPreviewToolBarView) <ref name="Outbound" />
 - **View:** RescheduleToolBarView (IRescheduleToolBarView) <ref name="Outbound" />
 - **View:** InteractionVoiceToolBarView (IInteractionVoiceToolBarView) <ref name="Voice" />
 - **Region:** InteractionVoiceCustomButtonRegion (Multi-view)
Updated in 8.5
 - **View:** SIPEndpointVideoView (ISIPEndpointVideoView) <ref name="SIPEndpointCommunication">Genesyslab.Desktop.Modules.SIPEndpointCommunication</ref>
Added in: 8.5.0
 - **View:** SIPEndpointSoundView (ISIPEndpointSoundView) <ref name="SIPEndpointCommunication" />
 - **View:** SIPMonitor (ISIPMonitor) <ref name="ActiveRecording">Genesyslab.Desktop.Modules.ActiveRecording</ref>
 - **View:** IxnMonitor (IIxnMonitor) <ref name="ActiveRecording"/>
 - **View:** InteractionIMToolBarView (IInteractionIMToolBarView) <ref name="Voice" />
 - **View:** InteractionChatToolBarView (IInteractionChatToolBarView) <ref

```
name="OpenMedia" />
```

- **View:** InteractionChatConsultToolBarView (IInteractionChatConsultToolBarView) <ref name="OpenMedia" />
- **Region:** CustomInfoOnInteractionsBundleRegion (Multi-view)
 - **View:** InteractionCallbackInformationView (IInteractionCallbackInformationView) <ref name="Callback" />
 - **Region:** CallbackInformationRegion
 - **View:** CallbackInformationView (IGenericDataView) <ref name="Windows" />
 - **View:** OutboundRecordView (IOutboundRecordView) <ref name="Outbound" />
 - **Region:** UpdateOutboundFieldsRegion
 - **View:** UpdateOutboundFieldsView (IGenericDataView) <ref name="Windows" />
- **Region:** InteractionsBundleRegion (Multi-view)
 - **View:** IMView (IIMCallView) <ref name="Voice" />
 - **View:** ChatView (IChatView) <ref name="OpenMedia" />
 - **View:** InboundEmailView (IInboundEmailView) <ref name="OpenMedia" />
 - **Region:** EmailAttachmentsContentControlRegion
 - **View:** EmailAttachmentsContentControlView (IAttachmentView) <ref name="Windows" />
 - **View:** OutboundEmailView (IOutboundEmailView) <ref name="OpenMedia" />
 - **Region:** OutboundEmailAttachmentsContentControlRegion
 - **View:** OutboundEmailAttachmentsContentControlView (IAttachmentView) <ref name="Windows" />
 - **View:** WorkItemView (IWorkItemView) <ref name="OpenMedia" />
 - **View:** SmsView (ISmsView) <ref name="OpenMedia" />
- **Region:** InteractionWorksheetRegion
 - **View:** InteractionContainerView (IInteractionContainerView) <ref name="Contacts">Genesyslab.Desktop.Modules.Contacts</ref>
 - **Region:** AssignContactDirectoryRegion (Multi-view)
 - **View:** AssignContactDirectoryView (IContactDirectoryView) <ref name="Contacts">Genesyslab.Desktop.Modules.Contacts</ref>
 - **Region:** ContactDirectoryErrorRegion
 - **View:** ContactDirectoryErrorView (ILastAlertErrorView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>

- **Region:** CaseContactRegion (Multi-view)
 - **View:** InteractionContactInformationView (IContactInformationView) <ref name="Contacts">Genesyslab.Desktop.Modules.Contacts</ref>
 - **Region:** ContactInformationErrorRegion
 - **View:** ContactInformationErrorView (ILastAlertErrorView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>
 - **View:** InteractionContactHistoryView (IMyContactHistoryView) <ref name="Contacts">Genesyslab.Desktop.Modules.Contacts</ref>
 - **Region:** ContactHistoryErrorRegion
 - **View:** ContactHistoryErrorView (ILastAlertErrorView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>
 - **Region:** ContactTabBottomHistoryMultiRegion (Multi-view)
 - **View:** ContactDetailView (IContactDetailView) <ref name="Contacts">Genesyslab.Desktop.Modules.Contacts</ref>
 - **Region:** ContactDetailRegion
 - **View:** FormerContactDetailContentView (IFormerContactDetailContentView) <ref name="Contacts">Genesyslab.Desktop.Modules.Contacts</ref>
 - **Region:** ControlListAttachmentRegion
 - **View:** ControlListAttachmentView (IAttachmentView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>
 - **View:** CallbackContactDetailContentView (ICallbackContactDetailContentView) <ref name="Callback" />
 Added in: 8.1.4
 - **Region:** ContactDetailCallbackInformationRegion
 - **View:** ContactDetailCallbackInformationView (IGenericDataView) <ref name="Windows" />
 - **View:** OpenMediaContactDetailContentView (IContactDetailView|OpenMediaContactDetailContentView) <ref name="OpenMedia">Genesyslab.Desktop.Modules.OpenMedia</ref>
 - **Region:** ControlListAttachmentRegion
 - **View:** ControlListAttachmentView (IAttachmentView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>
 - **View:** <ContactHistoryNotepadView (INotepadView) <ref

- ```
name=
"Windows">Genesyslab.Desktop.Modules.Windows</ref>
```
- **View:** ContactHistoryCaseDataView (IStaticCaseDataView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>
    - **Region:** StaticCaseDataRegion
      - **View:** StaticCaseDataView (IGenericDataView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>
    - **Region:** StaticDispositionCodeRegion
      - **View:** StaticDispositionCodeView (IGenericDataView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>
  - **View:** ResponsesView (IResponsesView) <ref name="StandardResponse">Genesyslab.Desktop.Modules.StandarResponse</ref>
    - **Region:** ResponsesErrorRegion
      - **View:** ResponsesErrorView (ILastAlertErrorView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>
    - **Region:** ResponseAttachmentsContentControlRegion
      - **View:** AttachmentViewResponseAttachmentsContentControlView (IAttachmentView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>

## Window PopupInteractionContainerView

### Added in 8.5

- **Window:** PopupInteractionContainerView (IPopupInteractionContainerView) <ref name="Windows"/>
  - **Region:** CasesRegion
    - **View:** CaseView (ICaseView) <ref name="Windows"/>
      - **Region:** CaseViewButtonRegion (Multi-view)
      - **Region:** CaseViewSideButtonRegion (Multi-view)
        - **View:** InteractionContainerSideButtonView (IInteractionContainerSideButtonView) <ref name="Contacts"/>
        - **View:** SRLSideButtonView (ISRLSideButtonView) <ref name="StandardResponse"/>
    - **Region:** CaseViewPopupSideButtonRegion (Multi-view)
      - **View:** InteractionContainerPopupSideButtonView (IInteractionContainerPopupSideButtonView) <ref name="Contacts"/>

- **View:** SRLPopupSideButtonView (ISRLPopupSideButtonView) <ref name="StandardResponse"/>
- **Region:** InteractionsErrorRegion
  - **View:** InteractionsErrorView (ILastAlertErrorView) <ref name="Windows"/>
- **Region:** CaseDataRegion
  - **View:** CaseDataView (ICaseDataView) <ref name="Windows"/>
    - **Region:** UpdateCaseDataRegion
      - **View:** UpdateCaseDataView (IGenericDataView) <ref name="Windows"/>
- **Region:** ConsultationBundlesRegion (Multi-view)
  - **View:** MainBundleView (IMainBundleView) <ref name="Windows"/>
    - **Region:** MainBundleRegion
      - **View:** BundleView (IBundleView) <ref name="Windows"/>
        - **Region:** BundlePartyRegion
          - **View:** BundlePartyView (IBundlePartyView) <ref name="Windows"/>
            - **Region:** PartyRegion (Multi-view)
              - **View:** PartyView (IPartyView) <ref name="Windows"/>
                - **Region:** CustomBundlePartyRegion (Multi-view)
                  - **View:** IScreenRecordingMonitorView (IScreenRecordingMonitorView) <ref name="ActiveRecording"/>
                  - **View:** InteractionQueueView (IInteractionQueueView) <ref name="Contacts"/>
                  - **View:** ContactNotificationView (IContactNotificationView) <ref name="Contacts"/>
    - **Region:** BundleToolbarContainerRegion
      - **View:** TeamLeadToolbarView (ITeamLeadToolbarView) <ref name="TeamLead"/>
      - **View:** InteractionPreviewToolbarView (IInteractionPreviewToolbarView) <ref name="Outbound"/>
      - **View:** InteractionCallbackToolbarView (IInteractionCallbackToolbarView) <ref name="Callback"/>
      - **View:** InteractionSmsToolbarView (IInteractionSmsToolbarView) <ref name="OpenMedia"/>
      - **View:** InteractionWorkItemToolbarView

```
(IInteractionWorkItemToolBarView) <ref name=
"OpenMedia"/>
```

- **Region:** CustomWorkItemToolBarRegion (Multi-view)
- **View:** InteractionOutboundEmailToolBarView (IInteractionOutboundEmailToolBarView) <ref name="OpenMedia"/>
- **View:** InteractionInboundEmailToolBarView (IInteractionInboundEmailToolBarView) <ref name="OpenMedia"/>
- **View:** BundleToolBarView (IBundleToolBarView) <ref name="Windows"/>
  - **Region:** BundleCustomButtonRegion (Multi-view)
    - **View:** InteractionCallbackRescheduleToolBarView (IInteractionCallbackRescheduleToolBarView) <ref name="Callback"/>
    - **View:** DonePullPreviewToolBarView (IDonePullPreviewToolBarView) <ref name="Outbound"/>
    - **View:** RescheduleToolBarView (IRescheduleToolBarView) <ref name="Outbound"/>
- **View:** InteractionVoiceToolBarView (IInteractionVoiceToolBarView) <ref name="Voice"/>
  - **Region:** InteractionVoiceCustomButtonRegion (Multi-view)
    - **View:** SIPEndpointSoundView (ISIPEndpointSoundView) <ref name="SIPEndpointCommunication"/>
    - **View:** SIPEndpointVideoView (ISIPEndpointVideoView) <ref name="SIPEndpointCommunication"/>
    - **View:** SIPMonitor (ISIPMonitor) <ref name="ActiveRecording"/>
- **View:** InteractionIMToolBarView (IInteractionIMToolBarView) <ref name="Voice"/>
- **View:** InteractionChatToolBarView (IInteractionChatToolBarView) <ref name="OpenMedia"/>
- **View:** InteractionChatConsultToolBarView (IInteractionChatConsultToolBarView) <ref name="OpenMedia"/>
- **Region:** CustomInfoOnInteractionsBundleRegion (Multi-view)
  - **View:** InteractionCallbackInformationView (IInteractionCallbackInformationView) <ref name="Callback"/>

- **Region:** CallbackInformationRegion
      - **View:** CallbackInformationView (IGenericDataView) <ref name="Windows"/>
  - **View:** OutboundRecordView (IOutboundRecordView) <ref name="Outbound"/>
    - **Region:** UpdateOutboundFieldsRegion
      - **View:** UpdateOutboundFieldsView (IGenericDataView) <ref name="Windows"/>
- **Region:** InteractionsBundleRegion (Multi-view)
  - **View:** IMView (IIMCallView) <ref name="Voice"/>
  - **View:** ChatView (IChatView) <ref name="OpenMedia"/>
  - **View:** InboundEmailView (IInboundEmailView) <ref name="OpenMedia"/>
    - **Region:** EmailAttachmentsContentControlRegion
      - **View:** EmailAttachmentsContentControlView (IAttachmentView) <ref name="Windows"/>
  - **View:** OutboundEmailView (IOutboundEmailView) <ref name="OpenMedia"/>
    - **Region:** OutboundEmailAttachmentsContentControlRegion
      - **View:** OutboundEmailAttachmentsContentControlView (IAttachmentView) <ref name="Windows"/>
  - **View:** WorkItemView (IWorkItemView) <ref name="OpenMedia"/>
  - **View:** SmsView (ISmsView) <ref name="OpenMedia"/>
- **Region:** InteractionDetailsRegion (Multi-view)
  - **View:** NotepadVoiceNotepadView (INotepadView) <ref name="Windows"/>
  - **View:** DispositionsView (IDispositionsView) <ref name="Windows"/>
    - **Region:** InteractionDetailsDispositionsRegion (Multi-view)
      - **View:** OutboundDispositionView (IOutboundDispositionView) <ref name="Outbound"/>
      - **View:** DispositionCodeView (IDispositionCodeView) <ref name="Windows"/>
- **View:** BundleView (IBundleView) <ref name="Windows"/>
  - **Region:** BundlePartyRegion
    - **View:** BundlePartyView (IBundlePartyView) <ref name="Windows"/>

- **Region:** PartyRegion (Multi-view)
  - **View:** PartyView (IPartyView) <ref name="Windows"/>
    - **Region:** CustomBundlePartyRegion (Multi-view)
      - **View:** IScreenRecordingMonitorView (IScreenRecordingMonitorView) <ref name="ActiveRecording"/>
      - **View:** InteractionQueueView (IInteractionQueueView) <ref name="Contacts"/>
      - **View:** ContactNotificationView (IContactNotificationView) <ref name="Contacts"/>
- **Region:** BundleToolbarContainerRegion
  - **View:** TeamLeadToolBarView (ITeamLeadToolBarView) <ref name="TeamLead"/>
  - **View:** InteractionPreviewToolBarView (IInteractionPreviewToolBarView) <ref name="Outbound"/>
  - **View:** InteractionCallbackToolBarView (IInteractionCallbackToolBarView) <ref name="Callback"/>
  - **View:** InteractionSmsToolBarView (IInteractionSmsToolBarView) <ref name="OpenMedia"/>
  - **View:** InteractionWorkItemToolBarView (IInteractionWorkItemToolBarView) <ref name="OpenMedia"/>
    - **Region:** CustomWorkItemToolBarRegion (Multi-view)
  - **View:** InteractionOutboundEmailToolBarView (IInteractionOutboundEmailToolBarView) <ref name="OpenMedia"/>
  - **View:** InteractionInboundEmailToolBarView (IInteractionInboundEmailToolBarView) <ref name="OpenMedia"/>
  - **View:** BundleToolBarView (IBundleToolBarView) <ref name="Windows"/>
    - **Region:** BundleCustomButtonRegion (Multi-view)
      - **View:** InteractionCallbackRescheduleToolBarView (IInteractionCallbackRescheduleToolBarView) <ref name="Callback"/>
      - **View:** DonePullPreviewToolBarView (IDonePullPreviewToolBarView) <ref name="Outbound"/>
      - **View:** RescheduleToolBarView (IRescheduleToolBarView) <ref name="Outbound"/>
  - **View:** InteractionVoiceToolBarView (IInteractionVoiceToolBarView) <ref name="Voice"/>
    - **Region:** InteractionVoiceCustomButtonRegion (Multi-view)
      - **View:** SIPEndpointSoundView (ISIPEndpointSoundView) <ref



- 
- name="SIPEndpointCommunication"/>
      - **View:** SIPEndpointVideoView (ISIPEndpointVideoView) <ref name="SIPEndpointCommunication"/>
      - **View:** SIPMonitor (ISIPMonitor) <ref name="ActiveRecording"/>
    - **View:** InteractionIMToolBarView (IInteractionIMToolBarView) <ref name="Voice"/>
    - **View:** InteractionChatToolBarView (IInteractionChatToolBarView) <ref name="OpenMedia"/>
    - **View:** InteractionChatConsultToolBarView (IInteractionChatConsultToolBarView) <ref name="OpenMedia"/>
  - **Region:** CustomInfoOnInteractionsBundleRegion (Multi-view)
    - **View:** InteractionCallbackInformationView (IInteractionCallbackInformationView) <ref name="Callback"/>
      - **Region:** CallbackInformationRegion
        - **View:** CallbackInformationView (IGenericDataView) <ref name="Windows"/>
    - **View:** OutboundRecordView (IOutboundRecordView) <ref name="Outbound"/>
      - **Region:** UpdateOutboundFieldsRegion
        - **View:** UpdateOutboundFieldsView (IGenericDataView) <ref name="Windows"/>
  - **Region:** InteractionsBundleRegion (Multi-view)
    - **View:** IMView (IIMCallView) <ref name="Voice"/>
    - **View:** ChatView (IChatView) <ref name="OpenMedia"/>
    - **View:** InboundEmailView (IInboundEmailView) <ref name="OpenMedia"/>
      - **Region:** EmailAttachmentsContentControlRegion
        - **View:** EmailAttachmentsContentControlView (IAttachmentView) <ref name="Windows"/>
    - **View:** OutboundEmailView (IOutboundEmailView) <ref name="OpenMedia"/>
      - **Region:** OutboundEmailAttachmentsContentControlRegion
        - **View:** OutboundEmailAttachmentsContentControlView (IAttachmentView) <ref name="Windows"/>
    - **View:** WorkItemView (IWorkItemView) <ref name="OpenMedia"/>
    - **View:** SmsView (ISmsView) <ref name="OpenMedia"/>
  - **Region:** InteractionWorksheetRegion
-

- **View:** InteractionContainerView (IInteractionContainerView) <ref name="Contacts"/>
  - **Region:** AssignContactDirectoryRegion
    - **View:** AssignContactDirectoryView (IContactDirectoryView) <ref name="Contacts"/>
      - **Region:** ContactDirectoryErrorRegion
        - **View:** ContactDirectoryErrorView (ILastAlertErrorView) <ref name="Windows"/>
  - **Region:** CaseContactRegion (Multi-view)
    - **View:** InteractionContactInformationView (IContactInformationView) <ref name="Contacts"/>
      - **Region:** ContactInformationErrorRegion
        - **View:** ContactInformationErrorView (ILastAlertErrorView) <ref name="Windows"/>
  - **View:** InteractionContactHistoryView (IMyContactHistoryView) <ref name="Contacts"/>
    - **Region:** ContactHistoryErrorRegion
      - **View:** ContactHistoryErrorView (ILastAlertErrorView) <ref name="Windows"/>
    - **Region:** ContactTabBottomHistoryMultiRegion (Multi-view)
      - **View:** ContactDetailView (IContactDetailView) <ref name="Contacts"/>
        - **Region:** ContactDetailRegion
          - **View:** OpenMediaContactDetailContentView (IOpenMediaContactDetailContentView) <ref name="OpenMedia"/>
            - **Region:** ControlListAttachmentRegion
              - **View:** ControlListAttachmentView (IAttachmentView) <ref name="Windows"/>
          - **View:** CallbackContactDetailContentView (ICallbackContactDetailContentView) <ref name="Callback"/>
            - **Region:** ContactDetailCallbackInformationRegion
              - **View:** ContactDetailCallbackInformationView (IGenericDataView) <ref name="Windows"/>
      - **View:** FormerContactDetailContentView (IFormerContactDetailContentView) <ref name="

```
"Contacts"/>
```

- **Region:** ControllistAttachmentRegion
  - **View:** ControllistAttachmentView (IAttachmentView) <ref name="Windows"/>
- **View:** ContactHistoryNotepadView (INotepadView) <ref name="Windows"/>
- **View:** ContactHistoryCaseDataView (IStaticCaseDataView) <ref name="Windows"/>
  - **Region:** StaticCaseDataRegion
    - **View:** StaticCaseDataView (IGenericDataView) <ref name="Windows"/>
  - **Region:** StaticDispositionCodeRegion
    - **View:** StaticDispositionCodeView (IGenericDataView) <ref name="Windows"/>

## References

# Gadget Windows

## Tip

If you want to highlight the functional areas in the Workspace UI with the red boxes and descriptions shown in the screen shots in this topic, on a *non-production* developer workstation, either configure the value of the `keyboard.shortcut.action.test.hidden` developer option in the `interactionworkspace.exe.config` file or in Configuration Server using GAX, to a keyboard short cut, such as: `Ctrl+P`. When you press the keyboard shortcut, the functional areas are highlighted.

Genesys recommends that you set this option at the agent level for testing purposes before setting it higher up in the option hierarchy or in the `interactionworkspace.exe.config` file.

Upon execution of the file, the following files are generated in the execution folder:

- `DumpChainOfCommandsInHTML.htm` containing all chains of commands available in the product in html format.
- `DumpChainOfCommandsInWikiMarkup.txt` containing all chains of commands available in the product in wiki format.
- `DumpKeyboardShortcut.htm` containing the list of all commands that can be triggered by a shortcut, and the shortcut configured if any.
- `DumpViewTree.htm` containing all regions.
- `DumpFullTreeView.htm` (when full tree view is specified) containing view tree names.



**Purpose:** To present the list of gadget windows that your application can customize.  
Updated in 8.1.3, 8.1.4, 8.5

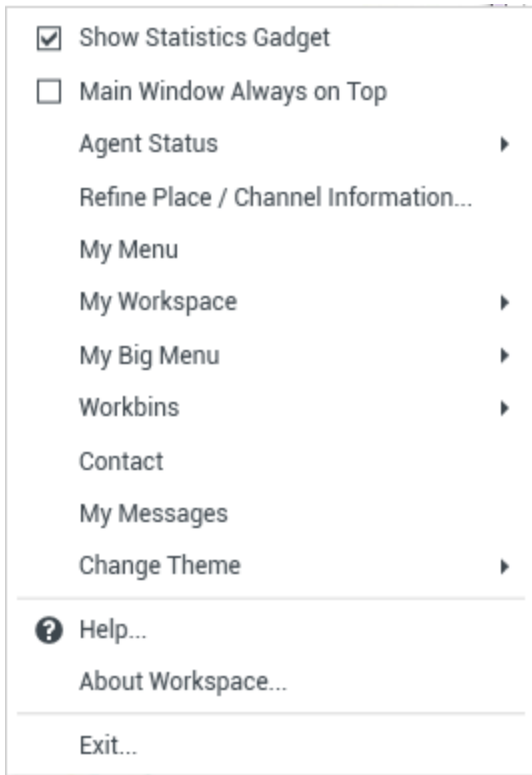
## Important

The Gadget mode is not compatible with the floating and pinned mode in 8.5. See the [migration page for further details](#).

## Window SystemTrayView

Available since: 8.1.0

- **Window:** `SystemTrayView (ISystemTrayContextMenu)` `<ref name="Windows" />`



SystemTrayView is a pop-up menu that you get by right-clicking on your minimized application.

## Window GadgetMyWorkspaceWindow

Updated in: 8.1.3

The GadgetMyWorkspaceWindow window contains a GadgetMyWorkspaceViewRegion region which enables you to display views and regions of the workspace:

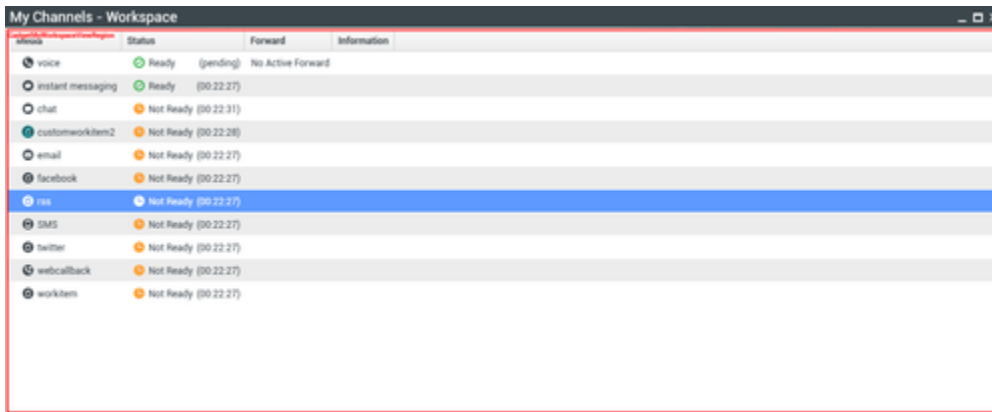
- **View: GadgetMyChannelsWindow**—The agent channels, with the place status.
- **View: GadgetMyHistoryWindow**—The agent's history.
- **View: GadgetWorkbinsContainerView** —The workbins explorer and menu.
- **View: GadgetMyStatisticsWindow**—The agent's statistics.
- **View: GadgetCallCenterStatisticsWindow**—The call center statistics.
- **View: GadgetMyMessages**—The agent's messages.
- **View: GadgetContactContainerView**—Displays the contact directory
- **View: GadgetMyCampaignsWindow**—The campaign's messages .

## View GadgetMyChannelsWindow

Available since: 8.0.1

**Description:** Displays the channel and place status of the agent.

- **Window:** A window identifier (IGadgetMyWorkspaceWindowView) <ref name="Windows" />
- **Region:** GadgetMyWorkspaceViewRegion
- **View:** GadgetMyChannelsWindow (IMyPlaceStatusView) <ref name="Windows" />



The GadgetMyChannelsWindow in the GadgetMyWorkspaceViewRegion

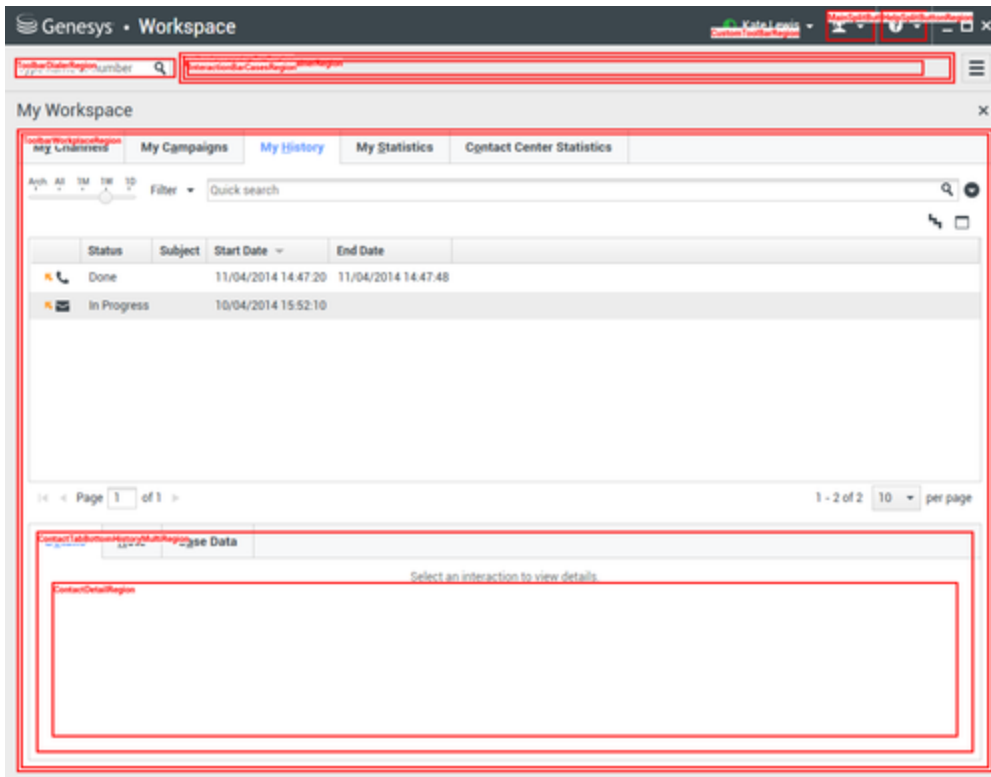
## View GadgetMyHistoryWindow

Available since: 8.0.1; updated in: 8.1.2

**Description:** Displays the history view of the agent:

- **Window:** A window identifier (IGadgetMyWorkspaceWindowView) <ref name="Windows" />
- **Region:** GadgetMyWorkspaceViewRegion
  - **View:** GadgetMyHistoryWindow (IMyContactHistoryView) <ref name="Contacts" > Module: Genesyslab.Desktop.Modules.Contacts</ref>
    - **Region:** ContactHistoryErrorRegion
      - **View:** ContactHistoryErrorView (ILastAlertErrorView) <ref name="Windows" />
    - **Region:** ContactTabBottomHistoryMultiRegion (Multi-view)
      - **View:** ContactDetailView (IContactDetailView) <ref name="Contacts" > Module: Genesyslab.Desktop.Modules.Contacts</ref>
        - **Region:** ContactDetailRegion
          - **View:** FormerContactDetailContentView (IFormerContactDetailContentView) <ref name="Contacts" />
            - **Region:** ControlListAttachmentRegion

- **View:** AttachmentView (IAttachmentView) (Module: Genesyslab.Desktop.Modules.Windows)
- **View:** CallbackContactDetailContentView (ICallbackContactDetailContentView) <ref name="Callback" >Module: Genesyslab.Desktop.Modules.Callback</ref>  
Added in: 8.1.4
  - **Region:** ContactDetailCallbackInformationRegion
    - **View:** ContactDetailCallbackInformationView (IGenericDataView) <ref name="Windows" />
  - **View:** OpenMediaContactDetailContentView (IContactDetailContentView) <ref name="OpenMedia" > Module: Genesyslab.Desktop.Modules.OpenMedia</ref>
    - **Region:** ControllistAttachmentRegion
      - **View:** AttachmentView (IAttachmentView) <ref name="Windows" > Module: Genesyslab.Desktop.Modules.Windows</ref>
- **View:** NotepadView (INotepadView) <ref name="Windows" > Module: Genesyslab.Desktop.Modules.Windows</ref>
- **View:** ContactHistoryCaseDataView (IStaticCaseDataView) <ref name="Windows" > Module: Genesyslab.Desktop.Modules.Windows</ref>
  - **Region:** StaticCaseDataRegion
    - **View:** StaticCaseDataView (IGenericDataView) <ref name="Windows" > Module: Genesyslab.Desktop.Modules.Windows</ref>
  - **Region:** StaticDispositionCodeRegion
    - **View:** StaticDispositionCodeView (IGenericDataView) <ref name="Windows" > Module: Genesyslab.Desktop.Modules.Windows</ref>



GadgetMyHistoryWindow in the GadgetMyWorkspaceViewRegion (full resolution)

## View GadgetWorkbinsContainerView

Available since: 8.1.0; updated in: 8.1.3, 8.1.4

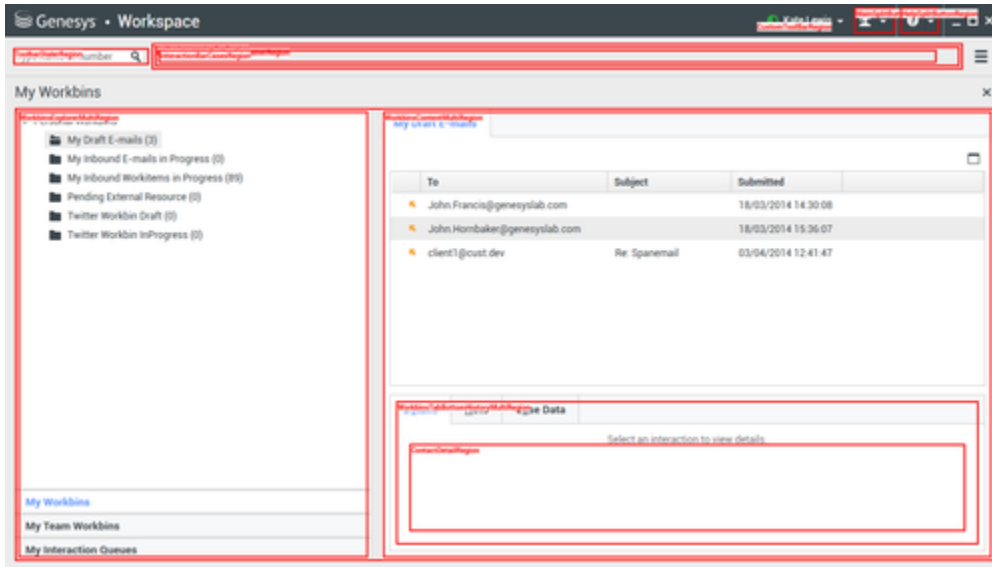
**Description:** Displays the workbins views of the selected workbins:

- **Window:** A window identifier (IGadgetMyWorkspaceWindowView) <ref name="Windows" />
- **Region:** GadgetMyWorkspaceViewRegion
  - **View:** GadgetWorkbinsContainerView (IMainToolBarWorkbinsContainerView) <ref name="OpenMedia" > Module: Genesyslab.Desktop.Modules.OpenMedia</ref>
    - **Region:** WorkbinsExplorerMultiRegion (Multi-view)
      - **View:** WorkbinsExplorerViewMyWorkbinsExplorerView (IWorkbinsExplorerView) <ref name="OpenMedia" /> Modified in 8.1.4
        - **Region:** WorkbinsExplorerViewErrorRegion
          - **View:** WorkbinsExplorerViewErrorView (ILastAlertErrorView) <ref name="Windows" />
    - **View:** MyTeamWorkbinsExplorerView (ITeamLeadWorkbinsExplorerView) <ref name="OpenMedia" />
      - **Region:** MyTeamWorkbinsExplorerViewErrorRegion



- **View:** MyTeamWorkbinsExplorerViewErrorView (ILastAlertErrorView) <ref name="Windows" />
- **View:** MyInteractionFiltersExplorerView (IInteractionFiltersExplorerView) <ref name="OpenMedia" />
- **Region:** WorkbinsContentMultiRegion (Multi-view)
  - **View:** WorkbinsView (IWorkbinsView) <ref name="OpenMedia" />
    - **Region:** WorkbinsViewErrorRegion
      - **View:** WorkbinsViewErrorView (ILastAlertErrorView) <ref name="Windows" />
  - **Region:** WorkbinsTabBottomHistoryMultiRegion (Multi-view)
    - **View:** ContactDetailView (IContactDetailView) (Module: Genesyslab.Desktop.Modules.Contacts)
      - **Region:** ContactDetailRegion
        - **View:** FormerContactDetailContentView (IFormerContactDetailContentView) <ref name="Contacts" > Module: Genesyslab.Desktop.Modules.Contacts</ref>
          - **Region:** ControlListAttachmentRegion
            - **View:** ControlListAttachmentView (IAttachmentView) <ref name="Windows" > Module: Genesyslab.Desktop.Modules.Windows</ref>
        - **View:** CallbackContactDetailContentView (ICallbackContactDetailContentView) <ref name="Callback" /> Added in: 8.1.4
          - **Region:** ContactDetailCallbackInformationRegion
            - **View:** ContactDetailCallbackInformationView (IGenericDataView) <ref name="Windows" />
        - **View:** OpenMediaContactDetailContentView (IOpenMediaContactDetailContentView) <ref name="OpenMedia" > Module: Genesyslab.Desktop.Modules.OpenMedia</ref>
          - **Region:** ControlListAttachmentRegion
            - **View:** ControlListAttachmentView (IAttachmentView) <ref name="Windows" > Module: Genesyslab.Desktop.Modules.Windows</ref>
  - **View:** ContactHistoryNotepadView (INotepadView) <ref name="Windows" />
  - **View:** ContactHistoryCaseDataView (IStaticCaseDataView) <ref name="Windows" />
    - **Region:** StaticCaseDataRegion
      - **View:** StaticCaseDataView (IGenericDataView) <ref name="Windows" > Module: Genesyslab.Desktop.Modules.Windows</ref>

- **Region:** StaticDispositionCodeRegion
  - **View:** StaticDispositionCodeView (IGenericDataView) <ref name="Windows" > Module: Genesyslab.Desktop.Modules.Windows</ref>



GadgetWorkbinsContainerView in GadgetMyWorkspaceViewRegion (full resolution)

## View GadgetMyStatisticsWindow

**Available since:** 8.1.3

**Description:** Displays the agent's statistics.

**Window:** A window identifier (IGadgetMyWorkspaceWindowView) <ref name="Windows" />

- **Region:** GadgetMyWorkspaceViewRegion
  - **View:** **GadgetMyStatisticsWindow**(IMyStatisticsView) <ref name="Statistics">Module: Genesyslab.Desktop.Modules.Statistics </ref>

| My Performance Indicator           | Personal | Agent Group G3 | Agent Group 80001 | Agent Group Everybody | Web-Callback distribution for processing |
|------------------------------------|----------|----------------|-------------------|-----------------------|------------------------------------------|
| Total number inbound calls (Voice) | 0%       |                |                   |                       |                                          |
| Total number internal calls        | 0%       |                |                   |                       |                                          |
| Total number outbound calls        | 0%       | 0              | 0                 | 0                     | 0                                        |
| Total Number of Refused Calls      | 100%     | 0              | 0                 | 0                     | 0                                        |
| Total Number of Transferred Calls  | 0%       | 0              | 0                 | 0                     | 0                                        |
| Total login time                   | 00:00:29 |                |                   |                       |                                          |
| Current Ready duration             | 00:00:00 |                |                   |                       |                                          |
| Current Wrap up duration           | 00:00:00 |                |                   |                       |                                          |
| Current talk duration              | 00:11:47 |                |                   |                       |                                          |
| Current hold duration              | 00:00:00 |                |                   |                       |                                          |

GadgetMyStatisticsWindow in GadgetMyWorkspaceViewRegion

## View GadgetCallCenterStatisticsWindow

Available since: 8.1.3

**Description:** Displays the Call Center Statistics panel: **Window:** A window identifier (IGadgetMyWorkspaceWindowView) (Module: Genesyslab.Desktop.Modules.Windows)

- **Region:** GadgetMyWorkspaceViewRegion
  - **View: GadgetCallCenterStatisticsWindow**(IMyCallCenterStatisticsView) <ref name="Statistics" />



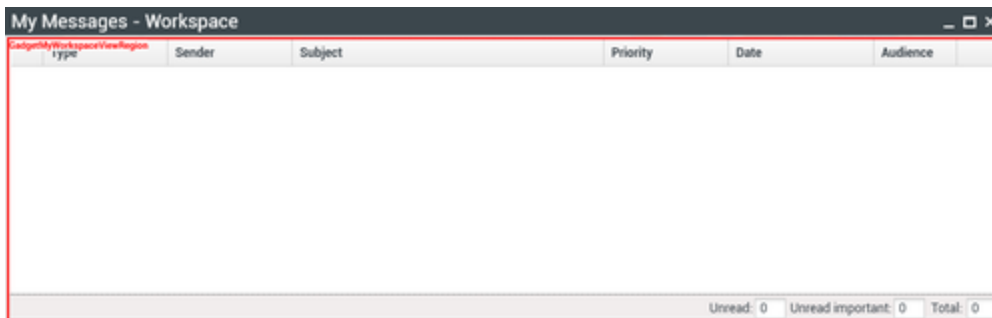
GadgetCallCenterStatisticsWindow in GadgetMyWorkspaceViewRegion

## View GadgetMyMessages

Available since: 8.0.1

**Description:** Displays the agent's messages.

- **Window:** A window identifier (IGadgetMyWorkspaceWindowView) <ref name="Windows" />
  - **Region:** GadgetMyWorkspaceViewRegion
    - **View: GadgetMyMessagesWindow** (IMyBroadcastMessageView) <ref name="Windows" />



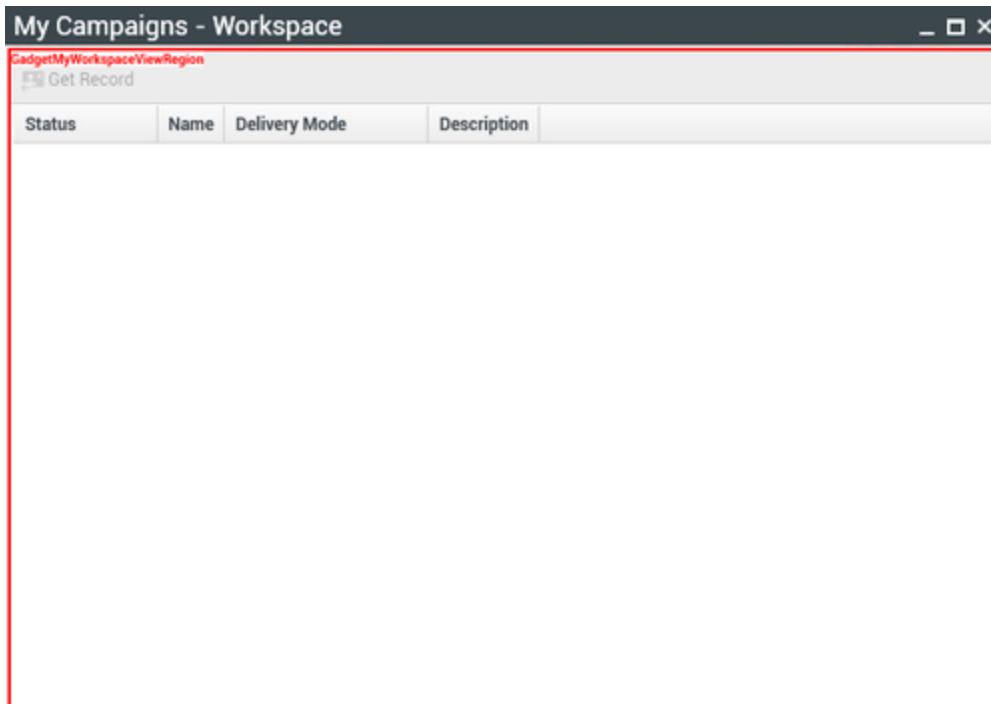
GadgetMyMessagesWindow in GadgetMyWorkspaceViewRegion

## View GadgetMyCampaignsWindow

Available since: 8.1.1

**Description:** Displays the campaign's messages.

- **Window:** A window identifier (IGadgetMyWorkspaceWindow) <ref name="Windows" />
  - **Region:** GadgetMyWorkspaceViewRegion
    - **View:** GadgetMyCampaignsWindow (IMyCampaignsView) <ref name="Outbound">Module: Genesyslab.Desktop.Modules.Outbound</ref>
      - **Region:** CampaignsErrorRegion
        - **View:** campaignsErrorView (ILastAlertErrorView) <ref name="Windows" />



GadgetMyCampaignsWindow

## View GadgetContactContainerView

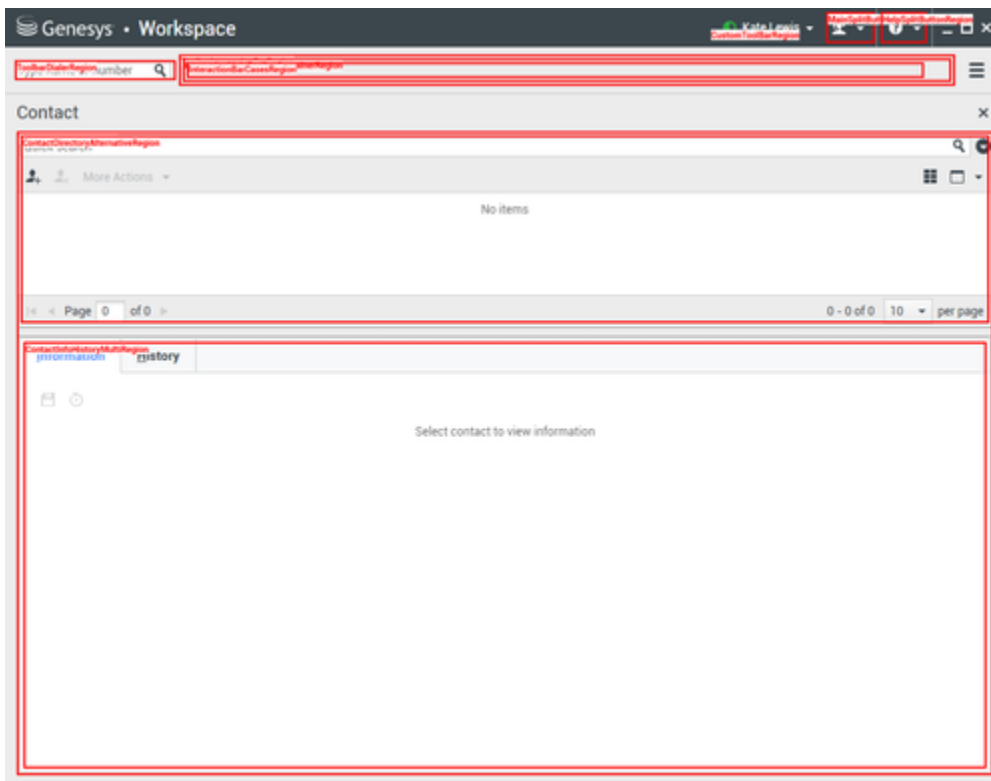
Added in 8.5

**Description:** Displays the contact directory.

- **Window:** GadgetMyWorkspaceWindow (IGadgetMyWorkspaceWindow) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>
  - **Region:** GadgetMyWorkspaceViewRegion
    - **View:** GadgetContactContainerView (IMainToolbarContainerView) <ref name="Contacts">Genesyslab.Desktop.Modules.Contacts</ref>
      - **Region:** ContactDirectoryMultiRegion (Multi-view)
        - **View:** ContactDirectoryView (IContactDirectoryView) <ref name="Contacts">Genesyslab.Desktop.Modules.Contacts</ref>

- **Region:** ContactDirectoryErrorRegion
  - **View:** ContactDirectoryErrorView (ILastAlertErrorView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>
- **Region:** ContactDirectoryAlternativeRegion  
Added in 8.5
  - **View:** ContactDirectoryView (IContactDirectoryView) <ref name="Contacts" />
  - **Region:** ContactDirectoryErrorRegion
    - **View:** ContactDirectoryErrorView (ILastAlertErrorView) <ref name="Windows" />
- **Region:** ContactInfoHistoryMultiRegion (Multi-view)
  - **View:** ContactInformationView (IContactInformationView) <ref name="Contacts">Genesyslab.Desktop.Modules.Contacts</ref>
  - **Region:** ContactInformationErrorRegion
    - **View:** ContactInformationErrorView (ILastAlertErrorView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>
- **View:** ContactHistoryView (IMyContactHistoryView) <ref name="Contacts">Genesyslab.Desktop.Modules.Contacts</ref>
  - **Region:** ContactHistoryErrorRegion
    - **View:** ContactHistoryErrorView (ILastAlertErrorView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>
- **Region:** ContactTabBottomHistoryMultiRegion (Multi-view)
  - **View:** ContactDetailView (IContactDetailView) <ref name="Contacts">Genesyslab.Desktop.Modules.Contacts</ref>
    - **Region:** ContactDetailRegion
      - **View:** FormerContactDetailContentView (IFormerContactDetailContentView) <ref name="Contacts">Genesyslab.Desktop.Modules.Contacts</ref>
        - **Region:** ControlListAttachmentRegion
          - **View:** AttachmentView ControlListAttachmentView (IAttachmentView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>
    - **View:** CallbackContactDetailContentView (ICallbackContactDetailContentView) <ref name="Callback" />  
Added in: 8.1.4
      - **Region:** ContactDetailCallbackInformationRegion
        - **View:** ContactDetailCallbackInformationView (IGenericDataView) <ref name="Windows" />
  - **View:** OpenMediaContactDetailContentView (IOpenMediaContactDetailContentView) <ref name="OpenMedia">Genesyslab.Desktop.Modules.OpenMedia</ref>

- **Region:** ControlListAttachmentRegion
  - **View:** AttachmentView ControlListAttachmentView (AttachmentView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>
- **View:** NotepadViewContactHistoryNotepadView (INotepadView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>
- **View:** ContactHistoryCaseDataView (IStaticCaseDataView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>
  - **Region:** StaticCaseDataRegion
    - **View:** StaticCaseDataView (IGenericDataView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>
  - **Region:** StaticDispositionCodeRegion
    - **View:** StaticDispositionCodeView (IGenericDataView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>



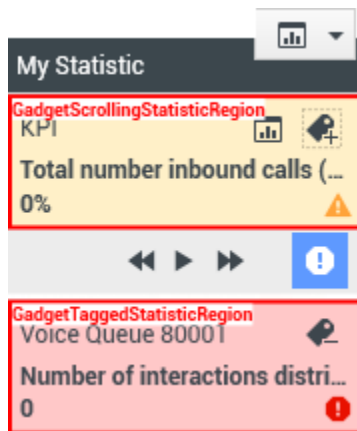
(full resolution)

## Window GadgetStatisticsWindow

Available since: 8.1.0

**Description:** This gadget window displays a specific GUI for agent statistics.

- **Window:** GadgetStatisticsWindow (IGadgetStatisticsWindow) <ref name="Statistics" >Module: Genesyslab.Desktop.Modules.Statistics</ref>
  - **Region:** GadgetScrollingStatisticRegion
    - **View:** GadgetSingleStatView (IGadgetSingleStatView) <ref name="Statistics" />
  - **Region:** GadgetTaggedStatisticRegion (Multi-view)
    - **View:** GadgetSingleStatView (IGadgetSingleStatView) <ref name="Statistics" />
    - **View:** GadgetSingleStatView (IGadgetSingleStatView) <ref name="Statistics" />



GadgetStatisticsWindow

## References

# Toaster Window

## Tip

If you want to highlight the functional areas in the Workspace UI with the red boxes and descriptions shown in the screen shots in this topic, on a *non-production* developer workstation, either configure the value of the `keyboard.shortcut.action.test.hidden` developer option in the `interactionworkspace.exe.config` file or in Configuration Server using GAX, to a keyboard short cut, such as: `Ctrl+P`. When you press the keyboard shortcut, the functional areas are highlighted.

Genesys recommends that you set this option at the agent level for testing purposes before setting it higher up in the option hierarchy or in the `interactionworkspace.exe.config` file.

Upon execution of the file, the following files are generated in the execution folder:

- `DumpChainOfCommandsInHTML.htm` containing all chains of commands available in the product in html format.
- `DumpChainOfCommandsInWikiMarkup.txt` containing all chains of commands available in the product in wiki format.
- `DumpKeyboardShortcut.htm` containing the list of all commands that can be triggered by a shortcut, and the shortcut configured if any.
- `DumpViewTree.htm` containing all regions.
- `DumpFullTreeView.htm` (when full tree view is specified) containing view tree names.



**Purpose:** To present for each window, the list of default views and regions that your application can customize.

## Window ToasterMessageWindow

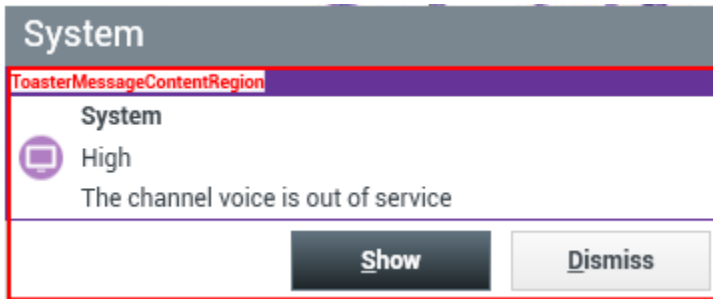
Available since: 8.1.0; updated in 8.1.4

**Description:** The Toaster Message Window displays the business-type or system-type messages that your application receives. The `ToasterMessageWindow` window identifier is available since 8.1.4.

- **Window:** `ToasterMessageWindow (IToasterMessageWindow)` <ref name="Windows">Module: `Genesyslab.Desktop.Modules.Windows`</ref>
  - **Region:** `ToasterMessageContentRegion`
    - **View:** `ToasterBroadcastMessageCampaignContentView (IToasterBroadcastMessageCampaignContentView)` <ref name="Outbound">Module: `Genesyslab.Desktop.Modules.Outbound`</ref>
    - **View:** `ToasterBroadcastMessageContentView (IToasterBroadcastMessageContentView)` <ref



```
name="Windows" />
```



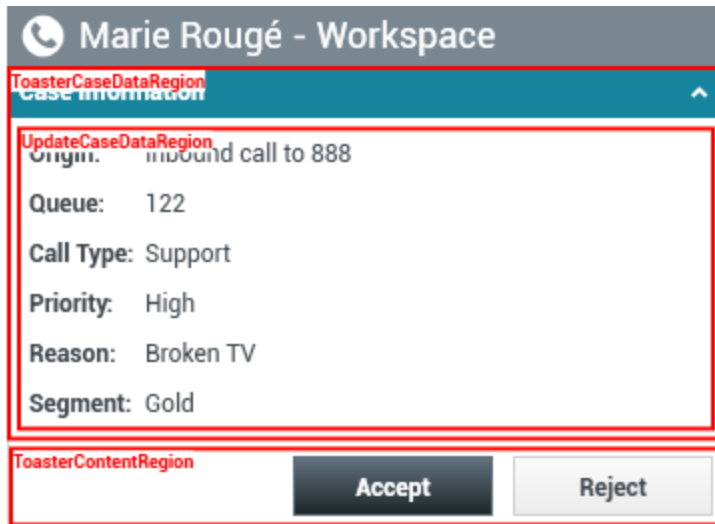
System message in a Toaster Window

## Window ToasterWindow

Available since: 8.1.0; updated in 8.1.4

**Description:** The toaster window displays light information about the current case in progress. The ToasterWindow window identifier is available since 8.1.4.

- **Window:** ToasterWindow (IToasterWindow) <ref name="Windows" />
  - **Region:** ToasterCaseDataRegion (Multi-view)
    - **View:** CaseDataView (ICaseDataView) <ref name="Windows" />
      - **Region:** UpdateCaseDataRegion
        - **View:** UpdateCaseDataView (IGenericDataView) <ref name="Windows" >Module: Genesyslab.Desktop.Modules.Windows</ref>
  - **Region:** ToasterContentRegion
    - **View:** ToasterOpenMediaContent (IToasterOpenMediaContentView) <ref name="OpenMedia" >Module: Genesyslab.Desktop.Modules.OpenMedia</ref>
    - **View:** ToasterVoiceContentView (IToasterVoiceContentView) <ref name="Voice" >Module: Genesyslab.Desktop.Modules.Voice</ref>
    - **View:** ToasterPullPreviewContentView (IToasterPullPreviewContentView) <ref name="Outbound" >Module: Genesyslab.Desktop.Modules.Outbound</ref>



ToasterOpenMediaContent and ToasterVoiceContent in the IToasterWindow

## References

# Toolbar Window

## Tip

If you want to highlight the functional areas in the Workspace UI with the red boxes and descriptions shown in the screen shots in this topic, on a *non-production* developer workstation, either configure the value of the `keyboard.shortcut.action.test.hidden` developer option in the `interactionworkspace.exe.config` file or in Configuration Server using GAX, to a keyboard short cut, such as: `Ctrl+P`. When you press the keyboard shortcut, the functional areas are highlighted.

Genesys recommends that you set this option at the agent level for testing purposes before setting it higher up in the option hierarchy or in the `interactionworkspace.exe.config` file.

Upon execution of the file, the following files are generated in the execution folder:

- `DumpChainOfCommandsInHTML.htm` containing all chains of commands available in the product in html format.
- `DumpChainOfCommandsInWikiMarkup.txt` containing all chains of commands available in the product in wiki format.
- `DumpKeyboardShortcut.htm` containing the list of all commands that can be triggered by a shortcut, and the shortcut configured if any.
- `DumpViewTree.htm` containing all regions.
- `DumpFullTreeView.htm` (when full tree view is specified) containing view tree names.



**Purpose:** To present for each window, the list of default views and regions that your application can customize.

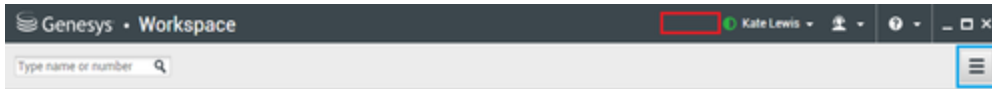
## Window: ToolbarWindow

**Available since:** 8.0.1; **updated in:** 8.1.0, 8.1.1, 8.1.2, 8.1.3, 8.1.4, 8.5

The `ToolbarWindow` (`IToolbarWindow`) [<ref name="Windows" >Module](#):

`Genesyslab.Desktop.Modules.Windows` is the top-level window which contains predefined buttons in the `ToolbarWorksheetButtonRegion` associated with specific working views. This toolbar contains views and regions which are activated depending on the user's input by clicking expandable button for instance.

In 8.5, you can include Custom Modules and Views in the `CustomToolBarRegion` area. Custom modules can no longer extend the drop-down menus of the former `Workspace` button through the `WorkspaceMenuRegion` region. Instead, you should use the `ToolbarHamburgerButtonRegion` area to add menu items in the hamburger menu along with the out of the box items of this area, such as My Channels, My Campaigns, and so on.



The `ToolBarWindow`; no working view is deployed. In blue, you can see `ToolBarHamburgerButtonRegion`. In red, `CustomToolBarRegion`.

See also [Changes in the Main Window Regions](#) section.

The components available in this toolbar are the following views and regions:

- **Window:** `ToolBarWindow` (`IToolbarWindow`) `<ref name="Windows" />`
  - **Region:** `CustomToolBarRegion` (Multi-view)  
Added in 8.5  
**View:** `MainToolBarMonitorView` (`IMainToolBarMonitorView`) `<ref name="ActiveRecording" />`
  - **Region:** `MainSplitButtonRegion` (Multi-view)
    - **View:** `MainButtonView` (`IMainButtonView`) `<ref name="Windows" >Module: Genesyslab.Desktop.Modules.Windows</ref>`
    - **View:** `MainToolBarMonitorView` (`IMainToolBarMonitorView`) `<ref name="ActiveRecording" >Module: Genesyslab.Desktop.Modules.ActiveRecording</ref>`  
Deprecated in 8.5
  - **Region:** `HelpSplitButtonRegion`
    - **View:** `ButtonHelpView` (`IButtonHelpView`) `<ref name="Windows" >Module: Genesyslab.Desktop.Modules.Windows</ref>`
  - **Region:** `ToolBarDialerRegion`
    - **View:** `ToolBarDialerView` (`ITeamCommunicatorView`) `<ref name="TeamCommunicator" >Module: Genesyslab.Desktop.Modules.TeamCommunicator</ref>`
  - **Region:** `ToolBarInteractionBarRegion`
    - **View:** `InteractionMainToolBarContainerView` (`IInteractionMainToolBarContainerView`) `<ref name="InteractionBar" >Module: Genesyslab.Desktop.Modules.InteractionBar</ref>`
      - **Region:** `InteractionMainToolBarContainerRegion`
        - **View:** `InteractionBar` (`IInteractionBarView`) `<ref name="InteractionBar" >Module: Genesyslab.Desktop.Modules.InteractionBar</ref>`
          - **Region:** `InteractionBarCasesRegion` (Multi-view)
  - ~~**Region:** `ToolBarStatusRegion`  
Deprecated in 8.5~~
    - ~~**View:** `AlertMessageView` (`IAlertMessageView`) `<ref name="Windows" />`~~
      - ~~**Region:** `MyMessageRegion`~~
        - ~~**View:** `MyMessagesView` (`IMyBroadcastMessageView`) `<ref name="Windows" />`~~
  - **Region:** `ToolBarWorksheetButtonRegion`  
Added in 8.5
  - **Region:** `ToolBarWorksheetRegion`



- 
- **View:** DonePullPreviewToolBarView (IDonePullPreviewToolBarView) <ref name="Outbound" />
  - **View:** RescheduleToolBarView (IRescheduleToolBarView) <ref name="Outbound" />
  - **View:** InteractionVoiceToolBarView (IInteractionVoiceToolBarView) (Module: Genesyslab.Desktop.Modules.Voice)
    - **Region:** InteractionVoiceCustomButtonRegion (Multi-view)
      - **View:** SIPEndpointSoundView (ISIPEndpointSoundView) (Module: Genesyslab.Desktop.Modules.SIPEndpointCommunication)
      - **View:** SIPEndpointVideoView (ISIPEndpointVideoView) (Module: Genesyslab.Desktop.Modules.SIPEndpointCommunication)
      - **View:** SIPMonitor (ISIPMonitor) (Module: Genesyslab.Desktop.Modules.ActiveRecording)
      - **View:** IxnMonitor (IIxnMonitor) <ref name="ActiveRecording">Genesyslab.Desktop.Modules.ActiveRecording</ref>
  - **View:** InteractionIMToolBarView (IInteractionIMToolBarView) (Module: Genesyslab.Desktop.Modules.Voice)
  - **View:** InteractionChatToolBarView (IInteractionChatToolBarView) <ref name="OpenMedia" />
  - **View:** InteractionChatConsultToolBarView (IInteractionChatConsultToolBarView) <ref name="OpenMedia" />
  - **Region:** InteractionBarBundleToolBarRegion (Multi-view)
    - **View:** TeamLeadToolBarView (ITeamLeadToolBarView) <ref name="TeamLead" />
    - **View:** BundleToolBarView (IBundleToolBarView) (Module: Genesyslab.Desktop.Modules.Windows)
      - **Region:** BundleCustomButtonRegion (Multi-view)
        - **View:** InteractionCallbackRescheduleToolBarView (IInteractionCallbackRescheduleToolBarView) <ref name="Callback" />
        - **View:** DonePullPreviewToolBarView (IDonePullPreviewToolBarView) <ref name="Outbound" />
        - **View:** RescheduleToolBarView (IRescheduleToolBarView) <ref name="Outbound" />
  - **Region:** InteractionBarConsultationBundlesRegion (Multi-view)
    - **View:** InteractionBarConsultationBundle (IInteractionBarConsultationBundleView) <ref name="InteractionBar" />
      - **Region:** InteractionBarInteractionsRegion (Multi-view)
        - **View:** InteractionBarInteraction (IInteractionBarInteractionView) <ref name="

---

"InteractionBar" />

- **Region:** BundleToolBarContainerRegion (Multi-view)
  - **View:** TeamLeadToolBarView (ITeamLeadToolBarView) <ref name="TeamLead" />
  - **View:** InteractionPreviewToolBarView (IInteractionPreviewToolBarView) <ref name="Outbound" />
  - **View:** InteractionCallbackToolBarView (IInteractionCallbackToolBarView) <ref name="Callback" />
  - **View:** InteractionSmsToolBarView (IInteractionSmsToolBarView) <ref name="OpenMedia" />
  - **View:** InteractionWorkItemToolBarView (IInteractionWorkItemToolBarView) <ref name="OpenMedia" />
    - **Region:** CustomWorkItemToolBarRegion (Multi-view)
  - **View:** InteractionOutboundEmailToolBarView (IInteractionOutboundEmailToolBarView) <ref name="OpenMedia" />
  - **View:** InteractionInboundEmailToolBarView (IInteractionInboundEmailToolBarView) <ref name="OpenMedia" />
  - **View:** BundleToolBarView (IBundleToolBarView) (Module: Genesyslab.Desktop.Modules.Windows)
    - **Region:** BundleCustomButtonRegion (Multi-view)
      - **View:** InteractionCallbackRescheduleToolBarView (IInteractionCallbackRescheduleToolBarView) <ref name="Callback" />
      - **View:** DonePullPreviewToolBarView (IDonePullPreviewToolBarView) <ref name="Outbound" />
      - **View:** RescheduleToolBarView (IRescheduleToolBarView) <ref name="Outbound" />
  - **View:** InteractionVoiceToolBarView (IInteractionVoiceToolBarView) (Module: Genesyslab.Desktop.Modules.Voice)
    - **Region:** InteractionVoiceCustomButtonRegion (Multi-view)
      - **View:** SIPEndpointSoundView (ISIPEndpointSoundView) (Module: Genesyslab.Desktop.Modules.SIPEndpointCommunication)
      - **View:** SIPEndpointVideoView (ISIPEndpointVideoView) (Module: Genesyslab.Desktop.Modules.SIPEndpointCommunication)
      - **View:** SIPMonitor (ISIPMonitor) (Module: Genesyslab.Desktop.Modules.ActiveRecording)
      - **View:** IxnMonitor (IIxnMonitor) <ref name="ActiveRecording">Genesyslab.Desktop.Modules.ActiveRecording</ref>
  - **View:** InteractionIMToolBarView (IInteractionIMToolBarView) (Module: Genesyslab.Desktop.Modules.Voice)

- **View:** InteractionChatToolBarView (IInteractionChatToolBarView) <ref name="OpenMedia" />
- **View:** InteractionChatConsultToolBarView (IInteractionChatConsultToolBarView) <ref name="OpenMedia" />
- **Region:** InteractionBarBundleToolBarRegion (Multi-view)
  - **View:** TeamLeadToolBarView (ITeamLeadToolBarView) <ref name="TeamLead" />
  - **View:** BundleToolBarView (IBundleToolBarView) (Module: Genesyslab.Desktop.Modules.Windows)
    - **Region:** BundleCustomButtonRegion (Multi-view)
      - **View:** InteractionCallbackRescheduleToolBarView (IInteractionCallbackRescheduleToolBarView) <ref name="Callback" />
      - **View:** DonePullPreviewToolBarView (IDonePullPreviewToolBarView) <ref name="Outbound" />
      - **View:** RescheduleToolBarView (IRescheduleToolBarView) <ref name="Outbound" />

## Multi-view MyWorkplaceContainerView

Available since: 8.0.1

Updated in: 8.1.0, 8.1.1, 8.1.2

The multi-view MyWorkPlace is part of the ToolbarWorksheetRegion region and includes the following regions and views:

- **View:** MyWorkplaceContainerView (IMyWorkplaceContainerView) <ref name="Windows" />
  - **Region:** ToolbarWorkplaceRegion (Multi-view)
    - **View:** MyPlaceStatusView (IMyPlaceStatusView) <ref name="Windows" />
    - **View:** myCampaignsListView (IMyCampaignsView) <ref name="Outbound" /> Module: Genesyslab.Desktop.Modules.Outbound</ref>
      - **Region:** CampaignsErrorRegion
        - **View:** campaignsErrorView (ILastAlertErrorView) <ref name="Windows" />
    - **View:** MyContactHistory (IMyContactHistoryView) <ref name="Contacts" />
      - **Region:** ContactHistoryErrorRegion
        - **View:** ContactHistoryErrorView (ILastAlertErrorView) <ref name="Windows" />
      - **Region:** ContactTabBottomHistoryMultiRegion (Multi-view)
        - **View:** ContactDetailView (IContactDetailView) <ref name="Contacts" />
          - **Region:** ContactDetailRegion
            - **View:** OpenMediaContactDetailContentView (IOpenMediaContactDetailContentView) <ref name="Contacts">Module:



Genesyslab.Desktop.Modules.Contacts</ref>

- **Region:** ControllistAttachmentRegion
  - **View:** ControllistAttachmentView (IAttachmentView) <ref name="Windows" />
- **View:** CallbackContactDetailContentView (ICallbackContactDetailContentView) <ref name="Callback" />
  - **Region:** ContactDetailCallbackInformationRegion
    - **View:** ContactDetailCallbackInformationView (IGenericDataView) <ref name="Windows" />
- **View:** FormerContactDetailContentView (IFormerContactDetailContentView) <ref name="Contacts">Module: Genesyslab.Desktop.Modules.Contacts</ref>
  - **Region:** ControllistAttachmentRegion
    - **View:** ControllistAttachmentView (IAttachmentView) <ref name="Windows" />
- **View:** ContactHistoryNotepadView(INotepadView) <ref name="Windows" />
- **View:** ContactHistoryCaseDataView (IStaticCaseDataView) <ref name="Windows" />
  - **Region:** StaticCaseDataRegion
    - **View:** StaticCaseDataView (IGenericDataView) <ref name="Windows">Module: Genesyslab.Desktop.Modules.Windows</ref>
  - **Region:** StaticDispositionCodeRegion
    - **View:** StaticDispositionCodeView (IGenericDataView) <ref name="Windows">Module: Genesyslab.Desktop.Modules.Windows</ref>
- **View:** MyStatisticsView (IMyStatisticsView) <ref name="Statistics">Genesyslab.Desktop.Modules.Statistics</ref>
- **View:** MyCallCenterStatisticsView (IMyCallCenterStatisticsView) <ref name="Statistics" />

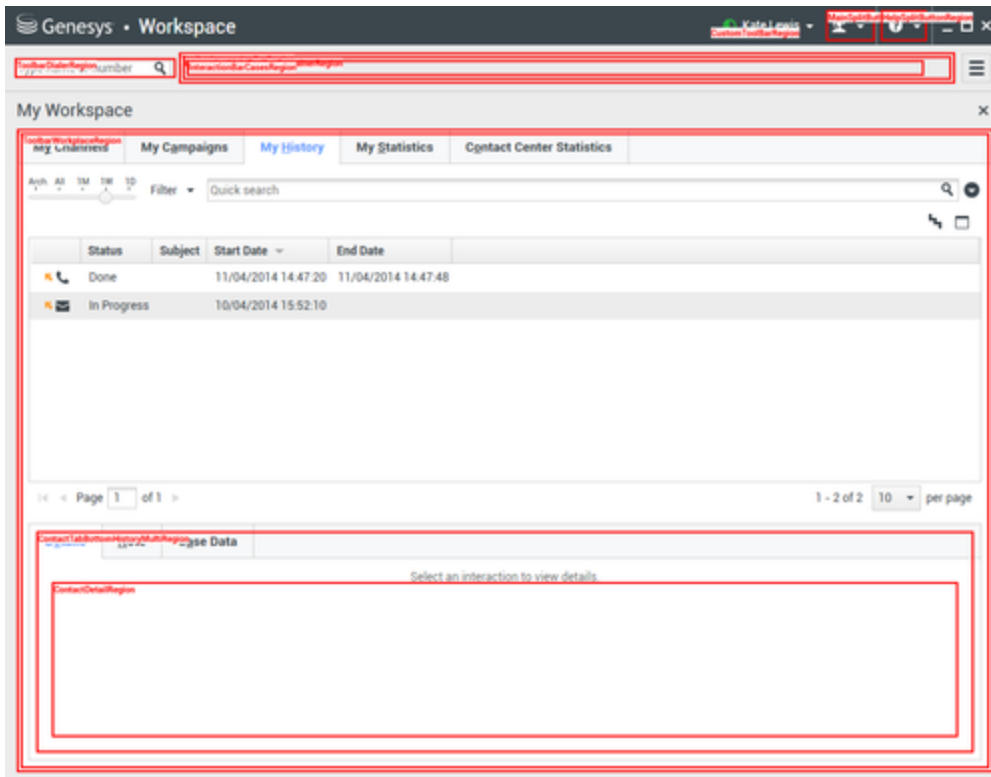
## Important

See the [Gadget Windows](#) for screenshots of [MyStatisticsView](#) and [MyCallCenterStatisticsView](#).

## Example MyContactHistory

In the following example, you can observe the views and regions which are visible when:

1. The Place Status Menu is selected in the ToolbarWorksheetButtonRegion.
2. The *MyHistory* tab is selected and makes visible the MyContactHistory.



MyContactHistory in IMyWorkplaceView (full resolution)

The views and regions which are visible in this screenshot are the following:

- **Window:** ToolbarWindow (IToolbarWindow) <ref name="Windows" />
  - **Region:** MainSplitButtonRegion (Multi-view)
    - **View:** MainButtonView (IMainButtonView) <ref name="Windows" >Module: Genesyslab.Desktop.Modules.Windows</ref>
  - **Region:** HelpSplitButtonRegion
    - **View:** ButtonHelpView (IButtonHelpView) <ref name="Windows" >Module: Genesyslab.Desktop.Modules.Windows</ref>
  - **Region:** ToolbarDialerRegion
    - **View:** ToolbarDialerView (ITeamCommunicatorView) <ref name="TeamCommunicator" />
  - **Region:** ToolbarWorksheetButtonRegion (Multi-view)
    - **View:** MyWorkplaceButtonView (IMyWorkplaceButtonView) <ref name="Windows" />
      - **Region:** WorkspaceMenuRegion
        - **View:** MyHistoryMenu (IHistoryMenuView)
  - **Region:** ToolbarStatusRegion
  - **Region:** ToolbarWorksheetRegion

- **View:** MyWorkplaceContainerView (IMyWorkplaceContainerView) <ref name="Windows" />
  - **Region:** ToolbarWorkplaceRegion (Multi-view)
    - **View:** MyContactHistory (IMyContactHistoryView) <ref name="Contacts" />
      - **Region:** ContactHistoryErrorRegion
        - **View:** ContactHistoryErrorView (ILastAlertErrorView) <ref name="Windows" />
    - **Region:** ContactTabBottomHistoryMultiRegion (Multi-view)
      - **View:** ContactDetailView (IContactDetailView) <ref name="Contacts" />
        - **Region:** ContactDetailRegion
          - **View:** OpenMediaContactDetailContentView (IOpenMediaContactDetailContentView) <ref name="Contacts">Module: Genesyslab.Desktop.Modules.Contacts</ref>
            - **Region:** ControlListAttachmentRegion
              - **View:** ControlListAttachmentView (IAttachmentView) <ref name="Windows" />
          - **View:** CallbackContactDetailContentView (ICallbackContactDetailContentView) <ref name="Callback" />
            - **Region:** ContactDetailCallbackInformationRegion
              - **View:** ContactDetailCallbackInformationView (IGenericDataView) <ref name="Windows" />
          - **View:** FormerContactDetailContentView (IFormerContactDetailContentView) <ref name="Contacts">Module: Genesyslab.Desktop.Modules.Contacts</ref>
            - **Region:** ControlListAttachmentRegion
              - **View:** ControlListAttachmentView (IAttachmentView) <ref name="Windows" />

## Multi-view MainToolbarContainerView

Updated in 8.5

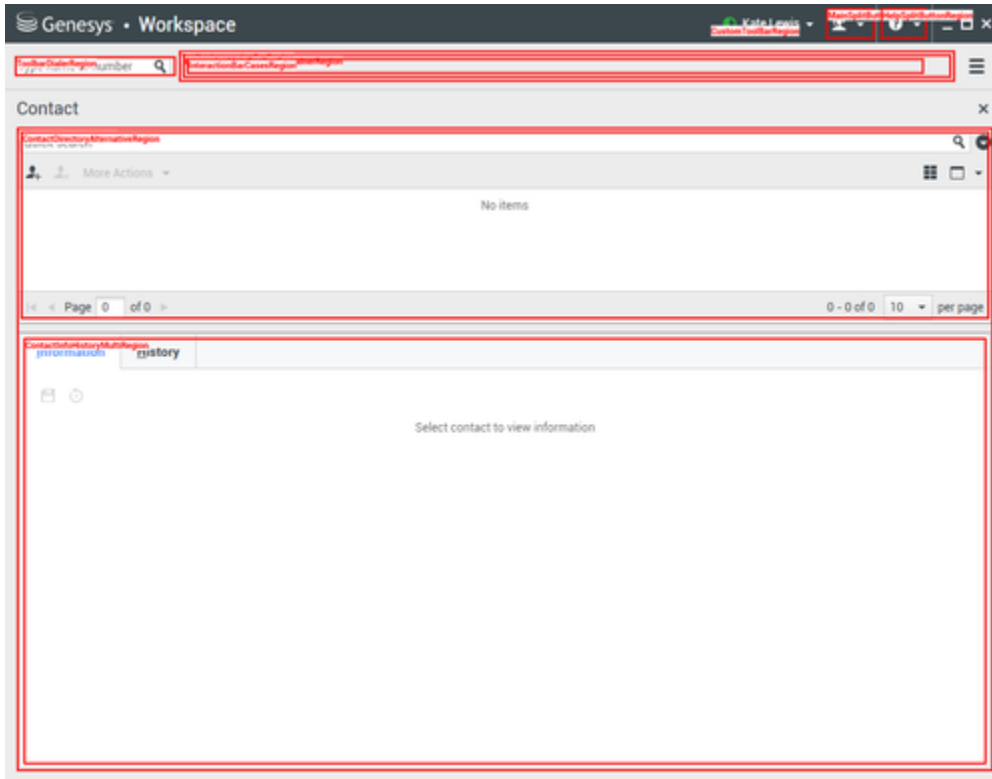
The multi-view MainToolbarWorkbinsContainerView is part of the ToolbarWorksheetRegion region, which includes the following regions and views:

- **View:** MainToolbarContainerView (IMainToolbarContainerView) <ref name="Contacts" />
  - **Region:** ContactDirectoryMultiRegion (Multi-view)
    - **View:** ContactDirectoryView (IContactDirectoryView) <ref name="Contacts" />
      - **Region:** ContactDirectoryErrorRegion
        - **View:** ContactDirectoryErrorView (ILastAlertErrorView) <ref name="Windows" />

- **Region:** ContactDirectoryAlternativeRegion  
**Added** in 8.5
  - **View:** ContactDirectoryView (IContactDirectoryView) <ref name="Contacts" />
    - **Region:** ContactDirectoryErrorRegion
      - **View:** ContactDirectoryErrorView (ILastAlertErrorView) <ref name="Windows" />
- **Region:** ContactInfoHistoryMultiRegion (Multi-view)
  - **View:** ContactInformationView (IContactInformationView) <ref name="Contacts" />
    - **Region:** ContactInformationErrorRegion
      - **View:** ContactInformationErrorView (ILastAlertErrorView) <ref name="Windows" />
  - **View:** ContactHistoryView (IMyContactHistoryView) <ref name="Contacts" />
    - **Region:** ContactHistoryErrorRegion
      - **View:** ContactHistoryErrorView (ILastAlertErrorView) <ref name="Windows" />
  - **Region:** ContactTabBottomHistoryMultiRegion (Multi-view)
    - **View:** ContactDetailView (IContactDetailView) <ref name="Contacts" />
      - **Region:** ContactDetailRegion
        - **View:** FormerContactDetailContentView FormerContactDetailContentView (IFormerContactDetailContentView) <ref name="Contacts">Module: Genesyslab.Desktop.Modules.Contacts</ref>
          - **Region:** ControllistAttachmentRegion
            - **View:** ControllistAttachmentView (IAttachmentView) <ref name="Windows" />
        - **View:** CallbackContactDetailContentView (ICallbackContactDetailContentView) <ref name="Callback" />
          - **Region:** ContactDetailCallbackInformationRegion
            - **View:** ContactDetailCallbackInformationView (IGenericDataView) <ref name="Windows" />
        - **View:** OpenMediaContactDetailContentView (IOpenMediaContactDetailContentView) <ref name="OpenMedia">Module: Genesyslab.Desktop.Modules.OpenMedia</ref>
          - **Region:** ControllistAttachmentRegion
            - **View:** AttachmentView ControllistAttachmentView (IAttachmentView) <ref name="Windows" />
    - **View:** ContactHistoryNotepadView(INotepadView) <ref name="Windows" />
    - **View:** ContactHistoryCaseDataView (IStaticCaseDataView) <ref name="Windows" />
      - **Region:** StaticCaseDataRegion
        - **View:** StaticCaseDataView (IGenericDataView) <ref name="Windows">Module: Genesyslab.Desktop.Modules.Windows</ref>

- **Region:** StaticDispositionCodeRegion
  - **View:** StaticDispositionCodeView (IGenericDataView) <ref name="Windows" >Module: Genesyslab.Desktop.Modules.Windows</ref>

The following screenshot shows the MainToolBarContainerView and its nested views and regions (but not all of them):



The ContactDirectoryView view, opened from the Hamburger menu.(full resolution)

### Multi-view MainToolBarWorkbinsContainerView

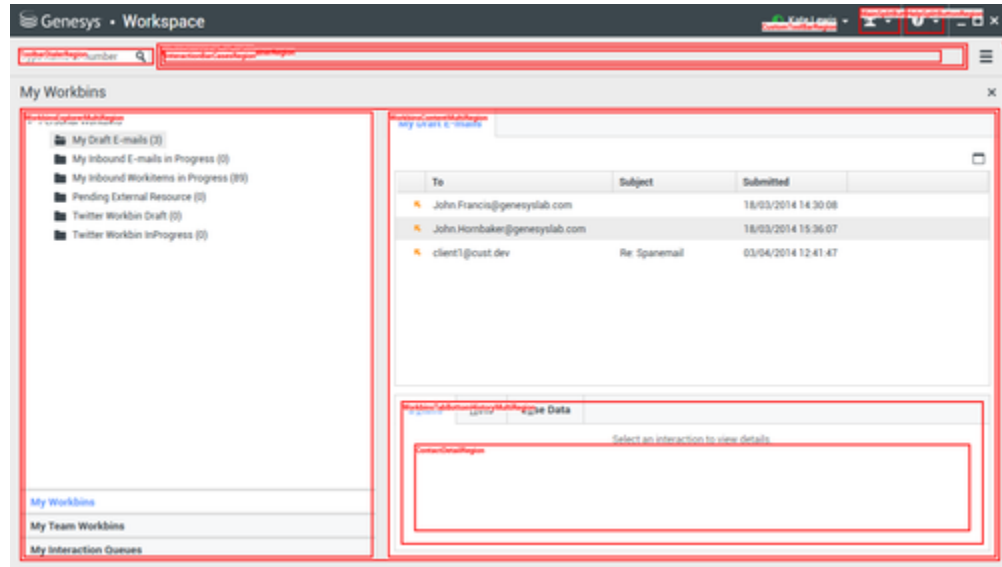
The multi-view MainToolBarWorkbinsContainerView is part of the ToolBarWorksheetRegion region, which includes the following regions and views:

- **View:** MainToolBarWorkbinsContainerView (IMainToolBarWorkbinsContainerView) <ref name="OpenMedia" />
  - **Region:** WorkbinsExplorerMultiRegion (Multi-view)
    - **View:** MyWorkbinsExplorerView (IWorkbinsExplorerView) <ref name="OpenMedia" />
      - **Region:** WorkbinsExplorerViewErrorRegion
        - **View:** WorkbinsExplorerViewErrorView (ILastAlertErrorView) <ref name="Windows" />
  - **View:** MyTeamWorkbinsExplorerView (ITeamLeadWorkbinsExplorerView) <ref name="OpenMedia" />

- **Region:** MyTeamWorkbinsExplorerViewErrorRegion
  - **View:** MyTeamWorkbinsExplorerViewErrorView (ILastAlertErrorView) <ref name="Windows" />
- **View:** MyInteractionFiltersExplorerView (IInteractionFiltersExplorerView) <ref name="OpenMedia" />
- **Region:** WorkbinsContentMultiRegion (Multi-view)
  - **View:** WorkbinsView (IWorkbinsView) <ref name="OpenMedia" />
    - **Region:** WorkbinsViewErrorRegion
      - **View:** WorkbinsViewErrorView (ILastAlertErrorView) <ref name="Windows" />
    - **Region:** WorkbinsTabBottomHistoryMultiRegion (Multi-view)
      - **View:** ContactDetailView (IContactDetailView) <ref name="Contacts" />
        - **Region:** ContactDetailRegion
          - **View:** CallbackContactDetailContentView (ICallbackContactDetailContentView) <ref name="Callback" />
            - **Region:** ContactDetailCallbackInformationRegion
              - **View:** ContactDetailCallbackInformationView (IGenericDataView) <ref name="Windows" />
          - **View:** FormerContactDetailContentView (IOpenMediaContactDetailContentView) <ref name="Contacts">Module: Genesyslab.Desktop.Modules.Contacts</ref>
            - **Region:** ControlListAttachmentRegion
              - **View:** ControlListAttachmentView (IAttachmentView) <ref name="Windows" />
      - **View:** OpenMediaContactDetailContentView (IContactDetailContentView) <ref name="OpenMedia">Module: Genesyslab.Desktop.Modules.OpenMedia</ref>
        - **Region:** ControlListAttachmentRegion
          - **View:** ControlListAttachmentView (IAttachmentView) <ref name="Windows" />
  - ContactHistoryNotepadView <ref name="Windows" /> (INotepadView)
  - **View:** ContactHistoryCaseDataView (IStaticCaseDataView) <ref name="Windows" />
    - **Region:** StaticCaseDataRegion
      - **View:** StaticCaseDataView (IGenericDataView) <ref name="Windows" >Module: Genesyslab.Desktop.Modules.Windows</ref>
    - **Region:** StaticDispositionCodeRegion
      - **View:** StaticDispositionCodeView (IGenericDataView) <ref name="Windows" >Module:

Genesyslab.Desktop.Modules.Windows</ref>

The following screenshot shows this component and its nested views and regions (but not all of them):



Workbins components. (full resolution)

## References

# Windows Utilities

## Tip

If you want to highlight the functional areas in the Workspace UI with the red boxes and descriptions shown in the screen shots in this topic, on a *non-production* developer workstation, either configure the value of the `keyboard.shortcut.action.test.hidden` developer option in the `interactionworkspace.exe.config` file or in Configuration Server using GAX, to a keyboard short cut, such as: `Ctrl+P`. When you press the keyboard shortcut, the functional areas are highlighted.

Genesys recommends that you set this option at the agent level for testing purposes before setting it higher up in the option hierarchy or in the `interactionworkspace.exe.config` file.

Upon execution of the file, the following files are generated in the execution folder:

- `DumpChainOfCommandsInHTML.htm` containing all chains of commands available in the product in html format.
- `DumpChainOfCommandsInWikiMarkup.txt` containing all chains of commands available in the product in wiki format.
- `DumpKeyboardShortcut.htm` containing the list of all commands that can be triggered by a shortcut, and the shortcut configured if any.
- `DumpViewTree.htm` containing all regions.
- `DumpFullTreeView.htm` (when full tree view is specified) containing view tree names.



**Purpose:** To present the list of accessories and utilities that your application can customize.

## Table Views

### Available since: 8.5

- **Window:** `InsertTableView (IInsertTableView)` <ref name="Windows"/>
  - **Region:** `InteractionsErrorRegion`
    - **View:** `InteractionsErrorView (ILastAlertErrorView)` <ref name="Windows"/>
- **Window:** `TablePropertiesView (ITablePropertiesView)` <ref name="Windows"/>
  - **Region:** `InteractionsErrorRegion`
    - **View:** `InteractionsErrorView (ILastAlertErrorView)` <ref name="Windows"/>



## Contacts and Favorites

**Available since:** 8.1.200.16; **updated in:** 8.1.300.17

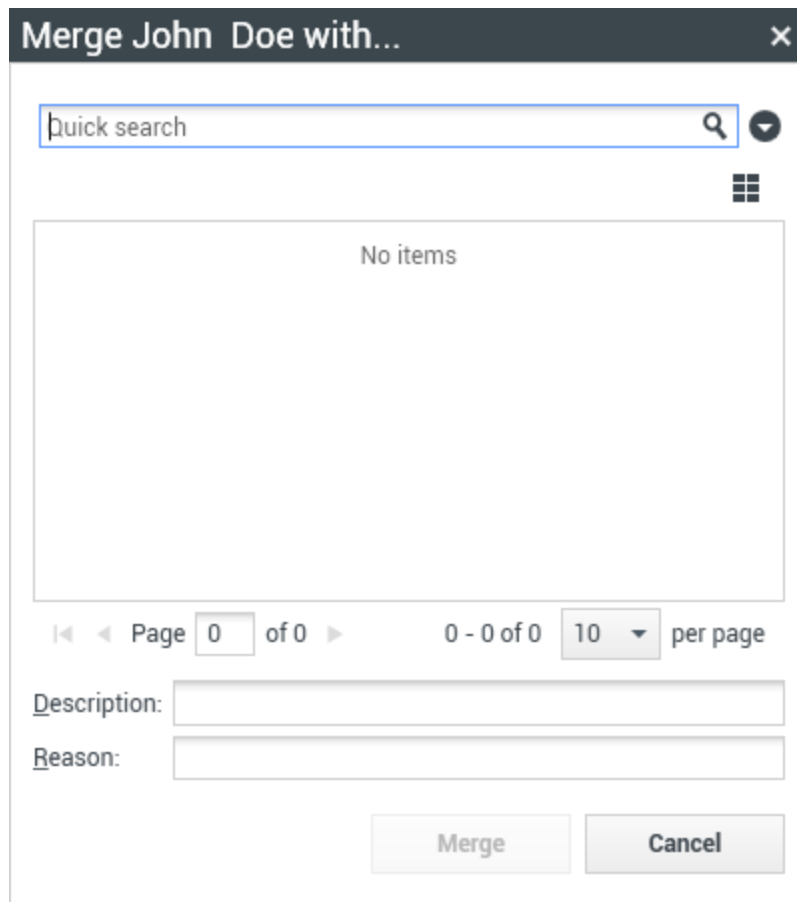
The following accessory is part of the `ToolBarWindowRegion` region, which includes the following regions and views.

### ContactSearchWindow

- **Window:** `ContactSearchWindow (IContactSearchWindow)` <ref name="Contacts">Genesyslab.Desktop.Modules.Contacts</ref>
  - **Region:** `SearchContactDirectoryRegion`
    - **View:** `SearchContactDirectoryView (IContactDirectoryView)` <ref name="Contacts">Genesyslab.Desktop.Modules.Contacts</ref>
      - **Region:** `ContactDirectoryErrorRegion`
        - **View:** `ContactDirectoryErrorView (ILastAlertErrorView)` <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>

### ContactMergeWindow

- **Window:** `ContactMergeWindow (IContactMergeWindow)` (Module: `Genesyslab.Desktop.Modules.Contacts`)
  - **Region:** `MergeContactDirectoryRegion`
    - **View:** `MergeContactDirectoryView (IContactDirectoryView)` (Module: `Genesyslab.Desktop.Modules.Contacts`)
      - **Region:** `ContactDirectoryErrorRegion`
        - **View:** `ContactDirectoryErrorView (ILastAlertErrorView)` (Module: `Genesyslab.Desktop.Modules.Windows`)



Utility for merging a contact in the Toolbar window

## ContainerView

- **Window:** ContainerView (IContainerView) (Module: Genesyslab.Desktop.Modules.TeamCommunicator)
  - **Region:** FavoriteViewRegion
    - **View:** NewContactView (INewContactView) (Module: Genesyslab.Desktop.Modules.TeamCommunicator)
      - **Region:** FavoriteViewErrorRegion
        - **View:** FavoriteViewErrorView (ILastAlertErrorView) (Module: Genesyslab.Desktop.Modules.Windows)

**New Favorite** [X]

Category: [Dropdown]

First Name: John

Last Name: Kochan

Phone Number: +1.415.9992716  
+1.650.4665028

E-mail Address: johnk@genesyslab.com

[OK] [Cancel]

Used to embed Team communicators' favorites.

## Forward and CancelForward

The following accessories are part of the MyPlaceStatusView view, which includes the following regions and views:

### Available since: 8.1.2; updated in 8.1.4

- **Window:** `forward ForwardView (IForwardView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>`  
Updated in 8.1.4
  - **Region:** `ForwardErrorRegion`
    - **View:** `ForwardErrorView (ILastAlertErrorView) <ref name="Windows" />`
- **Window:** `CancelForwardView (ICancelForwardView) (Module: Genesyslab.Desktop.Modules.Windows)`

## Additional Accessories

### Available since: 8.1.3; updated in 8.1.4

- **Window:** `IWMessageBoxView (IIWMessageBoxView) (Module: Genesyslab.Desktop.Modules.Windows)`
- **Window:** `AboutWindow (IAboutWindow) (Module: Genesyslab.Desktop.Modules.Windows)`
- **Window:** `AuthenticationWindow (IAuthenticationWindow) (Module: Genesyslab.Desktop.Modules.Windows)`
  - **Region:** `CapsLockRegion`
    - **View:** `CapsLockView (IErrorView) (Module: Genesyslab.Desktop.Modules.Windows)`

- **Window:** AuthenticationChangePasswordWindow (IAuthenticationChangePasswordWindow) (Module: Genesyslab.Desktop.Modules.Windows)
  - **Region:** AuthenticationChangePasswordErrorRegion
    - **View:** AuthenticationChangePasswordErrorView (IErrorView) (Module: Genesyslab.Desktop.Modules.Windows)
  - **Region:** CapsLockRegion
    - **View:** CapsLockView (IErrorView) (Module: Genesyslab.Desktop.Modules.Windows)
- **Window:** BroadcastMessageWindow (IBroadcastMessageWindow) (Module: Genesyslab.Desktop.Modules.Windows)
- **Window:** MediaInformationView (IMediaView) (Module: Genesyslab.Desktop.Modules.Windows)
  - **Region:** MediaCapsLockRegion
    - **View:** MediaCapsLockView(IErrorView) (Module: Genesyslab.Desktop.Modules.Windows)
- **Window:** PrintPreviewWindow (IPrintPreviewWindow) (Module: Genesyslab.Desktop.Modules.Windows)
- **Window:** EndWorkflowOpenMediaView (IEndWorkflowOpenMediaView) (Module: Genesyslab.Desktop.Modules.OpenMedia)
  - **Region:** ForwardErrorRegion
    - **View:** ForwardErrorView (ILastAlertErrorView) (Module: Genesyslab.Desktop.Modules.Windows)
- **Window:** CalendarWindowView (ICalendarWindowView) (Module: Genesyslab.Desktop.Modules.Contacts)

## References

# Adding web content to Workspace plugins using Microsoft Edge WebView2

To display web content inside Workspace Desktop Edition, developers of Workspace custom modules are often required to integrate a Web browser control into a custom Workspace view.

In the past, developers could leverage the built in .NET Framework WebBrowser; however, this embedded web technology based on Microsoft Internet Explorer became out of pace for modern business web application ecosystem, forcing the developer community to employ third-party solutions. Most of these solutions were based on the Chromium Open Source project. This approach came with the constraint of evaluating and adopting a third-party library which sometimes generated compatibility issues for multiple concurrent plugins used the same Chromium-based library.

Workspace Desktop Edition now embeds a WebView2 SDK, which allows Chromium-based rendering customizations with WebView2 control supported by [Microsoft Edge WebView2](#).

Microsoft Edge WebView2 is:

- A lightweight wrapper DLL-set for developers (embedded in Workspace) and a runtime (installed independently and shared with other applications).
- A solution to run multiple web-rendering plugins running side-by-side with full separation (if following code recommendations).
- An API supported by Microsoft.

Microsoft Edge WebView2 is NOT:

- A wrapper on top of Microsoft Edge. Microsoft Edge WebView2 runtime is installed independently from Microsoft Edge browser.
- The unique approach to render Web content in Workspace. Developers can continue to work with their favorite WebBrowser control.

## Installation

For WebView2 to operate correctly with a WebView2-based plugin in Workspace, Genesys requires WebView2 to be installed in the [Evergreen](#) installation mode.

Here are some useful links for getting started with WebView2:

- Download the runtime [here](#).
- Documentation starts [here](#).

## Pre-requisites for Microsoft Edge WebView2

Beginning with Workspace 8.5.147.05 or higher, you can access a [WebView2 sample](#) that you can use to create your own WebView2 app for Workspace.

Microsoft reference

<https://docs.microsoft.com/en-us/microsoft-edge/webview2/>

Build pre-requisites (for customization developers)

- .NET Framework 4.6.2 or higher.
- Visual Studio with .NET SDK 4.6.2 or above available (natively in the Visual Studio version or installed as an add-on).
- WebView2 Runtime installed through any Evergreen installation mode.
- WebView2 SDK is a part of WDE 8.5.147.05 or above. **Only DLLs located in the Workspace application should be used for developing.**
  - These can be updated in subsequent versions of Workspace.
  - Plugins based on older versions of WebView2 are supported.
  - WebView2 SDK is in the Workspace Library.

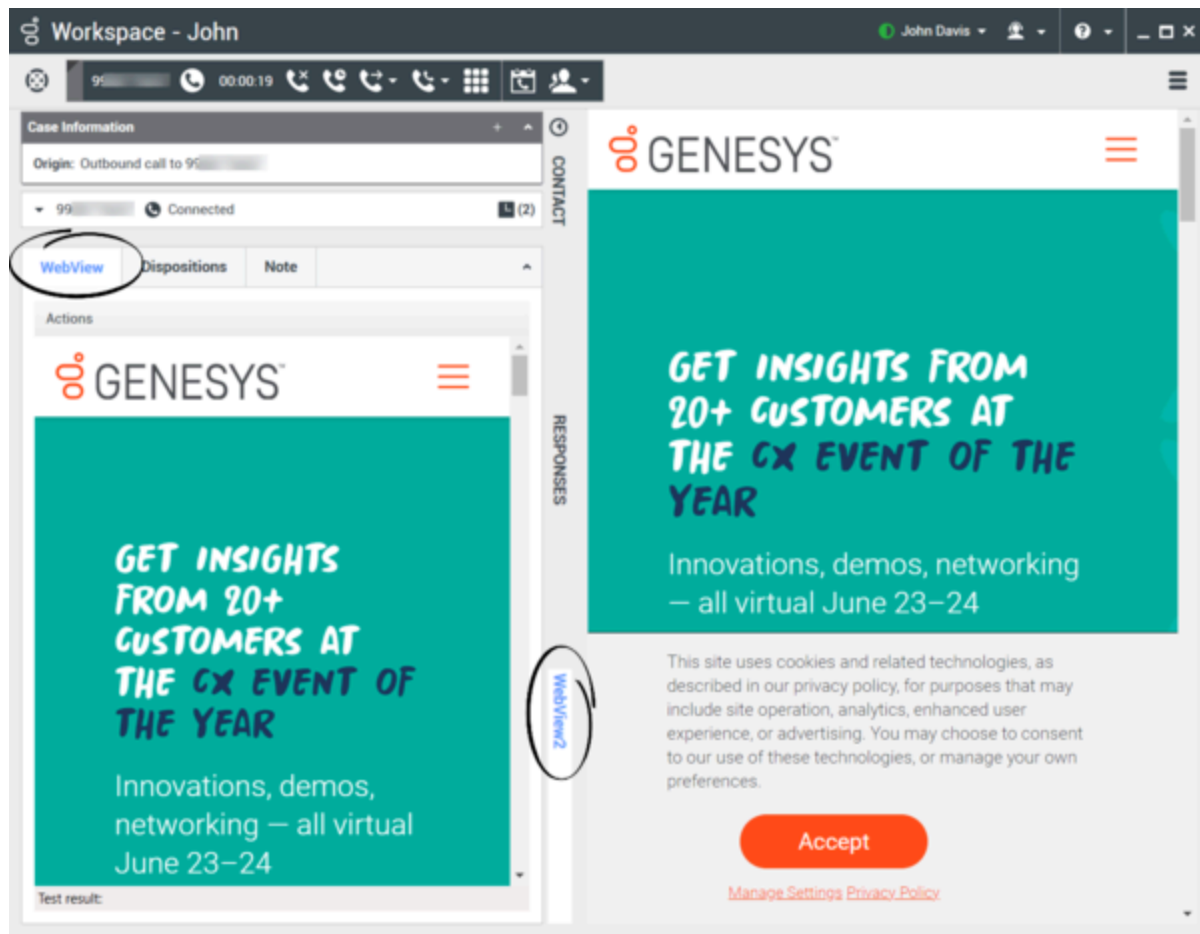
Runtime pre-requisites (for end-users of Workspace)

- .NET Framework 4.6.2 or higher.
- WebView2 Runtime installed through any Evergreen installation mode. **You must install this, it is not provided with Workspace.**
- WDE 8.5.147.05 or above.

## Sample Workspace plugin based on Microsoft Edge WebView2

Workspace 8.5.147.05 or higher includes a sample Workspace WebView2 plugin that you can use to help you develop your own app to embed web content in Workspace.

This image demonstrates some of the locations where you can implement a WebView2 plugin.



## Installing the sample

1. Download Workspace 8.5.147.05 or higher.
2. Install Workspace. Choose "[Install Workspace Desktop Edition Developer Toolkit](#)".
3. Find the folder with the WebView2 plugin sample in the **Samples** folder.
4. Open the solution in Visual Studio and Build it.
5. Run the Workspace application from the **Debug** folder.

This sample demonstrates some best practices to implement web content integration through the API of Microsoft Edge WebView2:

## Communication between host and web content

The host and web content can communicate with each other by using `postMessage` as follows:

- Web content in a `WebView2` control can post a message to the host using `window.chrome.webview.postMessage`. The host handles the message using any registered `WebMessageReceived` on the host.

- The Host can post messages to the web content in a WebView2 control using `CoreWebView2.PostWebMessageAsString` or `CoreWebView2.PostWebMessageAsJSON`. These messages are caught by handlers and are added to `window.chrome.webview.addEventListener`.

## Navigation events

During webpage navigation, the WebView2 control raises events. The application that hosts WebView2 controls listens for the following events:

- `NavigationStarting`
- `SourceChanged`
- `ContentLoading`
- `HistoryChanged`
- `NavigationCompleted`

## Process model

For information about how to configure your environment, refer to the [WebView2 Process Model](#) documentation. It is possible to use different configurations for each plugin you develop or use.

A single browser process is specified for each user data folder in a user session that serves WebView2 process requests specifying that user data folder. Therefore, one browser process can serve multiple process requests, and one requesting process can use multiple browser processes.

Each browser process has some number of associated renderer processes. Browser processes are created as needed to service (potentially) multiple frames in different instances of WebView2. The number of renderer processes varies based on the site isolation browser feature and the number of distinct disconnected origins rendered in associated instances of WebView2.

**Recommendation:** to fully separate the runtime environment of your WebView2-based plugin from the runtime of other WebView2-based plugins from other vendors, define a unique data folder name for your plugin. A simple way to do that is to assign the namespace of your plugin as the name of this folder.

### Example with two different environments:

```
CoreWebView2EnvironmentOptions opt = new CoreWebView2EnvironmentOptions();

opt.AdditionalBrowserArguments = "--enable-logging --log-
file=C:\\MyData\\tests\\WebView1.log";

CoreWebView2Environment env = await CoreWebView2Environment.CreateAsync(null,
"WebViewFolder1", opt);

await webView.EnsureCoreWebView2Async(env);

CoreWebView2EnvironmentOptions opt2 = new CoreWebView2EnvironmentOptions();

opt2.AdditionalBrowserArguments = "--enable-logging --log-
file=C:\\MyData\\tests\\WebView2.log";

CoreWebView2Environment env2 = await CoreWebView2Environment.CreateAsync(null,
"WebViewFolder2", opt2);
```



```
await webView2.EnsureCoreWebView2Async(env2);
```