



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 8.1.3

2/12/2022

Table of Contents

Interaction Workspace 8.1 Developer's Guide	3
Introduction	6
Introducing Extensions	10
About the Extension Samples	14
Write Custom Applications	17
Customizing Work Items	21
Advanced Customization	27
Frequently Asked Questions	34
Use Customizable Commands	39
Reference for Commands	43
Active Recording	52
Callback	54
Chat	57
E-Mails	67
IM	83
Open Media	86
Outbound	93
Place and Media	97
SMS	107
Supervision	115
UCS	118
Voice	122
Windows	134
Workbins	141
Work Items	143
Customize Views and Regions	148
Best Practices for Views	159
Reference for Windows	166
Interactions Window	168
Gadget Windows	176
Toaster Window	189
Toolbar Window	191
Windows Utilities	205

Interaction Workspace 8.1 Developer's Guide



Purpose: To provide recommendations and code samples that will help you to customize the out-of-the-box version of the Interaction 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 Interaction Workspace API and Extensions:

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

Limitation: Usage of Enterprise SDK that is provided with this release of Interaction Workspace is supported only for the purpose of Interaction Workspace customization.

New in This Document

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 Interaction Workspace API Reference chm.

Introduction



Purpose: Presents the architecture and design concepts of the Interaction Workspace.

Introducing Interaction Workspace

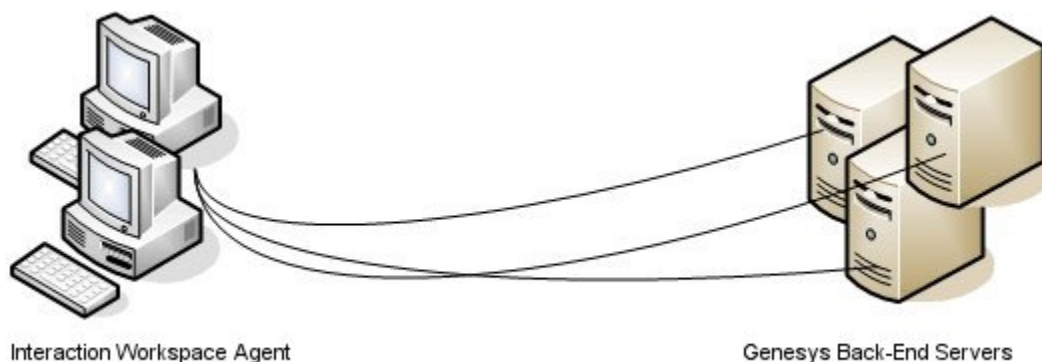
Interaction Workspace is the customer interaction interface for the Genesys 8 software suite. Interaction Workspace contains many components that you can use to enrich the content of Interaction Workspace with custom views and functionality. You can use the following Interaction Workspace components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

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

A set of Interaction Workspace Extension Samples is also provided to illustrate the best coding practices for Interaction Workspace customization. **Limitation:** Usage of Enterprise SDK that is provided with this release of Interaction Workspace is supported only for the purpose of Interaction Workspace 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 Interaction Workspace. 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

Note: Genesys Enterprise SDK 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 Interaction Workspace 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:

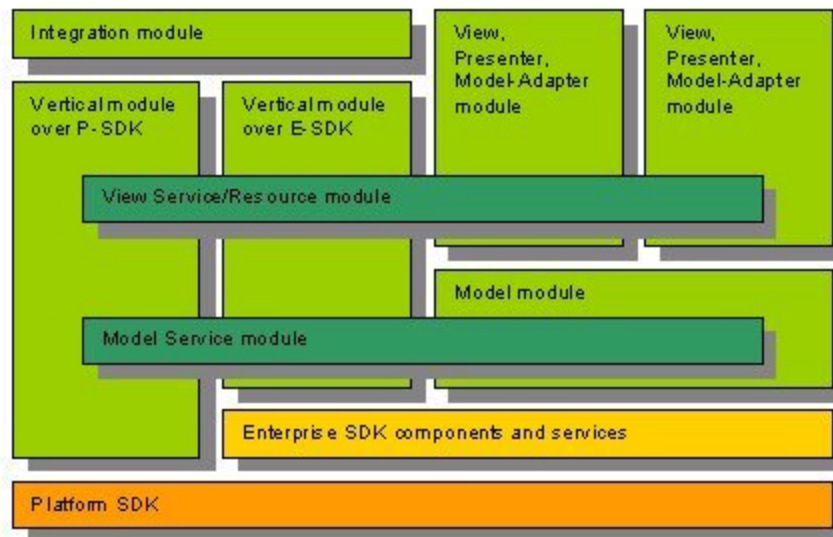
- Create a complex WPF application from modules that can be built, assembled, and optionally, deployed by independent teams.
- Minimize cross-team dependencies and enable teams to specialize in different areas, such as UI design, business logic implementation, and infrastructure code development.
- Use 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 Interaction Workspace 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)	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. • 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: • 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 construction. For example, in a typical application, a region can be a tab area.

Interaction Workspace 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. Interaction Workspace 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 Interaction Workspace. The figure below summarizes the various types of modules in Interaction Workspace.



Types of Modules in Interaction Workspace

<references/>

Read Next

 [Introducing Extensions](#)

Introducing Extensions



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

Introducing Extensions

You can customize Interaction Workspace by using two approaches:

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

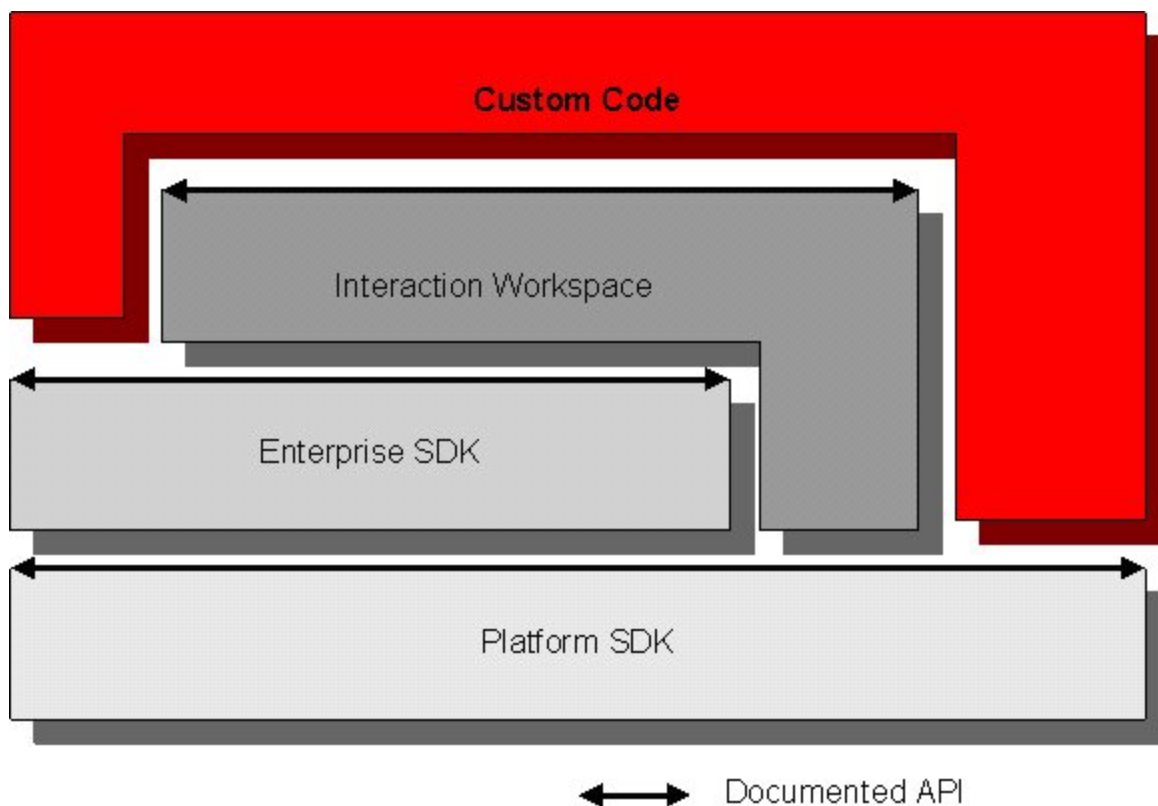
Developer Interfaces

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

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

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

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



Custom Code On Top of Developer Stack

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

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

Customization Options Using Extensions

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

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

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

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

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

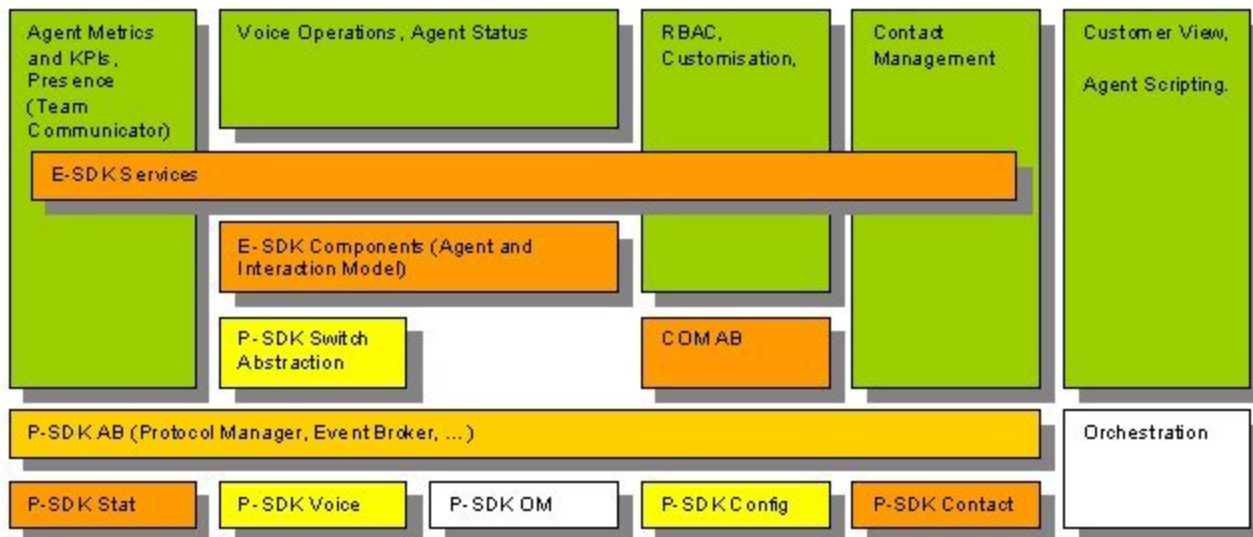
Certain extension use cases support two approaches to implementation, such as:

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

SDK Dependencies

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

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



SDK Dependencies

Read Next

 [About the Extension Samples](#)

About the Extension Samples



Purpose: To provide details about the Interaction Workspace Extension Samples.

Introducing Interaction Workspace Extension Samples

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

Locating the Extension Samples

The Interaction Workspace Extension Samples are included in the Interaction Workspace API along with the Interaction Workspace API Reference documentation. You can also download the samples available on this wiki:

- [InteractionWorkspaceExtensionSamples812.zip](#)
- [InteractionWorkspaceExtensionSamples811.zip](#)
- [InteractionWorkspaceExtensionSamples810.zip](#)

This version might differ from the version available in the latest installation package of the Interaction Workspace.

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

- A Bin directory that contains the Interaction Workspace API
- A Samples directory that contains code samples for developers that demonstrate Genesys' best practices recommendations
- An InteractionWorkspace directory that contains Interaction Workspace 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 in the Wiki

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 Interaction Workspace.
2. Click Next in the Welcome dialog box.
3. Select Install Interaction Workspace 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 [Interaction Workspace Deployment Guide](#).
7. Verify that the following directories are installed:
 - C:\Program Files\GCTI\Interaction Workspace\InteractionWorkspace\ (This folder contains all of the required binaries).
 - C:\Program Files\GCTI\Interaction Workspace\Samples\Genesyslab.Desktop.Modules.ExtensionSample\ (This folder contains the sample solution file).
8. To build and debug your custom module in Interaction Workspace combined with "Interaction

Workspace SIP Endpoint" or any Interaction Workspace 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\Interaction Workspace\InteractionWorkspace\".

For example:

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

Read Next

 [Write Custom Applications](#)

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 Interaction Workspace.

Using Interaction Workspace API

You can use the Interaction Workspace 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 Interaction Workspace API Reference.

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

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 Interaction Workspace. 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 Interaction Workspace 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 Interaction Workspace. The Interaction Workspace 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>();
        }
    }
}
```

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 Interaction Workspace (by using Deployment Manager if Click-Once is used, or by using a custom IP).
5. In the Interaction Workspace 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 Interaction Workspace.

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

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](#).

Read Next

 [Customizing Work Items](#)

Customizing Work Items



Purpose: To provide information about the customization of WorkItems.
Available since: 8.1.000.21

Before You Start

Work items are interactions which are not predefined in Interaction Workspace, in opposition to e-mail, instant messaging, voice, and chat interactions. You can define these work items to extend the Interaction Workspace 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 Interaction Workspace to integrate them. The code snippets are extracted from the Genesyslab.Desktop.Modules.CustomWorkItemSample, available in [the Interaction Workspace 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 called by the : **[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 SDK API to perform additional actions. 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;
                }
            }
        }
        catch { }
        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 Interaction Workspace.

Name	Object Id	Customization example
Media Channel Icon	Channel.OpenMedia.WorkItem.Image	<code><Value Id="Channel.OpenMedia.WorkItem.Image.customworkitem" ImageUrl="<ImagePath>" /></code>
Interaction Icon	InteractionWorkItem.Image	<code><Value Id="InteractionWorkItem.Image.customworkitem" Ico="<IcoPath>" ImageUrl="<ImagePath>" /></code>
Transfer Button	Windows.ToolbarWorkitemView. ButtonTransfer	<code><Value Id="Windows.ToolbarWorkitemView.ButtonTransfer" ToolTip="Custom Workitem Transfer" ImageUrl="<ImagePath>" /></code>
Transfer Menu	Windows.ToolbarWorkitemView. MenuItemOneStepTransferDialer	<code><Value Id="Windows.ToolbarWorkitemView.MenuItemOneStepTransferDialer" Text="Custom Workitem Transfer" ToolTip="Custom Workitem Transfer" ImageUrl="<ImagePath>" /></code>
'Move to Workbin' button	Windows.ToolbarWorkitemView. ButtonMoveToWorkbin	<code><Value Id="Windows.ToolbarWorkitemView.ButtonMoveToWorkbin" ToolTip="Move to workbin" ImageUrl="<ImagePath>" /></code>
New Custom Button in WorkItem ToolBar	Windows.ToolbarCustomWorkItemView. Button1	<code><Value Id="Windows.ToolbarCustomWorkItemView.Button1" ToolTip="Stop Processing" Text="Stop Processing" /></code>
Interaction icon in the Interaction History	Contacts.ContactHistoryView. InteractionWorkItem	<code><Value Id="Contacts.ContactHistoryView.InteractionWorkItem" ImageUrl="<ImagePath>" /></code>
Interaction Icon in Workin management	InteractionWorkItem.MediaType.Image	<code><Value Id="InteractionWorkItem.MediaType.Image.customworkitem" ImageUrl="<ImagePath>" /></code>

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>
```

Read Next

 [Use Customizable Commands](#)

Advanced Customization



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

Available since: 8.1.100.14; **updated in:** 8.1.300.17

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 Interaction Workspace. 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 Interaction Workspace 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 Interaction Workspace. The following table contains the list of downloadable API References available:

Interaction Workspace Release Version	Enterprise Service Release Version	Release Date	Enterprise Service CHM
8.1.300.17	8.1.300.14	07/31/2012	Download chm
8.1.200.16	8.1.200.10	01/31/2012	Download chm
8.1.100.14	8.1.100.10	07/29/2011	Download chm

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.EnterpriseService;
    //...
    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	• AgentProtocolRequest-"agent" • DeviceProtocolRequest-"device"
IChannelService	channelService	<i>none</i>
IDeviceService	deviceService	• DeviceProtocolRequest-"device"
IIdentityService	identityService	• OpenMediaProtocolRequest-"openmedia"
IIMService	IMService	• VoiceProtocolRequest-"voice" • DeviceProtocolRequest-"device" • IMProtocolRequest-"im"
IContactService	contactService	• ContactProtocolRequest -- "contacts"
IInteractionService	interactionService	<i>none</i>
IChatService	chatService	• OpenMediaProtocolRequest-"openmedia" • WebMediaProtocolRequest-"webmedia"
IOpenMediaService	openmediaService	<i>none</i>
IMonitorService	monitorService	• OpenMediaProtocolRequest-"openmedia"
IWorkbinService	workbinService	<i>none</i> • OpenMediaProtocolRequest-"openmedia"

Service Name	Service Key	Associated Protocols ^{<ref>Protocols can be used when you are managing channels.</ref>}
IPSTService	PSTService	none
ICampaignService	campaignService	none
IOutboundService	outboundService	none

<references />

Additional Entry Points

Interaction Workspace 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 Interaction Workspace.
IList<Genesyslab.Enterprise.Model.Interactions>	EnterpriseInteractions	The history of interactions.
Genesyslab.Platform.Commons.Protocol.Message	EnterpriseInteractionEvent	The last interaction event.
Genesyslab.Desktop.Modules.OpenMedia. Model.Interactions.Chat.IInteractionChatCommon	EnterpriseChatInteractionCurrent	Current chat interaction processed by Interaction Workspace.
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 Interaction Workspace.
Genesyslab.Desktop.Modules.OpenMedia. Interactions.IInteractionOpenMedia	EnterpriseOpenMediaInteractionCurrent	Current open media interaction processed by Interaction Workspace.
Genesyslab.Desktop.Modules.OpenMedia. Model.Interactions.Sms.IInteractionSms	EnterpriseSmsInteractionCurrent	Current sms interaction in page mode processed by Interaction Workspace.
	EnterpriseSmsSessionInteractionCurrent	Current sms interaction in session mode processed by Interaction Workspace.

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
IIdentityService	AgentServiceExtensions	Manage login, Ready, Not Ready
IDeviceService	DeviceServiceExtensions	Manage the call-forward and Do Not Disturb features.
IIMService	IMServiceExtensions	Manage the messages and transcripts of instant messaging sessions.
IInteractionService	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.

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

Managing Connections and Channels

Interaction Workspace 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 Interaction Workspace. 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)
{
    //...
}
```

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.EntrepriseInteractionCurrent 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 Interaction Workspace'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 Interaction Workspace 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 Interaction Workspace'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 CaseId, 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 Caseld, 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 Interaction Workspace 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 Interaction Workspace.

Before You Start

- 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](#)

Creating a Command

The command system is based on the chain of command (or chain of responsibility) design pattern. 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);

```

Inserting a Command in a Chain

Each element of command is unique across the given chain. You can use the `ICoManager.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"
    }
});
```

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	- Parameter1: <code>InteractionChat</code> - Parameter2: <code>KeyValuePairCollection</code>
	Command2	- Parameter1: <code>InteractionChat</code> - Parameter3: <code>KeyValuePairCollection</code>

- `InteractionChat`: `Genesyslab.Desktop.Modules.OpenMedia.Model.Interactions.Chat.InteractionChat`
- `KeyValuePairCollection`: `Genesyslab.Enterprise.Commons.Collections.KeyValuePairCollection`

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);
```

Read Next

 [Customize Views and Regions](#)

Reference for Commands



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

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

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

Chains of Commands

Available chains of command	Related pages
- ActiveRecordingPauseRecordIxnsScreenCommand - ActiveRecordingPauseRecordScreenCallCommand - ActiveRecordingResumeRecordIxnsScreenCommand - ActiveRecordingResumeRecordScreenCallCommand - ActiveRecordingStartRecordIxnsScreenCommand - ActiveRecordingStartRecordScreenCallCommand - ActiveRecordingStopRecordIxnsScreenCommand - ActiveRecordingStopRecordScreenCallCommand	Active Recording
- ApplicationClose - BroadcastMessageClose - BroadcastMessageShow	Windows
BundleClose	Chat Callback IM E-Mails Voice Open Media SMS Work Items

Available chains of command	Related pages
CampaignGetPreviewRecord	Outbound
CancelConnectionCommand	Windows
- CancelOutboundChain - ClosePullPreviewRecord	Outbound
- ContactIsModified - ContactNotification - ContactOpenSearchWindow - ContactOpenToDialog - CreateInteractionUCSFromESDKInteraction - DeleteInteractionUCS	UCS
- InteractionCallbackAccept - InteractionCallbackActiveCustomerSession - InteractionCallbackCompleteCustomerSession - InteractionCallbackDecline - InteractionCallbackIsPossibleToClose - InteractionCallbackParkParent - InteractionCallbackPullParent - InteractionCallbackWorkflow	Callback
- InteractionChatAcceptChat - InteractionChatAutoCloseInteraction - InteractionChatAutoDisconnect - InteractionChatCloseInteraction - InteractionChatCompleteConferenceAgent - InteractionChatCompleteTransferAgent - InteractionChatConferenceAgent - InteractionChatConsultPromptForEnd - InteractionChatConsultRelease	Chat

Available chains of command	Related pages
• InteractionChatDeclineChat • InteractionChatDisconnectChat • InteractionChatDisconnectChatSessionChannel • InteractionChatDisconnectMonitoredChat • InteractionChatDisconnectMonitoredCoachChat • InteractionChatInitConsultation • InteractionChatIsPossibleToClose • InteractionChatLeaveInteractionFromConference • InteractionChatPlaceInQueue • InteractionChatPushUrl • InteractionChatReconnect • InteractionChatReleasePartyFromConference • InteractionChatReleasePartyFromConsult • InteractionChatReopenChannel • InteractionChatSendMessage • InteractionChatTransferAgent • InteractionChatTypingStarted • InteractionChatTypingStopped • InteractionChatWorkflow	
• InteractionEmailActionOnOpenedEmailById • InteractionEmailClose • InteractionEmailDelete • InteractionEmailDeleteById • InteractionEmailForward • InteractionEmailIsPossibleToClose • InteractionEmailLoad • InteractionEmailMarkDoneById • InteractionEmailMoveToWorkbin • InteractionEmailPlaceInQueue • InteractionEmailPrint • InteractionEmailPullById	E-Mails

Available chains of command	Related pages
- InteractionEmailPullFromWorkbinById - InteractionEmailReply - InteractionEmailReplyById - InteractionEmailResend - InteractionEmailSave - InteractionEmailSaveNewOutbound - InteractionEmailSend - InteractionEmailSendInterimReply - InteractionEmailTransferAgent - InteractionEmailWorkflow - InteractionEmailWorkflowFromInteractionESDK	
- InteractionIMClose - InteractionIMIsPossibleToClose - InteractionIMReleaseIM - InteractionIMSendMessage	IM
- InteractionInboundEmailPossibleToMoveToWorkbin - InteractionInboundEmailSearchAndMoveToWorkbin	E-Mails
- InteractionOpenMediaCanPullInteraction - InteractionOpenMediaClose - InteractionOpenMediaCloseOnOutOfService - InteractionOpenMediaConferenceAgent - InteractionOpenMediaGetInteractionStateById - InteractionOpenMediaHandlingTimeout - InteractionOpenMediaPlaceInQueue - InteractionOpenMediaPullById - InteractionOpenMediaPullFromWorkbinById - InteractionOpenMediaTransferAgent - InteractionOpenMediaUCSStopProcessing - InteractionOpenMediaUCSUpdate	Open Media

Available chains of command	Related pages
InteractionOpenMediaWorkflow	
- InteractionOutboundEmailPossibleToMoveToWorkbin - InteractionOutboundEmailSearchAndMoveToWorkbin	E-Mails
InteractionPossibleToMoveToWorkbin	Workbins
- InteractionPushPreviewAccept - InteractionPushPreviewDecline - InteractionPushPreviewWorkflow	Outbound
InteractionSetAttachedDataWithDelay	- Callback - Chat - IM - E-Mails - Voice - SMS - Open Media - Work Items
- InteractionSmsAccept - InteractionSmsCreateReplyMessage - InteractionSmsDecline - InteractionSmsDelete - InteractionSmsDisconnectSmsSessionChannel - InteractionSmsGetHistoricMessage - InteractionSmsIsPossibleToClose - InteractionSmsPlaceInQueue - InteractionSmsSendMessage - InteractionSmsSendNewMessage - InteractionSmsSendReplyMessage - InteractionSmsTransferAgent	SMS

Available chains of command	Related pages
InteractionSmsWorkflow	
InteractionUCSInQueue	UCS
- InteractionVoiceAlternateCall - InteractionVoiceAnswerCall - InteractionVoiceAutoCloseConsult - InteractionVoiceBeforeClose - InteractionVoiceClose - InteractionVoiceCompleteConference - InteractionVoiceCompleteTransfer - InteractionVoiceDeleteFromConference - InteractionVoiceHoldCall - InteractionVoiceIfPossibleCloseInteraction - InteractionVoiceInitConference - InteractionVoiceInitTransfer - InteractionVoiceIsPossibleToClose - InteractionVoiceListenDisconnect - InteractionVoiceListenReconnect - InteractionVoiceMuteOff - InteractionVoiceMuteOn - InteractionVoicePauseRecordCall - InteractionVoicePauseRecordScreenCall - InteractionVoiceReconnectCall - InteractionVoiceRecordCall - InteractionVoiceRedirectCall - InteractionVoiceReleaseCall - InteractionVoiceResume - InteractionVoiceResumeRecordCall - InteractionVoiceResumeRecordScreenCall - InteractionVoiceRetrieveCall - InteractionVoiceSendDTMF	Voice

Available chains of command	Related pages
- InteractionVoiceSingleStepConference - InteractionVoiceSingleStepTransfer - InteractionVoiceStartRecordScreenCall - InteractionVoiceStopRecordCall - InteractionVoiceStopRecordScreenCall	
- InteractionWorkItemAccept - InteractionWorkItemDecline - InteractionWorkitemMarkDoneById - InteractionWorkItemMoveToWorkbin - InteractionWorkItemPossibleToMoveToWorkbin - InteractionWorkitemPutBackToOriginalSource - InteractionWorkItemSearchAndMoveToWorkbin - InteractionWorkitemWorkflowFromInteractionESDK	Work Items
MarkProcessedOutboundChain	Outbound
MediaEmailCreateNewOutboundEmail	Place and Media
- MediaOpenMediaChangeStateReason - MediaOpenMediaDNDOff - MediaOpenMediaDNDOn - MediaOpenMediaLogOff - MediaOpenMediaLogOffPlace - MediaOpenMediaLogOn - MediaOpenMediaLogOnPlace - MediaOpenMediaNotReady - MediaOpenMediaReady - MediaSmsCreateNewOutboundPageMode - MediaVoiceDNDOff - MediaVoiceDNDOn - MediaVoiceLogOff	Place and Media

Available chains of command	Related pages
- MediaVoiceLogOn - MediaVoiceMakeCall - MediaVoiceMakeIM - MediaVoiceNotReady - MediaVoiceReady - MediaVoiceStartIMSession	
- NotepadInsertTimeStamp - PossibleToCloseAllInteractions - PossibleToClosePullPreview - PossibleToClosePushPreview	Windows
RejectOutboundChain	Outbound
SetLastCalledAgentUCS	UCS
StartPreviewRecordOutbound	Outbound
- TeamCommunicatorAddToFavorite - TeamCommunicatorModifyFavorite - TeamCommunicatorOpenInteraction	Windows
- TeamLeadEndMonitoring - TeamLeadEndMonitoringChat - TeamLeadEndMonitoringVoice - TeamLeadMonitorAll - TeamLeadSetSupervisorBargeInChat - TeamLeadSetSupervisorBargeInVoice - TeamLeadSetSupervisorCoachChat - TeamLeadSetSupervisorCoachVoice - TeamLeadSetSupervisorMuteVoice - TeamLeadStopMonitor	Supervision

Available chains of command	Related pages
• <code>ToasterAcceptPreview</code> • <code>ToasterDeclinePreview</code> • <code>ToasterInteractionAccept</code> • <code>ToasterInteractionDecline</code> • <code>ToasterInteractionEmailAccept</code> • <code>ToasterInteractionEmailDecline</code> • <code>ToasterInteractionIMAccept</code> • <code>ToasterInteractionIMReject</code> • <code>ToasterInteractionOpenMediaAccept</code> • <code>ToasterInteractionOpenMediaDecline</code> • <code>ToasterInteractionPreviewAccept</code> • <code>ToasterInteractionPreviewDecline</code> • <code>ToasterShowBroadcastMessage</code>	Windows
• <code>UpdateOutboundRecord</code>	Outbound
• <code>WindowClose</code> • <code>WindowInteractionClose</code>	Windows
• <code>WorkbinPreSelectedByName</code> • <code>WorkbinPullInteraction</code> • <code>WorkbinPushInteraction</code> • <code>WorkbinSelected</code>	Workbins

Active Recording



Purpose: To present for each active recording command, the list of default subcommands available and the associated parameters.
Available since: 8.1.100.14

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	
• Pause	• CommandParameter: <code>InteractionOpenMedia<ref name="InteractionOpenMedia" > Genesyslab.Desktop.Modules.Core.Model.Interactions.IInteracti</code>
Chain ActiveRecordingPauseRecordScreenCallCommand	
• Pause	• CommandParameter: <code>InteractionVoice<ref name="InteractionVoice" > Genesyslab.Desktop.Modules.Core.Model.Interactions.IInteracti</code>
Chain ActiveRecordingResumeRecordIxnScreenCommand	
• Resume	• CommandParameter:

Default Commands	Parameters
	InteractionOpenMedia<ref name="InteractionOpenMedia" />
Chain ActiveRecordingResumeRecordScreenCallCommand	
Resume	CommandParameter: InteractionVoice<ref name="InteractionVoice" />
Chain ActiveRecordingStartRecordIxnScreenCommand	
Start	CommandParameter: InteractionOpenMedia<ref name="InteractionOpenMedia" />
Chain ActiveRecordingStartRecordScreenCallCommand	
Start	CommandParameter: InteractionVoice<ref name="InteractionVoice" />
Chain ActiveRecordingStopRecordIxnScreenCommand	
Stop	CommandParameter: InteractionOpenMedia<ref name="InteractionOpenMedia" />
Chain ActiveRecordingStopRecordScreenCallCommand	
Stop	CommandParameter: InteractionVoice<ref name="InteractionVoice" />

<references />

Callback



Purpose: To provide information about callback commands.

Available since: 8.1.300.17

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.

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	• CommandParameter: IInteractionOpenMedia<ref name="IInteractionOpenMedia">Genesyslab.Desktop.Modules.OpenM • Extensions: KeyValueCollection<ref name="KeyValueCollection" > Genesyslab.Enterprise.Commons.Collections.KeyValueCollection
Chain InteractionCallbackActiveCustomerSession	
• ActiveCustomerSessionCallback	• CommandParameter: IInteractionCallback<ref name="IInteractionCallback" > Genesyslab.Desktop.Modules.Callback.Model.Interactions.IInter

Default commands	Parameters
Chain InteractionCallbackCompleteCustomerSession	
CompleteCustomerSessionCallback	CommandParameter: IInteractionCallback<ref name="IInteractionCallback" />
Chain InteractionCallbackDecline Updated since: 8.1.300.17	
DeclineCallback	- CommandParameter: IInteractionOpenMedia<ref name="IInteractionOpenMedia" /> - Reasons: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
Close	CommandParameter: IInteractionOpenMedia<ref name="IInteractionOpenMedia" />
Chain InteractionCallbackIsPossibleToClose	
IsPossibleToClose	CommandParameter: IInteractionCallback<ref name="IInteractionCallback" />
Chain InteractionCallbackParkParent	
ParkCallback	CommandParameter: IInteraction<ref name="IInteraction">Genesyslab.Desktop.Modules.Core.Model.Interactions.IInteraction</ref>
Chain InteractionCallbackPullParent	
PullCallback	- CommandParameter: IInteraction<ref name="IInteraction" /> - IW_WebcallbackInteractionId: String<ref name="String">System.String</ref>
Chain InteractionCallbackWorkflow Updated since: 8.1.300.17	
RescheduleCallback	CommandParameter: IInteractionCallback<ref name="IInteractionCallback" />
UpdateInteractionUCS	CommandParameter: IInteraction<ref name="IInteraction" />

Default commands	Parameters
MarkProcessedCallback	CommandParameter: IInteractionCallback<ref name="IInteractionCallback" />
Commands common to all interactions	
Chain InteractionSetAttachedDataWithDelay	
SetAttachedData	- CommandParameter: IInteraction<ref name="IInteraction" /> - AttachedData: KeyValueCollection<ref name="KeyValueCollection" />
Chain BundleClose	
GetAttachedDataInformationUCS	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" > Genesyslab.Desktop.Modules.Core.Model.Interactions.IInteracti
UpdateNotePadForVoice Available since: 8.1.300.17	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />
ResetInteractionChatConsultation	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />
IsPossibleToClose	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />
CompleteDispositionCodeOnBundle	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />
Close	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />
StopInteractionVoiceUCS	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />
GetOutboundPreviewRecord	CommandParameter: ICampaign<ref name="ICampaign" > Genesyslab.Enterprise.Model.Interaction.ICampaign</ref>

<references />

Chat



Purpose: To present for each chat command, the list of default subcommands available and the associated parameters.
Updated in: 8.1.300.17

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.

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	
• Accept	• CommandParameter: IInteractionChat<ref name="IInteractionChat" > Genesyslab.Desktop.Modules.OpenMedia.Model.Interactions.CH
Chain InteractionChatAutoCloseInteraction	

Default Commands	Parameters
ResetInteractionChatConsultationCommand	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" >Genesyslab.Desktop.Modules.Core.Model.Interactions.IInterac
Close	CommandParameter: IInteractionChat<ref name="IInteractionChat" />
CloseConsultation	CommandParameter: IInteractionChat<ref name="IInteractionChat" />
Chain InteractionChatAutoDisconnect	
ReleaseMyParty	- CommandParameter: IInteractionChat<ref name="IInteractionChat" /> - AfterActionMode: String<ref name="String" >System.String</ref>
Chain InteractionChatCloseInteraction	
IsPossibleToClose	CommandParameter: IInteractionChat<ref name="IInteractionChat" />
Close	CommandParameter: IInteractionChat<ref name="IInteractionChat" />
Chain InteractionChatCompleteConferenceAgent	
CompleteConference	- CommandParameter: IInteractionChatCommon<ref name="IInteractionChatCommon" >Genesyslab.Desktop.Modules.OpenMedia.Model.Interactions.CH - Destination: String<ref name="String" /> - Reason: KeyValueCollection<ref name="KeyValueCollection" >Genesyslab.Enterprise.Commons.Collections.KeyValueCollection - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
Chain InteractionChatCompleteTransferAgent	
RequestSaveNotepad	CommandParameter: IInteraction<ref name="IInteraction" >Genesyslab.Desktop.Modules.Core.Model.Interactions.IInteracti

Default Commands	Parameters
	NotepadViewModel: INotepadViewModel<ref name="INotepadViewModel" > Genesyslab.Desktop.Modules.Windows.Interactions.INotepadVi
CompleteTransfer	- CommandParameter: InteractionChatCommon<ref name="InteractionChatCommon" /> - Destination: String<ref name="String" /> - Reason: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
Chain InteractionChatConferenceAgent	
ConferenceAgent	- CommandParameter: IInteractionOpenMedia<ref name="InteractionOpenMedia" > Genesyslab.Desktop.Modules.OpenMedia.Model.Interactions.IIn - Visibility: String<ref name="String" /> - Destination: String<ref name="String" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
AddRecentCall	RecentIndex: ObjectTC<ref name="ObjectTC" > Genesyslab.Desktop.Modules.TeamCommunicator.ObjectTC</ref>
Chain InteractionChatConsultPromptForEnd Available since: 8.1.200.16	
PromptEnd	- CommandParameter: IInteraction<ref name="Interaction" /> - ConfirmParentWindow: Window<ref name="Window" > System.Window.Window</ref>
Chain InteractionChatConsultRelease	
SetSupervisorMuteChat Available since: 8.1.300.17	
Release	CommandParameter: InteractionChatConsult<ref name="InteractionChatConsult" >

Default Commands	Parameters
	Genesyslab.Desktop.Modules.OpenMedia.Model.Interactions.Ch - Reason: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
Chain InteractionChatDeclineChat	
Decline	- CommandParameter: IInteractionChat<ref name="IInteractionChat" /> - Reasons: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
Close Available since: 8.1.200.16	CommandParameter: IInteractionOpenMedia<ref name="IInteractionOpenMedia" />
Chain InteractionChatDisconnectChat	
PromptEnd Available since: 8.1.200.16	- CommandParameter: IInteraction<ref name="IInteraction" /> - ConfirmParentWindow: Window<ref name="Window" />
SetSupervisorMuteChat Available since: 8.1.300.17	
Disconnect	- CommandParameter: IInteractionChat<ref name="IInteractionChat" /> - Reason: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
Chain InteractionChatDisconnectChatSessionChannel	
Disconnect	CommandParameter: IClientChannel<ref name="IClientChannel" > Genesyslab.Enterprise.Model.Channel.IClientChannel</ref>
Chain InteractionChatDisconnectMonitoredChat	

Default Commands	Parameters
Available since: 8.1.200.16	
Disconnect	- CommandParameter: <code>InteractionChat<ref name="InteractionChat" /></code> - Reason: <code>KeyValueCollection<ref name="KeyValueCollection" /></code> - Extensions: <code>KeyValueCollection<ref name="KeyValueCollection" /></code>
Chain <code>InteractionChatDisconnectMonitoredCoachChat</code> Available since: 8.1.200.16	
DisconnectCoach	- CommandParameter: <code>InteractionChat<ref name="InteractionChat" /></code> - Reason: <code>KeyValueCollection<ref name="KeyValueCollection" /></code> - Extensions: <code>KeyValueCollection<ref name="KeyValueCollection" /></code>
Chain <code>InteractionChatInitConsultation</code>	
SetTransferInformation	- CommandParameter: <code>Interaction<ref name="Interaction" /></code> - NotepadViewModel: <code>INotepadViewModel<ref name="INotepadViewModel" /></code>
RequestSaveNotepad	- CommandParameter: <code>Interaction<ref name="Interaction" /></code> - NotepadViewModel: <code>INotepadViewModel<ref name="INotepadViewModel" /></code>
InitConsultation	- CommandParameter: <code>InteractionChatCommon<ref name="InteractionChatCommon" /></code> - Destination: <code>String<ref name="String" /></code> - Reasons: <code>KeyValueCollection<ref name="KeyValueCollection" /></code> - Extensions: <code>KeyValueCollection<ref name="KeyValueCollection" /></code>
AddRecentCall	RecentIndex: <code>ObjectTC<ref name="ObjectTC" /></code>

Default Commands	Parameters
Chain InteractionChatIsPossibleToClose	
IsPossibleToClose	CommandParameter: IInteractionChat<ref name="IInteractionChat" />
Chain InteractionChatLeaveInteractionFromConference	
Leave	- CommandParameter: IInteractionChat<ref name="IInteractionChat" /> - Reason: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
Chain InteractionChatPlaceInQueue	
UpdateInteractionUCS	- CommandParameter: IInteraction<ref name="IInteraction" /> - NotepadViewModel: INotepadViewModel<ref name="INotepadViewModel" /> New parameter since: 8.1.300.17
SaveAttachedData Available since: 8.1.100.14	- CommandParameter: IInteractionOpenMedia<ref name="IInteractionOpenMedia" /> - UserData: KeyValueCollection<ref name="KeyValueCollection" />
PlaceInQueue	- CommandParameter: IInteractionOpenMedia<ref name="IInteractionOpenMedia" /> - Destination: String<ref name="String" /> - Reasons: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
SetAutoClose	CommandParameter: IInteractionOpenMedia<ref name="IInteractionOpenMedia" />
ReleaseMyParty	CommandParameter: IInteractionChat<ref name="IInteractionChat" />

Default Commands	Parameters
	AfterActionMode: String<ref name="String" />
AddRecentCall	RecentIndex: ObjectTC<ref name="ObjectTC" />
Chain InteractionChatPushUrl	
PushUrl	- CommandParameter: IInteractionChat<ref name="IInteractionChat" /> - Url: String<ref name="String" /> - Visibility: String<ref name="String" />
Chain InteractionChatReconnect Available since: 8.1.200.16	
Join	CommandParameter: IInteractionChat<ref name="IInteractionChat" />
Chain InteractionChatReleasePartyFromConference	
ReleaseParty	CommandParameter: IChatParty<ref name="IChatParty" /> Genesyslab.Desktop.Modules.OpenMedia.Model.Interactions.CH
Chain InteractionChatReleasePartyFromConsult	
ReleaseParty	CommandParameter: IChatParty<ref name="IChatParty" />
Chain InteractionChatReopenChannel Available since: 8.1.200.16	
Reopen	CommandParameter: IInteractionChat<ref name="IInteractionChat" />
Chain InteractionChatSendMessage	
SendMessage	- CommandParameter: IInteractionChatCommon<ref name="IInteractionChatCommon" /> - Visibility: String<ref name="String" /> - Message: String<ref name="String" />
Chain InteractionChatTransferAgent	

Default Commands	Parameters
RequestSaveNotepad	- CommandParameter: IInteraction<ref name="Interaction" /> - NotepadViewModel: INotepadViewModel<ref name="INotepadViewModel" />
TransferAgent	- CommandParameter: IInteractionOpenMedia<ref name="InteractionOpenMedia" /> - Destination: String<ref name="String" /> - Reasons: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
SetAutoClose	CommandParameter: IInteractionOpenMedia<ref name="InteractionOpenMedia" />
ReleaseMyParty	- CommandParameter: IInteractionChat<ref name="IInteractionChat" /> - AfterActionMode: String<ref name="String" />
AddRecentCall	RecentIndex: ObjectTC<ref name="ObjectTC" />
Chain InteractionChatTypingStarted	
TypingStarted	- CommandParameter: IInteractionChatCommon<ref name="InteractionChatCommon" /> - Visibility: String<ref name="String" />
Chain InteractionChatTypingStopped	
TypingStopped	- CommandParameter: IInteractionChatCommon<ref name="InteractionChatCommon" /> - Visibility: String<ref name="String" />
Chain InteractionChatWorkflow	

Default Commands	Parameters
CloseConsult	CommandParameter: IInteractionChat<ref name="IInteractionChat" />
Workflow	- CommandParameter: IInteractionOpenMedia<ref name="IInteractionOpenMedia" /> - Workflow: String<ref name="String" /> - Reasons: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
Commands common to all interactions	
Chain InteractionSetAttachedDataWithDelay	
SetAttachedData	- CommandParameter: IInteraction<ref name="IInteraction" /> - AttachedData: KeyValueCollection<ref name="KeyValueCollection" />
Chain BundleClose	
GetAttachedDataInformationUCS	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" /> Genesyslab.Desktop.Modules.Core.Model.Interactions.IInteracti
UpdateNotePadForVoice Available since: 8.1.300.17	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />
ResetInteractionChatConsultation	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />
IsPossibleToClose	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />
CompleteDispositionCodeOnBundle	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />
Close	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />

Default Commands	Parameters
StopInteractionVoiceUCS	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />
GetOutboundPreviewRecord	CommandParameter: ICampaign<ref name="ICampaign" > Genesyslab.Enterprise.Model.Interaction.ICampaign</ref>

<references />

E-Mails



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

Updated in: 8.1.300.17

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.

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 Available since: 8.1.301.xx	
• GetInteractionState	• CommandParameter: String<ref name="String" />

Default Commands	Parameters
CanPullInteraction	CommandParameter: Dictionary<ref name="Dictionary" />
DisplayWarningPullNotPossible	canPullInteractionresult: String<ref name="String" />
Confirm	- ActionToConfirm: String<ref name="String" /> - ConfirmParentWindow: Window<ref name="Window" />
Delete	- CommandParameter: String<ref name="String" /> - WorkbinId: String<ref name="String" />
Chain InteractionEmailActionFromWorkbinMarkDoneById Available since: 8.1.301.xx	
GetInteractionState	CommandParameter: String<ref name="String" />
CanPullInteraction	CommandParameter: Dictionary<ref name="Dictionary" />
DisplayWarningPullNotPossible	canPullInteractionresult: String<ref name="String" />
NoChildrenOpened	- CommandParameter: String<ref name="String" /> - WarningMessageTarget: String<ref name="String" /> - DoNotCheckForParentId: String<ref name="String" />
Confirm	- ActionToConfirm: String<ref name="String" /> - ConfirmParentWindow: Window<ref name="Window" />
MarkDone	CommandParameter: String<ref name="String" />

Default Commands	Parameters
	WorkbinId: String<ref name="String" />
Chain InteractionEmailActionFromWorkbinPullById Available since: 8.1.301.xx	
GetInteractionState	CommandParameter: String<ref name="String" />
CanPullInteraction	CommandParameter: Dictionary<ref name="Dictionary" />
DisplayWarningPullNotPossible	canPullInteractionresult: String<ref name="String" />
NoChildrenOpened	- CommandParameter: String<ref name="String" /> - WarningMessageTarget: String<ref name="String" /> - DoNotCheckForParentId: String<ref name="String" />
NoDraftWithSameParentOpened	ParentId: String<ref name="String" />
PullFromWorkbin	- CommandParameter: String<ref name="String" /> - WorkbinId: String<ref name="String" />
MoveToWorkbinOpenedParent	ParentId: String<ref name="String" />
Chain InteractionEmailActionOnOpenedEmailById	
IsNotOpened	- CommandParameter: String<ref name="String">System.String</ref> - OpenedInteractionCommand: String<ref name="String" />
Chain InteractionEmailClose	
IsPossibleToClose	CommandParameter: IInteractionEmail<ref name="IInteractionEmail" > Genesyslab.Desktop.Modules.OpenMedia.Model.Interactions.En

Default Commands	Parameters
Close	CommandParameter: IInteractionEmail<ref name="IInteractionEmail" />
Chain InteractionEmailDelete	
Confirm	- ActionToConfirm: String<ref name="String" /> - ConfirmParentWindow: Window<ref name="Window" > System.Window.Window</ref>
PullParentInboundEmail	- CommandParameter: IInteractionEmail<ref name="IInteractionEmail" /> - DispositionCode: IDispositionCode<ref name="IDispositionCode" > Genesyslab.Desktop.Modules.Core.Model.Interactions.IDispositi
Delete	CommandParameter: IInteractionEmail<ref name="IInteractionEmail" />
CleanConnectionInformation Available since: 8.1.200.16	CommandParameter: String<ref name="String" />
Chain InteractionEmailDeleteById	
Confirm	- ActionToConfirm: String<ref name="String" /> - ConfirmParentWindow: Window<ref name="Window" />
Delete	- CommandParameter: String<ref name="String" /> - WorkbinId: String<ref name="String" />
Chain InteractionEmailForward Available since: 8.1.200.16	
IsPossibleToForward	CommandParameter: IInteractionInboundEmail<ref name="IInteractionInboundEmail" > Genesyslab.Desktop.Modules.OpenMedia.Model.Interactions.En
AddForwardData	CommandParameter: IInteractionInboundEmail<ref name="IInteractionInboundEmail" />

Default Commands	Parameters
	Destination: String<ref name="String" />
SetAutoClose	CommandParameter: IInteractionOpenMedia<ref name="InteractionOpenMedia" />
SaveAttachedData	- CommandParameter: IInteractionOpenMedia<ref name="InteractionOpenMedia" /> - UserData: KeyValueCollection<ref name="KeyValueCollection" />
RequestSaveNotepad	- CommandParameter: IInteraction<ref name="Interaction" /> - NotepadViewModel: INotepadViewModel<ref name="INotepadViewModel" />
PlaceInQueue	- CommandParameter: IInteractionOpenMedia<ref name="InteractionOpenMedia" /> - Destination: String<ref name="String" /> - Reason: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
AddRecentCall Available since: 8.1.300.17	RecentIndex: ObjectTC<ref name="ObjectTC">Genesyslab.Desktop.Modules.TeamCommunicator.ObjectTC<
Chain InteractionEmailIsPossibleToClose	
IsPossibleToClose	CommandParameter: IInteractionEmail<ref name="IInteractionEmail" />
Chain InteractionEmailLoad	
Load	CommandParameter: IInteractionEmail<ref name="IInteractionEmail" />
Chain InteractionEmailMarkDoneById	
NoChildrenOpened	CommandParameter: String<ref name="String" />

Default Commands	Parameters
	/> - WarningMessageTarget: String<ref name="String" /> - DoNotCheckForParentId: String<ref name="String" /> Available since: 8.1.301.xx
Confirm	- ActionToConfirm: String<ref name="String" /> - ConfirmParentWindow: Window<ref name="Window" />
MarkDone	- CommandParameter: String<ref name="String" /> - WorkbinId: String<ref name="String" />
Chain InteractionEmailMoveToWorkbin	
IsWorkbinDefined Deprecated since: 8.1.300.17	- WorkbinId: String<ref name="String" /> - WorkbinOptionName: String<ref name="String" /> - CommandParameter: IInteractionEmail<ref name="IInteractionEmail" />
IsWorkbinDestinationDefined Available since: 8.1.300.17	- WorkbinId: String<ref name="String" /> - WorkbinOptionName: String<ref name="String" /> - CommandParameter: IInteractionOpenMedia<ref name="IInteractionOpenMedia" />
SetAttachedDadalInformation	CommandParameter: IInteractionOpenMedia<ref name="IInteractionOpenMedia" />
Save	CommandParameter: IInteractionEmail<ref name="IInteractionEmail" />
MoveToWorkbin	CommandParameter: String<ref name="String" />

Default Commands	Parameters
	WorkbinId: String<ref name="String" />
Chain InteractionEmailPlaceInQueue	
UpdateInteractionUCS Available since: 8.1.300.17	- CommandParameter: IInteraction<ref name="Interaction" /> - NotepadViewModel: INotepadViewModel<ref name="INotepadViewModel" />
Save	CommandParameter: IInteractionEmail<ref name="InteractionEmail" />
SetAutoClose	CommandParameter: InteractionOpenMedia<ref name="InteractionOpenMedia" /> Genesyslab.Desktop.Modules.OpenMedia.Model.Interactions.IInteractionOpenMedia
SaveAttachedData Available since: 8.1.100.14	- CommandParameter: InteractionOpenMedia<ref name="InteractionOpenMedia" /> - UserData: KeyValueCollection<ref name="KeyValueCollection" />
PlaceInQueue	- CommandParameter: InteractionOpenMedia<ref name="InteractionOpenMedia" /> - Destination: String<ref name="String" /> - Reasons: KeyValueCollection<ref name="KeyValueCollection">Genesyslab.Enterprise.Commons.Collections.IReasonCollection - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
AddRecentCall Available since: 8.1.301.xx	RecentIndex: ObjectTC<ref name="ObjectTC" />
Chain InteractionEmailPrint Available since: 8.1.300.17	
Print	CommandParameter: IInteractionEmail<ref name="InteractionEmail" />
Chain InteractionEmailPullById Available since: 8.1.300.17	

Default Commands	Parameters
PullInteraction	- CommandParameter: IInteraction<ref name="Interaction" /> - IW_WebcallbackInteractionId: String<ref name="String" />
Chain InteractionEmailPullFromWorkbinById	
NoChildrenOpened	- CommandParameter: String<ref name="String" /> - WarningMessageTarget: String<ref name="String" /> - DoNotCheckForParentId: String<ref name="String" /> Available since: 8.1.301.xx
NoDraftWithSameParentOpened	ParentId: String<ref name="String" />
PullFromWorkbin	- CommandParameter: String<ref name="String" /> - WorkbinId: String<ref name="String" />
MoveToWorkbinOpenedParent	ParentId: String<ref name="String" />
Chain InteractionEmailReply	
RequestSaveNotepad	- CommandParameter: IInteraction<ref name="Interaction" /> - NotepadViewModel: INotepadViewModel<ref name="INotepadViewModel" />
IsPossibleToCreate	CommandParameter: IInteractionInboundEmail<ref name="InteractionInboundEmail" /> Genesyslab.Desktop.Modules.OpenMedia.Model.Interactions.En
*IsInboundEmailWorkbinDefined Deprecated since: 8.1.300.17	CommandParameter: IInteractionEmail<ref name="InteractionEmail" />
IsWorkbinDestinationDefined Available since: 8.1.300.17	- WorkbinId: String<ref name="String" /> - WorkbinOptionName: String<ref name="String" />

Default Commands	Parameters
	/> CommandParameter: InteractionOpenMedia<ref name="InteractionOpenMedia" />
AddSignatureParameters Available since: 8.1.200.16	CommandParameter: InteractionInboundEmail<ref name="InteractionInboundEmail" />
AddResponseSignatureParameters Available since: 8.1.200.16	- SignatureOptionValue: String<ref name="String" /> - ESDKInboundEmail: Interaction<ref name="Interaction" > Genesyslab.Enterprise.Model.Interaction.Interaction</ref>
Reply	- CommandParameter: InteractionInboundEmail<ref name="InteractionInboundEmail" /> - ReplyAll: String<ref name="String" /> Available since: 8.1.200.16 - UserData: KeyValueCollection<ref name="KeyValueCollection" /> Available since: 8.1.200.16 - SignatureParameters: Dictionary<ref name="Dictionary" >System.Collections.Generic.Dictionary<string, object></ref> Available since: 8.1.200.16
InboundEmailSearchAndMoveToWorkbin	CommandParameter: InteractionEmail<ref name="InteractionEmail" />
Chain InteractionEmailReplyById	
IsPossibleToCreate	CommandParameter: InteractionInboundEmail<ref name="InteractionInboundEmail" />
NoChildrenOpened	- CommandParameter: String<ref name="String" /> - WarningMessageTarget: String<ref name="String" />

Default Commands	Parameters
	DoNotCheckForParentId: String<ref name="String" /> Available since: 8.1.301.xx
IsNotOpened	- CommandParameter: String<ref name="String" /> - OpenedInteractionCommand: String<ref name="String" />
AddSignatureParameters Available since: 8.1.200.16	CommandParameter: String<ref name="String" />
AddResponseSignatureParameters Available since: 8.1.200.16	- SignatureOptionValue: String<ref name="String" /> - ESDKInboundEmail: IInteraction<ref name="Interaction" />
Reply	- CommandParameter: String<ref name="String" /> - WorkbinId: String<ref name="String" /> Available since: 8.1.300.17 - ReplyAll: String<ref name="String" /> Available since: 8.1.200.16 - UserData: KeyValueCollection<ref name="KeyValueCollection" /> - SignatureParameters: Dictionary<ref name="Dictionary" /> Available since: 8.1.200.16 - ESDKInboundEmail: IEmailInteraction<ref name="IEmailInteraction" /> Genesyslab.Enterprise.Model.IEmailInteraction</ref> Available since: 8.1.200.16
Chain InteractionEmailResend Available since: 8.1.300.17	
IsPossibleToCreate	CommandParameter: IInteractionInboundEmail<ref name="InteractionInboundEmail" />
NoChildrenOpened	CommandParameter: String<ref name="String" />

Default Commands	Parameters
	- WarningMessageTarget: String<ref name="String" /> - DoNotCheckForParentId: String<ref name="String" />
IsNotOpened	- CommandParameter: String<ref name="String" /> - OpenedInteractionCommand: String<ref name="String" />
AddSignatureParameters	CommandParameter: String<ref name="String" />
AddResponseSignatureParameters Available since: 8.1.301.xx	- SignatureOptionValue: String<ref name="String" /> - ESDKInboundEmail: IInteraction<ref name="IInteraction" />
Create	- CommandParameter: String<ref name="String" /> - ParentInteractionId: String<ref name="String" /> - InitialEmail: IMediaOpenMedia<ref name="IMediaOpenMedia" > Genesyslab.Desktop.Modules.OpenMedia.Model.Agents.IMediaOpenMedia /> - InitialEmailId: String<ref name="String" /> - UserData: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" /> - SignatureParameters: Dictionary<string, object><ref name="DictionaryStringObject">Dictionary<string, object> </ref>
Chain InteractionEmailSave	
Save	CommandParameter: IInteractionEmail<ref name="IInteractionEmail" />
Chain InteractionEmailSaveNewOutbound Available since: 8.1.200.16	

Default Commands	Parameters
Save	CommandParameter: IInteractionEmail<ref name="IInteractionEmail" />
Chain InteractionEmailSend	
RequestSaveNotepad	- CommandParameter: IInteraction<ref name="IInteraction">Genesyslab.Desktop.Modules.Core.Model.Interaction</ref> - NotepadViewModel: INotepadViewModel<ref name="INotepadViewModel">Genesyslab.Desktop.Modules.Windows.Interactions.INotepadViewModel</ref>
IsPossibleToSend	CommandParameter: IInteractionOutboundEmail<ref name="IInteractionOutboundEmail" /> Genesyslab.Desktop.Modules.OpenMedia.Model.Interactions.IInteractionOutboundEmail</ref>
IsPossibleToClose	CommandParameter: CommandParameter<ref name="CommandParameter" /> CommandParameter</ref>
AddQARewiewData Available since: 8.1.300.17	CommandParameter: IInteractionEmail<ref name="IInteractionEmail" />
SetAttachedDataInformation Available in: 8.1.000.16; deprecated in: 8.1.100.02	CommandParameter: IInteractionOpenMedia<ref name="IInteractionOpenMedia" />
Save	CommandParameter: IInteractionEmail<ref name="IInteractionEmail" />
Send	- CommandParameter: IInteractionOutboundEmail<ref name="IInteractionOutboundEmail" /> - Reasons: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
Chain InteractionEmailSendInterimReply	
IsPossibleToSend	CommandParameter: IInteractionOutboundEmail<ref name="IInteractionOutboundEmail" />

Default Commands	Parameters
	"InteractionOutboundEmail" />
IsPossibleToClose	CommandParameter: CommandParameter<ref name="CommandParameter" />InteractionEmail<ref name="InteractionEmail" /> Modified in: 8.1.300.17
AddQAReviewData Available since: 8.1.300.17	CommandParameter: InteractionEmail<ref name="InteractionEmail" />
SetAttachedDadInformation	CommandParameter: InteractionOpenMedia<ref name="InteractionOpenMedia" />
Save	CommandParameter: InteractionEmail<ref name="InteractionEmail" />
PullParentInboundEmail	- CommandParameter: InteractionEmail<ref name="InteractionEmail" /> - DispositionCode: IDispositionCode<ref name="IDispositionCode" />
Send	- CommandParameter: InteractionOutboundEmail<ref name="InteractionOutboundEmail" /> - Reasons: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
Chain InteractionEmailTransferAgent	
Save	CommandParameter: InteractionEmail<ref name="InteractionEmail" />
SetAutoClose	CommandParameter: InteractionOpenMedia<ref name="InteractionOpenMedia" />
RequestSaveNotepad	CommandParameter: Interaction<ref name="

Default Commands	Parameters
	"Interaction" /> NotepadViewModel: INotepadViewModel<ref name="INotepadViewModel" />
TransferAgent	- CommandParameter: InteractionOpenMedia<ref name="InteractionOpenMedia" /> - Destination: String<ref name="String" /> - Reasons: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
AddRecentCall Available since: 8.1.301.xx	RecentIndex: ObjectTC<ref name="ObjectTC" />
Chain InteractionEmailWorkflow	
PromptDone Available since: 8.1.300.17	- CommandParameter: InteractionOpenMedia<ref name="InteractionOpenMedia" /> - ConfirmParentWindow: Window<ref name="Window" >System.Window.Window</ref>
Workflow	- CommandParameter: InteractionOpenMedia<ref name="InteractionOpenMedia" /> - Workflow: String<ref name="String" /> - Reasons: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
Chain InteractionEmailWorkflowFromInteractionESDK	
ESDKWorkflow	- CommandParameter: IIOpenMediaInteraction<ref name="IIOpenMediaInteraction" >Genesyslab.Enterprise.Model.Interaction.IIOpenMediaInteraction</ref> - Workflow: String<ref name="String" /> - Reasons: KeyValueCollection<ref name="KeyValueCollection" />

Default Commands	Parameters
	- Extensions: <code>KeyValuePairCollection<ref name="KeyValuePairCollection" /></code> - AttachedDataInformation: <code>IDispositionCode<ref name="IDispositionCode" /></code> - DispositionCode: <code>KeyValuePairCollection<ref name="KeyValuePairCollection" /></code>
Chain InteractionInboundEmailPossibleToMoveToWorkbin	
IsInboundEmailWorkbinDefined Deprecated since: 8.1.300.17	CommandParameter: <code>IInteractionEmail<ref name="IInteractionEmail" /></code>
IsWorkbinDestinationDefined Available since: 8.1.300.17	- WorkbinId: <code>String<ref name="String" /></code> - WorkbinOptionName: <code>String<ref name="String" /></code> - CommandParameter: <code>IInteractionOpenMedia<ref name="IInteractionOpenMedia" /></code>
Chain InteractionInboundEmailSearchAndMoveToWorkbin	
InboundEmailSearchAndMoveToWorkbin	CommandParameter: <code>IInteractionEmail<ref name="IInteractionEmail" /></code>
Chain InteractionOutboundEmailPossibleToMoveToWorkbin	
IsOutboundEmailWorkbinDefined Deprecated since: 8.1.300.17	CommandParameter: <code>IInteractionEmail<ref name="IInteractionEmail" /></code>
IsWorkbinDestinationDefined Available since: 8.1.300.17	- WorkbinId: <code>String<ref name="String" /></code> - WorkbinOptionName: <code>String<ref name="String" /></code> - CommandParameter: <code>IInteractionOpenMedia<ref name="IInteractionOpenMedia" /></code>
Chain InteractionOutboundEmailSearchAndMoveToWorkbin	
OutboundEmailSearchAndMoveToWorkbin	CommandParameter: <code>IInteractionEmail<ref name="IInteractionEmail" /></code>
Commands common to all interactions	

Default Commands	Parameters
Chain InteractionSetAttachedDataWithDelay	
SetAttachedData	- CommandParameter: IInteraction<ref name="Interaction" /> - AttachedData: KeyValueCollection<ref name="KeyValueCollection" />
Chain BundleClose	
GetAttachedDataInformationUCS	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" > Genesyslab.Desktop.Modules.Core.Model.Interactions.IInteractionsBundle</ref>
UpdateNotePadForVoice Available since: 8.1.300.17	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />
ResetInteractionChatConsultation	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />
IsPossibleToClose	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />
CompleteDispositionCodeOnBundle	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />
Close	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />
StopInteractionVoiceUCS	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />
GetOutboundPreviewRecord	CommandParameter: ICampaign<ref name="ICampaign" > Genesyslab.Enterprise.Model.Interaction.ICampaign</ref>

<references />

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.

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	
• IsPossibleToClose	• CommandParameter: IInteractionIM<ref name="InteractionIM" > Genesyslab.Desktop.Modules.Voice.Model.Interactions.IInteract
• Close	• CommandParameter: IInteractionIM<ref name="InteractionIM" />
Chain InteractionIMIsPossibleToClose	
• IsPossibleToClose	• CommandParameter: IInteractionIM<ref name="InteractionIM" />
Chain InteractionIMReleaseIM Available since: 8.1.200.16	

Default Commands	Parameters
PromptEnd	- CommandParameter: IInteraction<ref name="Interaction">Interaction</ref> - ConfirmParentWindow: Window<ref name="Window">Window</ref>
Release	- CommandParameter: IInteractionIM<ref name="InteractionIM" /> - Reasons: KeyValueCollection<ref name="KeyValueCollection" > Genesyslab.Enterprise.Commons.Collections.KeyValueCollection</ref> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
Chain InteractionIMSendMessage	
SendMessage	- CommandParameter: IInteractionIM<ref name="InteractionIM" /> - Content: string<ref name="string" > System.String</ref> - ContentType: string<ref name="string" />
Commands common to all interactions	
Chain InteractionSetAttachedDataWithDelay	
SetAttachedData	- CommandParameter: IInteraction<ref name="Interaction" /> - AttachedData: KeyValueCollection<ref name="KeyValueCollection" />
Chain BundleClose	
GetAttachedDataInformationUCS	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" > Genesyslab.Desktop.Modules.Core.Model.Interactions.IInteractionsBundle</ref>
UpdateNotePadForVoice Available since: 8.1.300.17	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />
ResetInteractionChatConsultation	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />

Default Commands	Parameters
IsPossibleToClose	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />
CompleteDispositionCodeOnBundle	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />
Close	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />
StopInteractionVoiceUCS	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />
GetOutboundPreviewRecord	CommandParameter: ICampaign<ref name="ICampaign" > Genesyslab.Enterprise.Model.Interaction.ICampaign</ref>

<references />

Open Media



Purpose: To provide information about customizable commands.
Updated in: 8.1.300.17

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.

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 Available since: 8.1.300.17	
• CanPullInteraction	• CommandParameter: Dictionary<ref name="Dictionary" >Dictionary<string, object></ref>
Chain InteractionOpenMediaClose	
• Close	• CommandParameter: InteractionOpenMedia<ref name="InteractionOpenMedia" />

Default commands	Parameters
Chain InteractionOpenMediaCloseOnOutOfService	
UpdateInteractionUCS	CommandParameter: IInteraction<ref name="Interaction" > Genesyslab.Desktop.Modules.Core.Model.Interactions.IInteracti
SmsReleaseParty Available since: 8.1.100.14	CommandParameter: IInteractionSms<ref name="InteractionSms" >Genesyslab.Desktop.Modules.OpenMedia.Model.Interactions.S
ReleaseMyParty	- CommandParameter: IInteractionChat<ref name="InteractionChat" > Genesyslab.Desktop.Modules.OpenMedia.Model.Interactions.CH - AfterActionMode: String <ref name="String" >System.String</ref>
CloseConsult	CommandParameter: IInteractionChat<ref name="IInteractionChat" />
Close	CommandParameter: IInteractionOpenMedia<ref name="InteractionOpenMedia" > Genesyslab.Desktop.Modules.OpenMedia.Model.Interactions.IIn
Chain InteractionOpenMediaConferenceAgent	
ConferenceAgent	- CommandParameter: IInteractionOpenMedia<ref name="InteractionOpenMedia" /> - Visibility: String<ref name="String" /> - Destination: String<ref name="String" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" > Genesyslab.Enterprise.Commons.Collections.KeyValueCollection
Chain InteractionOpenMediaGetInteractionStateById Available since: 8.1.300.17	
GetInteractionState	CommandParameter: String<ref name="String" />
Chain InteractionOpenMediaHandlingTimeout	
UpdateInteractionUCS	CommandParameter: IInteraction<ref name="

Default commands	Parameters
	"Interaction" />
SmsReleaseParty Available since: 8.1.100.14	CommandParameter: IInteractionSms<ref name="InteractionSms" />
ReleaseMyParty	- CommandParameter: IInteractionChat<ref name="InteractionChat" /> - AfterActionMode: String<ref name="String" />
CloseConsult	CommandParameter: IInteractionChat<ref name="InteractionChat" />
Close	CommandParameter: IInteractionOpenMedia<ref name="InteractionOpenMedia" />
Chain InteractionOpenMediaPlaceInQueue	
UpdateInteractionUCS	- CommandParameter: IInteractionOpenMedia<ref name="InteractionOpenMedia" /> - NotepadViewModel: INotepadViewModel<ref name="INotepadViewModel" /> Parameter available since: 8.1.300.17
SaveAttachedData Available since: 8.1.100.14	- CommandParameter: IInteraction<ref name="Interaction" /> - UserData: KeyValueCollection<ref name="KeyValueCollection" />
SetAutoClose Available since: 8.1.300.17	CommandParameter: IInteractionOpenMedia<ref name="InteractionOpenMedia" />
PlaceInQueue	- CommandParameter: IInteractionOpenMedia<ref name="InteractionOpenMedia" /> - Destination: String<ref name="String" /> - Reasons: KeyValueCollection<ref name="KeyValueCollection" />

Default commands	Parameters
	Extensions: <code>KeyValuePairCollection<ref name="KeyValuePairCollection" /></code>
<code>*Close</code> Deprecated since: 8.1.300.17	<code>*CommandParameter: InteractionOpenMedia<ref name="InteractionOpenMedia" /></code>
AddRecentCall	RecentIndex: <code>ObjectTC<ref name="ObjectTC" > Genesyslab.Desktop.Modules.TeamCommunicator.ObjectTC</ref></code>
Chain <code>InteractionOpenMediaPullById</code> Available since: 8.1.300.17	
PullInteraction	- CommandParameter: <code>String<ref name="String" /></code> - queue: <code>String<ref name="String" /></code> - sourceType: <code>String<ref name="String" /></code> - agentId: <code>String<ref name="String" /></code>
Chain <code>InteractionOpenMediaPullFromWorkbinById</code> Available since: 8.1.300.17	
GetInteractionState	CommandParameter: <code>String<ref name="String" /></code>
CanPullInteraction	CommandParameter: <code>Dictionary<ref name="Dictionary" /></code>
DisplayWarningPullNotPossible	canPullInteractionresult: <code>String<ref name="String" /></code>
PullFromWorkbin	- CommandParameter: <code>String<ref name="String" /></code> - WorkbinId: <code>String<ref name="String" /></code>
Chain <code>InteractionOpenMediaUCSStopProcessing</code>	
StopProcessing	- CommandParameter: <code>String<ref name="String" /></code> - UserData: <code>KeyValuePairCollection<ref name="KeyValuePairCollection" /></code>
Chain <code>InteractionOpenMediaUCSUpdate</code>	

Default commands	Parameters
UpdateInteraction	CommandParameter: IInteraction<ref name="Interaction" />
Chain InteractionOpenMediaTransferAgent	
SetAutoClose	CommandParameter: IInteractionOpenMedia<ref name="InteractionOpenMedia" />
RequestSaveNotepad	- CommandParameter: IInteraction<ref name="Interaction" /> - NotepadViewModel: INotepadViewModel<ref name="INotepadViewModel" > Genesyslab.Desktop.Modules.Windows.Interactions.INotepadVi
TransferAgent	- CommandParameter: IInteractionOpenMedia<ref name="InteractionOpenMedia" /> - Destination: String<ref name="String" /> - Reasons: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
AddRecentCall	RecentIndex: ObjectTC<ref name="ObjectTC" />
Chain InteractionOpenMediaWorkflow	
PromptDone Available since: 8.1.300.17	- CommandParameter: IInteractionOpenMedia<ref name="InteractionOpenMedia" /> - ConfirmParentWindow: Window<ref name="Window" >System.Window.Window</ref>
Workflow	- CommandParameter: IInteractionOpenMedia<ref name="InteractionOpenMedia" /> - Workflow: String<ref name="String" /> - Reasons: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="

Default commands	Parameters
	"KeyValueCollection" />
Chain InteractionPossibleToMoveToWorkbin	
*IsWorkbinDefined Deprecated since: 8.1.300.17	*CommandParameter: IInteractionOpenMedia<ref name="IInteractionOpenMedia" />
IsWorkbinDestinationDefined Available since: 8.1.300.17	- WorkbinId: String<ref name="String" /> - WorkbinOptionName: String<ref name="String" /> - CommandParameter: IInteractionOpenMedia<ref name="IInteractionOpenMedia" />
Commands common to all interactions	
Chain InteractionSetAttachedDataWithDelay	
SetAttachedData	- CommandParameter: IInteraction<ref name="IInteraction" /> - AttachedData: KeyValueCollection<ref name="KeyValueCollection" />
Chain BundleClose	
GetAttachedDataInformationUCS	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" > Genesyslab.Desktop.Modules.Core.Model.Interactions.IInteracti
UpdateNotePadForVoice Available since: 8.1.300.17	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />
ResetInteractionChatConsultation	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />
IsPossibleToClose	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />
CompleteDispositionCodeOnBundle	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />

Default commands	Parameters
Close	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />
StopInteractionVoiceUCS	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />
GetOutboundPreviewRecord	CommandParameter: ICampaign<ref name="ICampaign" > Genesyslab.Enterprise.Model.Interaction.ICampaign</ref>

<references />

Outbound



Purpose: To provide information about customizable commands.

Available since: 8.1.100.14

Updated in: 8.1.300.17

Learn about Outbound Commands

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

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

Managing Chains and Records

The following commands cover:

Available since: 8.1.100.14

Default Commands	Parameters
Chain CampaignGetPreviewRecord	
GetOutboundPreviewRecord	CommandParameter: ICampaign<ref name="ICampaign">Genesyslab.Enterprise.Model.Interaction.ICar
Chain CancelOutboundChain	
UpdateOutboundRecord	CommandParameter: IInteraction<ref name="IInteraction">Genesyslab.Desktop.Modules.Core.Model.Int
CancelOutboundChain	
Chain ClosePullPreviewRecord	
RescheduleOutboundRecord	CommandParameter: IInteraction<ref name="IInteraction">_</ref>
CancelOutboundChain	
RejectOutboundRecord	

Default Commands	Parameters
MarkProcessedOutboundChain	CommandParameter: IInteraction<ref name="Interaction">_</ref>
ClosePullPreviewRecord	CommandParameter: IInteractionPullPreview<ref name="InteractionPullPreview">Genesyslab.Desktop.Modules.Outbound</ref>
GetOutboundPreviewRecord Deprecated since 8.1.300.17	CommandParameter: ICampaign<ref name="ICampaign">_</ref>
Chain MarkProcessedOutboundChain	
MarkProcessedOutboundChain	CommandParameter: IInteraction<ref name="Interaction">_</ref>
Chain PossibleToClosePullPreview	
PossibleToClosePullPreview	
Chain PossibleToClosePushPreview	
PossibleToClosePushPreview	
Chain RejectOutboundChain	
UpdateOutboundRecord	CommandParameter: IInteraction<ref name="Interaction">_</ref>
RejectOutboundChain	
Chain StartPreviewRecordOutbound	
StartPreviewRecord	
Chain UpdateOutboundRecord	
UpdateOutboundRecord	CommandParameter: IInteraction<ref name="Interaction">_</ref>

Managing Outbound Interactions

Default Commands	Parameters
Chain InteractionPushPreviewAccept	
AcceptPushPreview	- CommandParameter: <code>InteractionOpenMedia<ref name="InteractionOpenMedia">Genesyslab.Desktop.Modules.OpenMedia</ref></code> - Extensions: <code>KeyValueCollection<ref name="KeyValueCollection">Genesyslab.Enterprise.Commons.Collections.KeyValueCollection</ref></code>
Chain InteractionPushPreviewDecline	
DeclinePushPreview	- CommandParameter: <code>InteractionOpenMedia<ref name="InteractionOpenMedia"/></code> - Reasons: <code>KeyValueCollection<ref name="KeyValueCollection">Genesyslab.Enterprise.Commons.Collections.KeyValueCollection</ref></code> - Extensions: <code>KeyValueCollection<ref name="KeyValueCollection"/></code>
Chain InteractionPushPreviewWorkflow	
Workflow	- CommandParameter: <code>InteractionOpenMedia<ref name="InteractionOpenMedia"/></code> - Workflow: <code>String<ref name="String">System.String</ref></code> - Reasons: <code>KeyValueCollection<ref name="KeyValueCollection" /></code> - Extensions: <code>KeyValueCollection<ref name="KeyValueCollection" /></code>
RescheduleOutboundRecord Available since: 8.1.200.16	CommandParameter: <code>Interaction<ref name="Interaction"/></code>
CancelOutboundChain	
RejectOutboundRecord	
MarkProcessedOutboundChain Available since: 8.1.200.16	CommandParameter: <code>Interaction<ref name="Interaction">_</ref></code>

Default Commands	Parameters
ClosePushPreviewRecord	CommandParameter: InteractionPushPreview<ref name= "InteractionPushPreview">Genesyslab.Desktop.Modules.Outbo

References

<references />

Place and Media



Purpose: To provide information about customizable commands which modify the media of a given place.
Updated in: 8.1.300.17

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.

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.

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

Default Commands	Parameters
Chain MediaOpenMediaChangeStateReason	
• ChangeStateReason	• EnterpriseAgent: IAgent<ref name="IAgent" > Genesyslab.Enterprise.Agent.IAgent</ref> • Device: IDevice<ref name="IDevice" > Genesyslab.Enterprise.Model.Device.IDevice</ref> • Reason: KeyValueCollection<ref name="KeyValueCollection" > Genesyslab.Enterprise.Commons.Collections.KeyValueCollection</ref> • Extensions: KeyValueCollection<ref name="KeyValueCollection" />

Default Commands	Parameters
Chain MediaOpenMediaDNDOff	
DNDOff	- EnterpriseAgent: IAgent<ref name="IAgent" /> - Channel: IClientChanel<ref name="IClientChanel" > Genesyslab.Enterprise.Model.Channel.IClientChanel</ref> - Reason: KeyValueCollection { reasonName<ref name="KeyValueCollection { reasonName" > Genesyslab.Enterprise.Commons.Collections.KeyValueCollection { reasonName</ref> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
Chain MediaOpenMediaDNDOn	
DNDOn	- EnterpriseAgent: IAgent<ref name="IAgent" /> - Channel: IClientChanel<ref name="IClientChanel" /> - Reason: KeyValueCollection { reasonName<ref name="KeyValueCollection { reasonName" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
Chain MediaOpenMediaLogOff	
LogOff	- Channel: IClientChanel<ref name="IClientChanel" /> - DevicePlace: IDevice<ref name="IDevice" /> - Device: IDevice<ref name="IDevice" /> - Reason: KeyValueCollection { reasonName<ref name="KeyValueCollection { reasonName" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
Chain MediaOpenMediaLogOffPlace	
LogOffPlace	- Channel: IClientChanel<ref name="IClientChanel" /> - EnterpriseAgent: IAgent<ref name="IAgent" /> - Reason: KeyValueCollection { reasonName<ref name="KeyValueCollection { reasonName" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />

Default Commands	Parameters
Chain MediaOpenMediaLogOn	
UpdateInteractionSignature	
LogOn	- Channel: IClientChanel<ref name="IClientChanel" /> - DevicePlace: IDevice<ref name="IDevice" /> - Device: IDevice<ref name="IDevice" /> - Reason: KeyValueCollection { reasonName<ref name="KeyValueCollection { reasonName" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
Chain MediaOpenMediaLogOnPlace	
UpdateInteractionSignature	
LogOnPlace	- EnterpriseAgent: IAgent<ref name="IAgent" /> - Channel: IClientChanel<ref name="IClientChanel" /> - Reason: KeyValueCollection { reasonName<ref name="KeyValueCollection { reasonName" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
WorkbinInitialize Deprecated since: 8.1.300.17	ReturnCommandValue: String<ref name="String" -> System.String</ref>
Chain MediaOpenMediaNotReady	
UpdateInteractionSignature	
NotReady	- Channel: IClientChanel<ref name="IClientChanel" /> - DevicePlace: IDevice<ref name="IDevice" /> - Device: IDevice<ref name="IDevice" /> - Reason: KeyValueCollection { reasonName<ref name="KeyValueCollection { reasonName" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />

Default Commands	Parameters
Chain MediaOpenMediaReady	
UpdateInteractionSignature	
Ready	- Channel: IClientChanel<ref name="IClientChanel" /> - DevicePlace: IDevice<ref name="IDevice" /> - Device: IDevice<ref name="IDevice" /> - Reason: KeyValueCollection { reasonName<ref name="KeyValueCollection { reasonName" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />

Manage SMS Media

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

Default Commands	Parameters
Chain MediaSmsCreateNewOutboundPageMode Available since: 8.1.100.14	
IsPossibleToCreate Available since: 8.1.100.14	SourcePhoneNumber: String<ref name="String" />
ThreadingOpenMedia Available since: 8.1.100.14	ParentInteractionId: String<ref name="String" />
Create Available since: 8.1.100.14	- CommandParameter: IMediaOpenMedia<ref name="IMediaOpenMedia">Genesyslab.Desktop.Modules.OpenMedia.Model.Agents.IMediaOpenMedia</ref> - Subtype: String<ref name="String" /> - ParentInteractionId: String (ESDK Id)<ref name="String" /> - SourcePhoneNumber: String<ref name="String" /> - DestinationPhoneNumber: String<ref name="String" /> - InputQueues: ICollection<string><ref name="ICollection" />

Default Commands	Parameters
	<code>>System.Collections.Generic.ICollection</ref></code> - OutputQueues: <code>ICollection<string><ref name="ICollection" /></code> - UserData: <code>KeyValueCollection<ref name="KeyValueCollection" /></code> - Extensions: <code>KeyValueCollection<ref name="KeyValueCollection" /></code> - CorrelatorData: <code>String<ref name="String" /></code>
AddRecentCall Available since: 8.1.100.14	RecentIndex: <code>ObjectTC<ref name="ObjectTC">ObjectTC</ref></code>

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	
IsPossibleToCreate	CommandParameter: <code>InteractionInboundEmail<ref name="InteractionInboundEmail">InteractionInboundEmail</ref></code>
AddSignatureParameters Available since: 8.1.200.16	CommandParameter: <code>IMediaOpenMedia<ref name="IMediaOpenMedia" /></code>
AddResponseSignatureParameters Available since: 8.1.200.16	- SignatureOptionValue: <code>String<ref name="String" /></code> - ESDKInboundEmail: <code>Interaction<ref name="Interaction">Interaction</ref></code>
ThreadingOpenMedia	ParentInteractionId: <code>String<ref name="String" /></code>
Create	- CommandParameter: <code>IMediaOpenMedia<ref name="IMediaOpenMedia" /></code> - ParentInteractionId: <code>String<ref name="String" /></code>

Default Commands	Parameters
	/> - InitialEmail: String<ref name="String" /> Available since: 8.1.300.17 - UserData: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" /> - SignatureParameters: Dictionary<string, object><ref name="Dictionary">Dictionary</ref> Available since: 8.1.200.16
AddRecentCall	RecentIndex: ObjectTC<ref name="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	
DNDOff	- EnterpriseAgent: IAgent<ref name="IAgent" /> - Channel: IClientChanel<ref name="IClientChanel" /> - Reasons: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
Chain MediaVoiceDNDOOn	

Default Commands	Parameters
DNDOn	- EnterpriseAgent: IAgent<ref name="IAgent" /> - Channel: IClientChanel<ref name="IClientChanel" /> - Reasons: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
Chain MediaVoiceLogOff	
NegotiatedLogout Available since: 8.1.100.14	Media: IMedia<ref name="IMedia" > Genesyslab.Desktop.Modules.Core.Model.Agents.IMedia</ref>
TLibUpdateSignature	
LogOff	- EnterpriseAgent: IAgent<ref name="IAgent" /> - Channel: IClientChanel<ref name="IClientChanel" /> - Queue: String<ref name="String" /> - Password: String<ref name="String" /> - Reasons: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
Chain MediaVoiceLogOn	
UpdateTlibSignature	
LogOn	- EnterpriseAgent: IAgent<ref name="IAgent" /> - Channel: IClientChanel<ref name="IClientChanel" /> - Queue: String<ref name="String" /> - WorkMode: Identity<ref name="Identity" > Genesyslab.Enterprise.Model.Identity</ref> - Password: String<ref name="String" /> - Reasons: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />

Default Commands	Parameters
	"KeyValueCollection" />
Chain MediaVoiceMakeCall	
CleanUpInteractionCode	- CommandParameter: IMediaVoice<ref name="IMediaVoice" > Genesyslab.Desktop.Modules.Core.Model.Agents.IMediaVoice< - UserData: KeyValueCollection<ref name="KeyValueCollection" />
ThreadingVoice	UserData: String<ref name="String" />
MakeCall	- CommandParameter: IMediaVoice<ref name="IMediaVoice" /> - Destination: String<ref name="String" /> - Location: String<ref name="String" /> - MakeCallType: MakeCallType<ref name="MakeCallType" > Genesyslab.Enterprise.Interaction.MakeCallType</ref> - UserData: KeyValueCollection<ref name="KeyValueCollection" /> - Reasons: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
AddRecentCall	RecentIndex: ObjectTC<ref name="ObjectTC" />
Chain MediaVoiceMakeIM	
MakeIM	- CommandParameter: IMediaVoice<ref name="IMediaVoice" /> - Destination: String<ref name="String" /> - Location: String<ref name="String" /> - MakeCallType: MakeCallType<ref name="MakeCallType" /> - UserData: KeyValueCollection<ref name="KeyValueCollection" /> - Reasons: KeyValueCollection<ref name="

Default Commands	Parameters
	"KeyValueCollection" /> Extensions: KeyValueCollection<ref name="KeyValueCollection" />
AddRecentCall	RecentIndex: ObjectTC<ref name="ObjectTC" />
Chain MediaVoiceNotReady	
UpdateTLibSignature	
NotReady	- EnterpriseAgent: IAgent<ref name="IAgent" /> - Channel: IClientChanel<ref name="IClientChanel" /> - Queue: String<ref name="String" /> - WorkMode: Identity<ref name="Identity" /> - Reasons: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
Chain MediaVoiceReady	
UpdateTLibSignature	
Ready	- EnterpriseAgent: IAgent<ref name="IAgent" /> - Channel: IClientChanel<ref name="IClientChanel" /> - Queue: String<ref name="String" /> - WorkMode: Identity<ref name="Identity" /> - Reasons: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
Chain MediaVoiceStartIMSession	
Start	

References

<references />

SMS



Purpose: To provide information about customizable commands.

Available since: 8.1.100.14

Updated in: 8.1.300.17

Learn about SMS Commands

These commands manage operations on SMS interactions.

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	
Accept	CommandParameter: InteractionSms<ref name="InteractionSms">Genesyslab.Desktop.Modules.OpenMedia
Chain InteractionSmsCreateReplyMessage	
IsPossibleToCreate	SourcePhoneNumber: String<ref name="String">System.String</ref>
IsItPossibleToSendPageSms	OutboundQueue: String<ref name="String" />
SetContext	CorrelatorData: String<ref name="String" />
Create	- CommandParameter: IMediaOpenMedia<ref name="IMediaOpenMedia">Genesyslab.Desktop.Modules.OpenMedia - Subtype: String<ref name="String" /> - ParentInteractionId: String<ref name="String" />(Enterprise Service ID) - SourcePhoneNumber: String<ref name="String"

Default Commands	Parameters
	/> - DestinationPhoneNumber: String<ref name="String" /> - InputQueues: ICollection<ref name="ICollection">System.Collections.Generic.ICollection</ref> - OutputQueues: ICollection<ref name="ICollection" /> - UserData: KeyValueCollection<ref name="KeyValueCollection">Genesyslab.Enterprise.Commons.Collect - Extensions: KeyValueCollection<ref name="KeyValueCollection" /> - CorrelatorData: String<ref name="String" />
CreateUCS	EsdkInteraction: IInteraction<ref name="Interaction">Genesyslab.Enterprise.Model.Interaction.IInterac - ParentInteractionId: String<ref name="String" />
Chain InteractionSmsDecline	
Decline	- CommandParameter: InteractionOpenMedia<ref name="InteractionOpenMedia">Genesyslab.Desktop.Modules.OpenM - Reasons: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
Close Available since: 8.1.200.16	CommandParameter: InteractionOpenMedia<ref name="InteractionOpenMedia" />
Chain InteractionSmsDelete	
Confirm	- ActionToConfirm: String<ref name="String" /> - ConfirmParentWindow: Window<ref name="Window"> System.Window.Window</ref>
Delete	CommandParameter: InteractionOpenMedia<ref name="InteractionOpenMedia" />

Default Commands	Parameters
Chain InteractionSmsDisconnectSmsSessionChannel	
Disconnect	CommandParameter: IInteractionOpenMedia<ref name="InteractionOpenMedia" />
Chain InteractionSmsGetHistoricMessage	
ExtractInteractions	- CommandParameter: String<ref name="String" /> (contactId) - ListAttributes: StringList<ref name="StringList">Genesyslab.Platform.Contacts.Protocols.ContactSe - MediaTypes: String<ref name="String" /> (list of media type seperated by comma.) - TimeFrame: System (time frame in hour) - ExcludedInteractions: IList<T><ref name="IList">System.Collections.Generic.IList<T></ref> (interaction id) - InteractionsUCS: IList<T><ref name="IList" /> (output parameter).
DecodeInteractions	- InteractionsUCS: IList<T><ref name="IList" /> - SmsMessages: ArrayList <ref name="ArrayList">System.Collections.ArrayList</ref>; the collection contains [ISmsMessage<ref name="ISmsMessage">Genesyslab.Desktop.Modules.Contacts.Decod output parameter].
Processing	- ModelView: ISmsViewModel<ref name="ISmsViewModel">Genesyslab.Desktop.Modules.OpenMedia.W - Interaction: Interaction<ref name="Interaction">Genesyslab.Desktop.Modules.Core.Model.Interac - SmsMessages: ArrayList <ref name="ArrayList">System.Collections.ArrayList</ref>; the collection contains [ISmsMessage<ref name="ISmsMessage">Genesyslab.Desktop.Modules.Contacts.Decod output parameter].
Chain InteractionSmsIsPossibleToClose	
IsPossibleToClose	CommandParameter: IInteractionSms<ref name="InteractionSms">Genesyslab.Desktop.Modules.OpenMedia.M
Chain InteractionSmsPlaceInQueue	

Default Commands	Parameters
UpdateInteractionUCS	- CommandParameter: IInteraction<ref name="Interaction" /> - NotepadViewModel: INotepadViewModel<ref name="INotepadViewModel" /> Available since: 8.1.300.17
SaveAttachedData	- CommandParameter: IInteractionOpenMedia<ref name="InteractionOpenMedia" /> - UserData: KeyValueCollection<ref name="KeyValueCollection" />
SetAutoClosePageMode	CommandParameter: IInteractionOpenMedia<ref name="InteractionOpenMedia" />
PlaceInQueue	- CommandParameter: IInteractionOpenMedia<ref name="InteractionOpenMedia" /> - Destination: String<ref name="String" /> - Reason: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
SetAutoCloseSessionMode	CommandParameter: IInteractionOpenMedia<ref name="InteractionOpenMedia" />
ReleaseParty	CommandParameter: IInteractionSms<ref name="InteractionSms" />
AddRecentCall	RecentIndex: ObjectTC<ref name="ObjectTC" > Genesyslab.Desktop.Modules.TeamCommunicator.ObjectTC</ref>
Chain InteractionSmsSendMessage	
IsDispositionCodeMandatory	CommandParameter: IInteractionSms<ref name="InteractionSms" />
SendMessage	CommandParameter: IInteractionSms<ref

Default Commands	Parameters
	name="InteractionSms" /> • Visibility: String<ref name="String" /> • Message: String<ref name="String" />
• MimizeWindow	• CommandParameter: InteractionSms<ref name="InteractionSms" /> • Window: Window<ref name="Window" />
Chain InteractionSmsSendMessage	
• IsItPossibleToSendPageSms	• OutboundQueue: String<ref name="String" />
• Send	• CommandParameter: InteractionSms<ref name="InteractionSms" /> • OutboundQueue: String<ref name="String" /> • SourcePhoneNumber: String<ref name="String" /> • Reason: KeyValueCollection<ref name="KeyValueCollection" /> • Extensions: KeyValueCollection<ref name="KeyValueCollection" /> • CorrelatorData: String<ref name="String" />
• UpdateInteractionUCS	• CommandParameter: Interaction<ref name="Interaction" />
Chain InteractionSmsSendReplyMessage	
• IsItPossibleToSendPageSms	• OutboundQueue: String<ref name="String" />
• Send	• CommandParameter: ISMSInteraction<ref name="ISMSInteraction" > Genesyslab.Enterprise.Model.Interaction.ISMSInteraction</ref> • OutboundQueue: String<ref name="String" /> • Reason: KeyValueCollection<ref name="KeyValueCollection" /> • Extensions: KeyValueCollection<ref name="KeyValueCollection" /> • CorrelatorData: String<ref name="String" />

Default Commands	Parameters
CloseParentInteraction	ParentInteraction: CommandParameter<ref name="CommandParameter">CommandParameter</ref>
Chain InteractionSmsTransferAgent	
SetAutoClosePageMode	CommandParameter: IInteractionOpenMedia<ref name="InteractionOpenMedia" />
RequestSaveNotepad	- CommandParameter: IInteraction<ref name="Interaction" /> - NotepadViewModel: INotepadViewModel<ref name="INotepadViewModel">Genesyslab.Desktop.Modules.Windows.Interactions.INotepad
TransferAgent	- CommandParameter: IInteractionOpenMedia<ref name="InteractionOpenMedia" /> - Destination: String<ref name="String" /> - Reasons: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
SetAutoCloseSessionMode	CommandParameter: IInteractionOpenMedia<ref name="InteractionOpenMedia" />
ReleaseParty	CommandParameter: IInteractionSms<ref name="InteractionSms" />
AddRecentCall	RecentIndex: ObjectTC<ref name="ObjectTC" />
Chain InteractionSmsWorkflow	
PromptDone Available since: 8.1.300.17	- CommandParameter: IInteractionOpenMedia<ref name="InteractionOpenMedia" /> - ConfirmParentWindow: Window<ref name="Window">System.Window.Window</ref>

Default Commands	Parameters
ReleaseParty	CommandParameter: IInteractionSms<ref name="InteractionSms" />
UpdateInteractionUCS	CommandParameter: IInteraction<ref name="Interaction" />
Workflow	- CommandParameter: IInteractionOpenMedia<ref name="InteractionOpenMedia" /> - Workflow: String<ref name="String" /> - Reasons: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
Commands common to all interactions	
Chain InteractionSetAttachedDataWithDelay	
SetAttachedData	- CommandParameter: IInteraction<ref name="Interaction" /> - AttachedData: KeyValueCollection<ref name="KeyValueCollection" />
Chain BundleClose	
GetAttachedDataInformationUCS	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" > Genesyslab.Desktop.Modules.Core.Model.Interactions.IInteracti
UpdateNotePadForVoice Available since: 8.1.300.17	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />
ResetInteractionChatConsultation	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />
IsPossibleToClose	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />
CompleteDispositionCodeOnBundle	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />

Default Commands	Parameters
Close	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />
StopInteractionVoiceUCS	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />
GetOutboundPreviewRecord	CommandParameter: ICampaign<ref name="ICampaign" > Genesyslab.Enterprise.Model.Interaction.ICampaign</ref>

References

<references />

Supervision



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

Created in: 8.1.300.17

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.

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	
EndMonitoring	
Chain TeamLeadEndMonitoringChat	
EndMonitoringChat	
Chain TeamLeadEndMonitoringVoice	
EndMonitoringVoice	
Chain TeamLeadIntrudeCurrentChat Available since: 8.1.300.17	
IntrudeCurrentChat	
AddRecentCall	RecentIndex: ObjectTC<ref name="ObjectTC"

Default Commands	Parameters
	<code>/></code>
Chain TeamLeadIntrudeCurrentVoice Available since: 8.1.300.17	
IntrudeCurrentVoice	
AddRecentCall	RecentIndex: ObjectTC<ref name="ObjectTC">/>
Chain TeamLeadIsItPossibleToMonitor Available since: 8.1.300.17	
IsItPossibleToMonitor	- CommandParameter: String<ref name="String">System.String</ref> (employeeID) - IsItPossibleToMonitor: Boolean<ref name="Boolean">System.Boolean</ref>, output parameter
Chain TeamLeadMonitorAll	
MonitorAllVoiceCalls	
MonitorAllChatInteractions	
AddRecentCall	RecentIndex: ObjectTC<ref name="ObjectTC">Genesyslab.Desktop.Modules.TeamCommunicator.ObjectTC</ref>
AddToTrackingList	
Chain TeamLeadSetSupervisorBargelInChat	
SetSupervisorBargelInChat	
Chain TeamLeadSetSupervisorBargelInVoice	
SetSupervisorBargelInVoice	
Chain TeamLeadSetSupervisorCoachChat	
SetSupervisorCoachChat	
Chain TeamLeadSetSupervisorCoachVoice	

Default Commands	Parameters
SetSupervisorCoachVoice	
Chain TeamLeadSetSupervisorMuteVoice	
SetSupervisorMuteVoice	
Chain TeamLeadStopMonitor	
StopMonitorVoiceCalls	
StopMonitorChatInteractions	
RemoveFromTrackingList	
AddRecentCall	RecentIndex: ObjectTC<ref name="ObjectTC" />

<references />

References

<references />

UCS



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

Available since: 8.1.100.14

Updated in: 8.1.300.17

Learn about the UCS Commands

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

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 ContactAssignInteractionToContact Available since: 8.1.301.10	
Assign	- DestinationContactId: String<ref name="String" /> - ContactIdBeforeMerge: String<ref name="String" /> - UCSInteractionId: String<ref name="String" /> - AlertSectionName: String<ref name="String" /> - AssignInteractionToContactError: String<ref name="String" />, output parameter
Chain ContactAssignInteractionToContactForCustom Available since: 8.1.301.10	
AssignInteractionToContact	- DestinationContactId: String<ref name="String" /> - ContactIdBeforeMerge: String<ref name="

Default Commands	Parameters
	"String" /> - UCSInteractionId: String<ref name="String" /> - AlertSectionName: String<ref name="String" /> - AssignInteractionToContactError: String<ref name="String" />, output parameter
GetContact	- CommandParameter: String<ref name="String" /> (ContactId) - ContactAttributes: IList<ref name="IList">System.Collections.Generic.IList</ref>, if key is missing get ContactAttributes defined in option 'contact.displayed-attributes' - IWContact: IIWContact<ref name="IIWContact">Genesyslab.Desktop.Modules.Contacts.IWContact</ref>, output parameter
RefreshInformation	- IWContact: IIWContact<ref name="IIWContact">Genesyslab.Desktop.Modules.Contacts.IWContact</ref>, output parameter - Caseld: String<ref name="String" /> - InteractionId : String<ref name="String" />
Chain ContactAttributeInfoOrder	
GetContactAttributeInfoOrder	AttributesListOrdered:IContactAttributeField[], output parameter
Chain ContactCreateUCSInteraction	
Create	- ContactId: String<ref name="String" /> - InteractionToCreate: IInteraction<ref name="IInteraction" /> - InteractionUCSId: String<ref name="String" />, output parameter
Chain CreateInteractionUCSFromESDKInteraction	
CreateUCS	EsdkInteraction: IInteraction<ref name="IInteraction" > Genesyslab.Enterprise.Model.Interaction.IInteraction</ref>

Default Commands	Parameters
	- ParentInteractionId: String<ref name="String">System.String </ref> - StartDate: DateTime<ref name="DateTime">System.DateTime</ref> Parameter available since: 8.1.300.17
Chain DeleteInteractionUCS	
DeleteUCS	CommandParameter: String<ref name="String"/>
Chain GetContactUCS Available since: 8.1.300.17	
GetContact	- CommandParameter: String<ref name="String"/> (ContactId) - ContactAttributes: IList<ref name="IList"/>, if key is missing get ContactAttributes defined in option 'contact.displayed-attributes' - IWContact: IIWContact<ref name="IIWContact">Genesyslab.Desktop.Modules.Contacts.IWContact</ref> output parameter
Chain InteractionUCSInQueue	
GetInteraction	- CommandParameter: String (contactId)<ref name="String"/> - ListAttributes: StringList<ref name="StringList">Genesyslab.Platform.Contacts.Protocols.ContactServer.StringList</ref> - ExcludedInteractions: IList<ref name="IList"/> <String<ref name="String"/>> (list of String interaction id) - CurrentInteraction : IInteraction<ref name="IInteraction"/> Available since: 8.1.300.17 - InteractionsUCS: GenericList<IDictionary<string,object>>, output parameter<ref name="GenericList">System.Collections.GenericList</ref>
Chain SetLastCalledAgentUCS Available since: 8.1.300.17	
SetLCA	

Default Commands	Parameters
Chain ContactIsModified	
IsModified	Caseld: String<ref name="String" >System.String</ref>
Chain ContactNotification	
GetInteraction Available since: 8.1.300.17	- CommandParameter: String (contactId)<ref name="String (contactId)" >System.String (contactId)</ref> - TimeFrame: System (time frame in day)<ref name="System (time frame in day)" >System (time frame in day)</ref> - ListAttributes: StringList<ref name="StringList" >Genesyslab.Platform.Contacts.Protocols.ContactServer.String - ExcludedInteractions: IList<String>(interaction id)<ref name="IList" /> - CurrentInteraction : Interaction<ref name="Interaction" /> - InteractionsUCS: GenericList<IDictionary<string,object>> <ref name="GenericList" >System.Collections.GenericList<IDictionary<string,object>>, output parameter</ref>
Chain ContactOpenSearchWindow	
Open	Parent: UserControl<ref name="UserControl" >System.Windows.Controls.UserControl</ref>
Chain ContactOpenToDialog Available since: 8.1.200.16	
Open	ActionTarget: ActionTarget<ref name="ActionTarget" >Genesyslab.Desktop.Modules.Windows.Interactions.ActionTarget

<references />

Voice



Purpose: To provide information about customizable commands.
Updated in: 8.1.300.17

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](#).

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.

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

Chains of Commands to Manage Voice Interactions

Default Commands	Parameters
Chain InteractionVoiceAlternateCall	
• AlternateCall	• CommandParameter: IInteractionVoice<ref name="IInteractionVoice" > Genesyslab.Desktop.Modules.Core.Model.Interactions.IInteracti • Reasons: KeyValueCollection<ref name="KeyValueCollection" >

Default Commands	Parameters
	Genesyslab.Enterprise.Commons.Collections.KeyValueCollection Extensions: KeyValueCollection<ref name="KeyValueCollection" />
Chain InteractionVoiceAnswerCall	
AnswerCall	- CommandParameter: IInteractionVoice<ref name="IInteractionVoice" /> - Reasons: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
Chain InteractionVoiceAutoCloseConsult	
AutoCloseConsult	CommandParameter: IInteractionVoice<ref name="IInteractionVoice" />
Chain InteractionVoiceBeforeClose	
DoNotCallOutboundChain Available since: 8.1.100.14	CommandParameter: IInteraction<ref name="IInteraction" />
SetCallResultOutboundRecord Available since: 8.1.100.14	CommandParameter: IInteraction<ref name="IInteraction" />
RescheduleOutboundRecord Available since: 8.1.100.14	CommandParameter: IInteraction<ref name="IInteraction" />
UpdateOutboundRecord Available since: 8.1.100.14	CommandParameter: IInteraction<ref name="IInteraction" />
MarkProcessedOutboundChain Available since: 8.1.100.14	CommandParameter: IInteraction<ref name="IInteraction" />
Close Available since: 8.1.100.14	CommandParameter: IInteractionVoice<ref name="IInteractionVoice" />
Chain InteractionVoiceClose	
DoNotCallOutboundChain Available since: 8.1.100.14	CommandParameter: IInteraction<ref name="IInteraction" />

Default Commands	Parameters
SetCallResultOutboundRecord Available since: 8.1.100.14	CommandParameter: IInteraction<ref name="Interaction" />
RescheduleOutboundRecord Available since: 8.1.100.14	CommandParameter: IInteraction<ref name="Interaction" />
UpdateOutboundRecord Available since: 8.1.100.14	CommandParameter: IInteraction<ref name="Interaction" />
MarkProcessedOutboundChain Available since: 8.1.100.14	CommandParameter: IInteraction<ref name="Interaction" />
Close	CommandParameter: IInteractionVoice<ref name="InteractionVoice" />
Chain InteractionVoiceCompleteConference	
CompleteConference	- CommandParameter: IInteractionVoice<ref name="InteractionVoice" /> - Reasons: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
Chain InteractionVoiceCompleteTransfer	
ParkCallback Available since: 8.1.300.17	CommandParameter: IInteraction<ref name="Interaction" />
CompleteTransfer	- CommandParameter: IInteractionVoice<ref name="InteractionVoice" /> - Reasons: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
Chain InteractionVoiceDeleteFromConference	
DeleteFromConference	- CommandParameter: Party<ref name="Party" > Genesyslab.Desktop.Modules.Core.Model.Interactions.Party</ref> - Reasons: KeyValueCollection<ref name="KeyValueCollection" />

Default Commands	Parameters
	Extensions: <code>KeyValueCollection<ref name="KeyValueCollection" /></code>
Chain InteractionVoiceHoldCall	
HoldCall	- CommandParameter: <code>InteractionVoice<ref name="InteractionVoice" /></code> - Reasons: <code>KeyValueCollection<ref name="KeyValueCollection" /></code> - Extensions: <code>KeyValueCollection<ref name="KeyValueCollection" /></code>
Chain InteractionVoiceIfPossibleCloseInteraction	
DoNotCallOutboundChain Available since: 8.1.100.14	CommandParameter: <code>Interaction<ref name="Interaction" /></code>
SetCallResultOutboundRecord Available since: 8.1.100.14	CommandParameter: <code>Interaction<ref name="Interaction" /></code>
RescheduleOutboundRecord Available since: 8.1.100.14	CommandParameter: <code>Interaction<ref name="Interaction" /></code>
UpdateOutboundRecord Available since: 8.1.100.14	CommandParameter: <code>Interaction<ref name="Interaction" /></code>
MarkProcessedOutboundChain Available since: 8.1.100.14	CommandParameter: <code>Interaction<ref name="Interaction" /></code>
GetAttachedDataInformationUCS Available since: 8.1.300.17	CommandParameter: <code>InteractionsBundle<ref name="InteractionsBundle">Genesyslab.Desktop.Modules.Core.Model.Interactions.InteractionsBundle</ref></code>
UpdateNotePad	CommandParameter: <code>InteractionsBundle<ref name="InteractionsBundle" /></code>
IsPossibleToMarkDone	CommandParameter: <code>InteractionVoice<ref name="InteractionVoice" /></code>
CompleteDispositionCode	CommandParameter: <code>Interaction<ref name="Interaction" /></code>

Default Commands	Parameters
	"Interaction" > Genesyslab.Desktop.Modules.Core.Model.Interactions.IInteracti
Close	CommandParameter: IInteractionVoice<ref name="IInteractionVoice" />
StopInteractionVoiceUCS	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />
Chain InteractionVoiceInitConference	
ConferenceNotepad	- CommandParameter: IInteraction<ref name="Interaction" /> - UserData: KeyValueCollection<ref name="KeyValueCollection" />
RequestSaveNotepad	- CommandParameter: IInteraction<ref name="Interaction" /> - NotepadViewModel: INotepadViewModel<ref name="INotepadViewModel" > Genesyslab.Desktop.Modules.Windows.Interactions.INotepadVi
InitConference	- CommandParameter: IInteractionVoice<ref name="IInteractionVoice" /> - Destination: <ref name="String" > System.String</ref> - Location: String<ref name="String" /> - UserData: KeyValueCollection<ref name="KeyValueCollection" /> - Reasons: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
AddRecentCall	RecentIndex: ObjectTC<ref name="ObjectTC" />
Chain InteractionVoiceInitTransfer	
TransferNotepad	CommandParameter: IInteraction<ref name="Interaction" />

Default Commands	Parameters
	UserData: KeyValueCollection<ref name="KeyValueCollection" />
RequestSaveNotepad	- CommandParameter: IInteraction<ref name="IInteraction" /> - NotepadViewModel: INotepadViewModel<ref name="INotepadViewModel" />
InitTransfer	- CommandParameter: IInteractionVoice<ref name="IInteractionVoice" /> - Destination: String<ref name="String" /> - Location: String<ref name="String" /> - UserData: KeyValueCollection<ref name="KeyValueCollection" /> - Reasons: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" /> - CorrelatorData: String<ref name="String" /> Available since: 8.1.300.17
AddRecentCall	RecentIndex: ObjectTC<ref name="ObjectTC" > Genesyslab.Desktop.Modules.TeamCommunicator.ObjectTC</ref>
Chain InteractionVoiceIsPossibleToClose	
IsPossibleToMarkDone	CommandParameter: IInteractionVoice<ref name="IInteractionVoice" />
CompleteDispositionCode	CommandParameter: IInteraction<ref name="IInteraction" />
Chain InteractionVoiceListenDisconnect	
ListenDisconnect	- CommandParameter: Party<ref name="Party" /> - Reasons: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />

Default Commands	Parameters
Chain InteractionVoiceListenReconnect	
ListenReconnect	- CommandParameter: Party<ref name="Party" /> - Reasons: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
Chain InteractionVoiceMuteOff Deprecated since 8.1.2	
MuteOff	- CommandParameter: IInteractionVoice<ref name="IInteractionVoice" /> - Reasons: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
Chain InteractionVoiceMuteOn Deprecated since 8.1.2	
MuteOn	- CommandParameter: IInteractionVoice<ref name="IInteractionVoice" /> - Reasons: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
Chain InteractionVoicePauseRecordCall Available since: 8.1.200.16	
PauseRecordCall	CommandParameter: IInteractionVoice<ref name="IInteractionVoice" />
Chain InteractionVoicePauseRecordScreenCall Available since: 8.1.200.16	
PauseRecordScreenCall	CommandParameter: IInteractionVoice<ref name="IInteractionVoice" />
Chain InteractionVoiceReconnectCall	
PromptEnd Available since: 8.1.200.16	- CommandParameter: IInteraction<ref name="IInteraction" /> - ConfirmParentWindow: Window<ref name="Window">Window</ref>

Default Commands	Parameters
ReconnectCall	- CommandParameter: IInteractionVoice<ref name="IInteractionVoice" /> - Reasons: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
Chain InteractionVoiceRecordCall	
RecordCall	CommandParameter: IInteractionVoice<ref name="IInteractionVoice" />
Chain InteractionVoiceRedirectCall	
RedirectCall	- CommandParameter: IInteractionVoice<ref name="IInteractionVoice" /> - OtherDn: String<ref name="String" /> - OtherDnLocation: String<ref name="String" /> - Reasons: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
Chain InteractionVoiceReleaseCall	
SetSupervisorSilentCommand Available since: 8.1.300.17	
PromptEnd Available since: 8.1.200.16	- CommandParameter: IInteraction<ref name="IInteraction" /> - ConfirmParentWindow: Window<ref name="Window" />
ReleaseCall	- CommandParameter: IInteractionVoice<ref name="IInteractionVoice" /> - Reasons: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
Chain InteractionVoiceResume	

Default Commands	Parameters
Resume	- CommandParameter: <code>InteractionVoice</code><ref name="InteractionVoice" /> - Reasons: <code>KeyValueCollection</code><ref name="KeyValueCollection" /> - Extensions: <code>KeyValueCollection</code><ref name="KeyValueCollection" />
Chain <code>InteractionVoiceResumeRecordCall</code> Available since: 8.1.200.16	
ResumeRecordCall	CommandParameter: <code>InteractionVoice</code><ref name="InteractionVoice" />
Chain <code>InteractionVoiceResumeRecordScreenCall</code> Available since: 8.1.200.16	
ResumeRecordScreenCall	CommandParameter: <code>InteractionVoice</code><ref name="InteractionVoice" />
Chain <code>InteractionVoiceRetrieveCall</code>	
RetrieveCall	- CommandParameter: <code>InteractionVoice</code><ref name="InteractionVoice" /> - Reasons: <code>KeyValueCollection</code><ref name="KeyValueCollection" /> - Extensions: <code>KeyValueCollection</code><ref name="KeyValueCollection" />
Chain <code>InteractionVoiceSendDTMF</code>	
SendDTMF	- CommandParameter: <code>InteractionVoice</code><ref name="InteractionVoice" /> - DtmfDigits: <code>String</code><ref name="String" /> - Reasons: <code>KeyValueCollection</code><ref name="KeyValueCollection" /> - Extensions: <code>KeyValueCollection</code><ref name="KeyValueCollection" />
Chain <code>InteractionVoiceSingleStepConference</code>	
ConferenceSingleStepNotepad	- CommandParameter: <code>Interaction</code><ref name="Interaction" /> - UserData: <code>KeyValueCollection</code><ref name="KeyValueCollection" />

Default Commands	Parameters
RequestSaveNotepad	- CommandParameter: IInteraction<ref name="Interaction" /> - NotepadViewModel: INotepadViewModel<ref name="INotepadViewModel" />
SingleStepConference	- CommandParameter: IInteractionVoice<ref name="IInteractionVoice" /> - Destination: String<ref name="String" /> - Location: String<ref name="String" /> - UserData: KeyValueCollection<ref name="KeyValueCollection" /> - Reasons: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
AddRecentCall	RecentIndex: ObjectTC<ref name="ObjectTC" />
Chain InteractionVoiceSingleStepTransfer	
ParkCallback Available since: 8.1.300.17	CommandParameter: IInteraction<ref name="Interaction" />
TransferSingleStepNotepad	- CommandParameter: IInteraction<ref name="Interaction" /> - UserData: KeyValueCollection<ref name="KeyValueCollection" />
RequestSaveNotepad	- CommandParameter: IInteraction<ref name="Interaction" /> - NotepadViewModel: INotepadViewModel<ref name="INotepadViewModel" />
SingleStepTransfer	- CommandParameter: IInteractionVoice<ref name="IInteractionVoice" /> - Destination: String<ref name="String" /> - Location: String<ref name="String" /> - UserData: KeyValueCollection<ref name="KeyValueCollection" />

Default Commands	Parameters
	"KeyValueCollection" /> - Reasons: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
AddRecentCall	RecentIndex: ObjectTC<ref name="ObjectTC" />
Chain InteractionVoiceStartRecordScreenCall Available since: 8.1.200.16	
StartRecordCaScreenIi	CommandParameter: IInteractionVoice<ref name="IInteractionVoice" />
Chain InteractionVoiceStopRecordCall	
StopRecordCall	CommandParameter: IInteractionVoice<ref name="IInteractionVoice" />
Chain InteractionVoiceStopRecordScreenCall Available since: 8.1.200.16	
StopRecordScreenCall	CommandParameter: IInteractionVoice<ref name="IInteractionVoice" />
Commands common to all interactions	
Chain InteractionSetAttachedDataWithDelay	
SetAttachedData	- CommandParameter: IInteraction<ref name="IInteraction" /> - AttachedData: KeyValueCollection<ref name="KeyValueCollection" />
Chain BundleClose	
GetAttachedDataInformationUCS	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" /> Genesyslab.Desktop.Modules.Core.Model.Interactions.IInteracti
UpdateNotePadForVoice Available since: 8.1.300.17	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />

Default Commands	Parameters
ResetInteractionChatConsultation	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />
IsPossibleToClose	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />
CompleteDispositionCodeOnBundle	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />
Close	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />
StopInteractionVoiceUCS	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />
GetOutboundPreviewRecord	CommandParameter: ICampaign<ref name="ICampaign" > Genesyslab.Enterprise.Model.Interaction.ICampaign</ref>

<references />

Windows



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

Updated in: 8.1.300.17

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.

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	
IsPossibleToClose	CommandParameter: IInteractionsWindow<ref name="IInteractionsWindow" > Genesyslab.Desktop.Modules.Windows.Interactions.IInteraction
Close	CommandParameter: IInteractionsWindow<ref name="IInteractionsWindow" />
Chain WindowInteractionClose	
IsPossibleToClose	CommandParameter: IInteractionsWindow<ref name="IInteractionsWindow" />
Close	CommandParameter: IInteractionsWindow<ref name="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 Available since: 8.1.300.17	
RestoreInteractionsWindow	- CommandParameter: IInteractionsWindow<ref name="InteractionsWindow">Genesyslab.Desktop.Modules.Windows. - Interaction: IInteraction<ref name="Interaction">Genesyslab.Desktop.Modules.Core.Model.Interaction - Case: ICase<ref name="ICase">Genesyslab.Desktop.Modules.Core.Model.Interactions.
Chain ShowInteractionsWindow Available since: 8.1.300.17	
ShowInteractionsWindow	- CommandParameter: IInteractionsWindow<ref name="IInteractionsWindow" /> - Interaction: IInteraction<ref name="Interaction" /> - Case: ICase<ref name="ICase" />

Manage Toaster

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

Chain of Commands to Manage Toaster

Default Commands	Parameters
Chain ShowToasterWindow Available since: 8.1.300.17	
ShowToasterWindow	- CommandParameter: IToasterWindow<ref name="IToasterWindow">Genesyslab.Desktop.Modules.Windows.View - Interaction: IInteraction<ref name="Interaction" />
Chain ToasterAcceptPreview	

Default Commands	Parameters
Accept	CommandParameter: IInteraction<ref name="Interaction" > Genesyslab.Desktop.Modules.Core.Model.Interactions.IInteracti
Chain ToasterDeclinePreview	
Decline	CommandParameter: IInteraction<ref name="Interaction" />
Chain ToasterInteractionAccept	
Accept	CommandParameter: IInteraction<ref name="Interaction" />
Chain ToasterInteractionDecline	
Decline	- CommandParameter: IInteraction<ref name="Interaction" /> - OtherDn: String<ref name="String" > System.String</ref> - OtherDnLocation: String<ref name="String" />
Chain ToasterInteractionEmailAccept	
Accept	- CommandParameter: InteractionOpenMedia<ref name="InteractionOpenMedia" > Genesyslab.Desktop.Modules.OpenMedia.Model.Interactions.II - Extensions: KeyValueCollection<ref name="KeyValueCollection" > Genesyslab.Enterprise.Commons.Collections.KeyValueCollection
Chain ToasterInteractionEmailDecline	
Decline	- CommandParameter: InteractionOpenMedia<ref name="InteractionOpenMedia" /> - Reasons: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
Close Available since: 8.1.200.16	CommandParameter: InteractionOpenMedia<ref name="InteractionOpenMedia" />

Default Commands	Parameters
Chain ToasterInteractionIMAccept	
Accept	CommandParameter: IInteractionIM<ref name="InteractionIM" /> Genesyslab.Desktop.Modules.Voice.Model.Interactions.IInteract
Chain ToasterInteractionIMReject	
Reject	- CommandParameter: IInteractionIM<ref name="InteractionIM" /> - RejectReason: String<ref name="String" /> - Reasons: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
Reject	CommandParameter: IInteraction<ref name="Interaction" />
Chain ToasterInteractionOpenMediaAccept	
Accept	- CommandParameter: IInteractionOpenMedia<ref name="InteractionOpenMedia" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
Chain ToasterInteractionOpenMediaDecline	
Decline	- CommandParameter: IInteractionOpenMedia<ref name="InteractionOpenMedia" /> - Reasons: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
Close Available since: 8.1.200.16	CommandParameter: IInteractionOpenMedia<ref name="InteractionOpenMedia" />
Chain ToasterInteractionPreviewAccept	
Accept	CommandParameter: IInteraction<ref name="Interaction" />

Default Commands	Parameters
	"Interaction" />
Chain ToasterInteractionPreviewDecline	
Decline	CommandParameter: IInteraction<ref name="Interaction" />
Chain ToasterShowBroadcastMessage	
ShowToaster	CommandParameter: IBroadcastMessage<ref name="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	
Close	CommandParameter: String<ref name="String" > System.String</ref>
Chain BroadcastMessageShow	
Show	- CommandParameter: IBroadcastMessage<ref name="IBroadcastMessage" > Genesyslab.Desktop.Modules.Core.Model.Broadcast.IBroadcast - Source: BroadcastMessageWindowSource<ref name="BroadcastMessageWindowSource" > Genesyslab.Desktop.Modules.Windows.Views.Broadcast.Broadcast
Chain ShowMessageToasterWindow Available since: 8.1.300.17	
ShowMessageToasterWindow	- CommandParameter: IToasterWindow<ref name="IToasterWindow" /> - BroadcastMessage: IBroadcastMessage<ref name="IBroadcastMessage" />

Additional GUI Commands

Additionally, you can use the following commands:

Additional Chains of Commands

Default Commands	Parameters
Chain ApplicationClose	
IsPossibleToClose	
CloseAllInteractions	
StopAllMonitoringCommand Available since: 8.1.300.17	
ChannelsLogOff Available since: 8.1.100.14	
NegotiatedLogout Available since: 8.1.100.14	
Close	
Chain CancelConnectionCommand Available since: 8.1.300.17	
Chain NotepadInsertTimeStamp	
InsertTimeStamp	- CommandParameter: IInteraction<ref name="Interaction" /> - TextBoxBase: TextBoxBase<ref name="TextBoxBase" >System.Windows.Controls.TextBoxBase</ref>
Chain PossibleToCloseAllInteractions	
IsPossibleToClose	
Chain TeamCommunicatorAddToFavorites Available since: 8.1.200.16	
AddObjectTCToFavorite	
Chain TeamCommunicatorModifyFavorite Available since: 8.1.300.17	

Default Commands	Parameters
ModifyFavorite	
Chain TeamCommunicatorOpenInteraction	
Open	ActionTarget: ActionTarget<ref name="ActionTarget" > Genesyslab.Desktop.Modules.Windows.Interactions.ActionTarget
Chain ContactIsModified	
IsModified	Caseld: String<ref name="String" />

References

<references />

Workbins



Purpose: To provide information about customizable commands.

Learn about Workbin Commands

These commands manage selections in workbins.

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 InteractionPossibleToMoveToWorkbin	
• IsWorkbinDestinationDefined	• WorkbinId: String<ref name="String" /> • WorkbinOptionName: String<ref name="String" /> • CommandParameter: InteractionOpenMedia<ref name="InteractionOpenMedia">Genesyslab.Desktop.Modules.OpenMedia.Model.Interaction.II
Chain WorkbinPreSelectedByName	
• ShowSelectedDetail	
Chain WorkbinPullInteraction	
• ShowSelectedDetail	
Chain WorkbinPushInteraction	

Default Commands	Parameters
ShowSelectedDetail	
Chain WorkbinSelected	
NewWorkbinSelected	- WorkbinName: string<ref name="string" >System.string</ref> - WorkbinNameDisplay: string<ref name="string" /> - WorkbinGroupName: string<ref name="string" /> - WorkbinGroupNameDisplay: string<ref name="string" />

References

<references />

Work Items



Purpose: To provide information about customizable commands.
Updated in: 8.1.300.17

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.

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	
• Accept	• CommandParameter: InteractionOpenMedia<ref name="InteractionOpenMedia" > Genesyslab.Desktop.Modules.OpenMedia.Model.Interactions.Ilr • Extensions: KeyValueCollection<ref name="KeyValueCollection" />
Chain InteractionWorkItemDecline	
• Decline	• CommandParameter:

Default Commands	Parameters
	InteractionOpenMedia<ref name="InteractionOpenMedia" /> - Reasons: KeyValueCollection<ref name="KeyValueCollection" > Genesyslab.Enterprise.Commons.Collections.KeyValueCollection - Extensions: KeyValueCollection<ref name="KeyValueCollection" />
Close Available since: 8.1.200.16	CommandParameter: InteractionOpenMedia<ref name="InteractionOpenMedia" />
Chain InteractionWorkitemMarkDoneById	
ConfirmWorkitem	- ActionToConfirm: String<ref name="String" > System.String</ref> - ConfirmParentWindow: Window<ref name="Window" > System.Window.Window</ref>
MarkDoneIdWorkitem	- CommandParameter: String<ref name="String" /> - WorkbinId: String<ref name="String" />
Chain InteractionWorkItemMoveToWorkbin	
IsWorkitemWorkbinDefined Deprecated since: 8.1.300.17	- WorkbinId: String<ref name="String" /> - WorkbinOptionName: String<ref name="String" /> - CommandParameter: InteractionOpenMedia<ref name="InteractionOpenMedia"> Genesyslab.Desktop.Modules.OpenMedia.Model.Interactions.Ir
IsWorkbinDestinationDefined	- WorkbinId: String<ref name="String" /> - WorkbinOptionName: String<ref name="String" /> - CommandParameter: InteractionOpenMedia<ref name="InteractionOpenMedia" />
SetAttachedDadInformation	CommandParameter: InteractionOpenMedia<ref name="

Default Commands	Parameters
	"InteractionOpenMedia" />
UpdateInteractionUCS	CommandParameter: Interaction<ref name="Interaction" > Genesyslab.Desktop.Modules.Core.Model.Interactions.Interaction
MoveToWorkbin	- CommandParameter: InteractionOpenMedia<ref name="InteractionOpenMedia" /> - WorkbinId: String<ref name="String" />
Chain InteractionWorkItemPossibleToMoveToWorkbin	
IsWorkitemWorkbinDefined Deprecated since: 8.1.300.17	- WorkbinId: String<ref name="String" /> - WorkbinOptionName: String<ref name="String" /> - CommandParameter: InteractionOpenMedia<ref name="InteractionOpenMedia" />
IsWorkbinDestinationDefined	- WorkbinId: String<ref name="String" /> - WorkbinOptionName: String<ref name="String" /> - CommandParameter: InteractionOpenMedia<ref name="InteractionOpenMedia" />
MoveToWorkbin Deprecated since: 8.1.300.17	- CommandParameter: InteractionOpenMedia<ref name="InteractionOpenMedia" /> - WorkbinId: String<ref name="String" />
Chain InteractionWorkitemPutBackToOriginalSource	
PutBackWorkitem	- CommandParameter: InteractionOpenMedia<ref name="InteractionOpenMedia" /> - Destination: String<ref name="String" /> - Reason: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" />

Default Commands	Parameters
	"KeyValueCollection" />
Close	CommandParameter: IInteractionOpenMedia<ref name="InteractionOpenMedia" />
Chain InteractionWorkItemSearchAndMoveToWorkbin	
SearchAndMoveToWorkbin	
Chain InteractionWorkitemWorkflowFromInteractionESDK	
ESDKWorkflow	- CommandParameter: IIOpenMediaInteraction<ref name="IIOpenMediaInteraction" > Genesyslab.Enterprise.Model.Interaction.IIOpenMediaInteraction - Workflow: String<ref name="String" /> - Reasons: KeyValueCollection<ref name="KeyValueCollection" /> - Extensions: KeyValueCollection<ref name="KeyValueCollection" /> - AttachedDataInformation: IDispositionCode<ref name="IDispositionCode" > Genesyslab.Desktop.Modules.Core.Model.Interactions.IDispositi - DispositionCode: KeyValueCollection<ref name="KeyValueCollection" />
Commands common to all interactions	
Chain InteractionSetAttachedDataWithDelay	
SetAttachedData	- CommandParameter: IInteraction<ref name="Interaction" /> - AttachedData: KeyValueCollection<ref name="KeyValueCollection" />
Chain BundleClose	
GetAttachedDataInformationUCS	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" > Genesyslab.Desktop.Modules.Core.Model.Interactions.IInteracti
UpdateNotePadForVoice	CommandParameter: IInteractionsBundle<ref

Default Commands	Parameters
Available since: 8.1.300.17	name="IInteractionsBundle" />
ResetInteractionChatConsultation	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />
IsPossibleToClose	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />
CompleteDispositionCodeOnBundle	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />
Close	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />
StopInteractionVoiceUCS	CommandParameter: IInteractionsBundle<ref name="IInteractionsBundle" />
GetOutboundPreviewRecord	CommandParameter: ICampaign<ref name="ICampaign" > Genesyslab.Enterprise.Model.Interaction.ICampaign</ref>

References

<references />

Customize Views and Regions



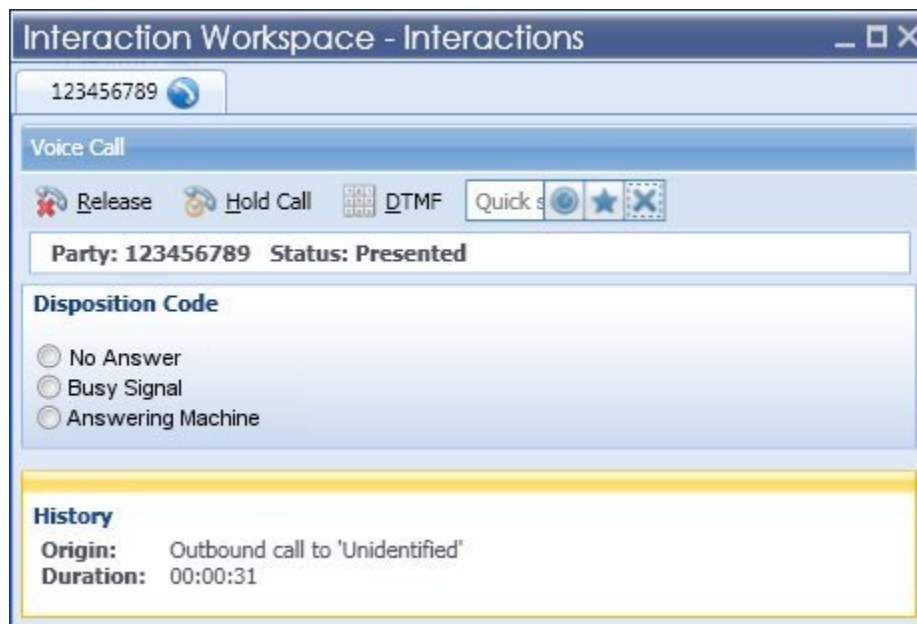
Purpose: To provide information about customizable views and their regions.

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 Interaction Workspace. 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#]

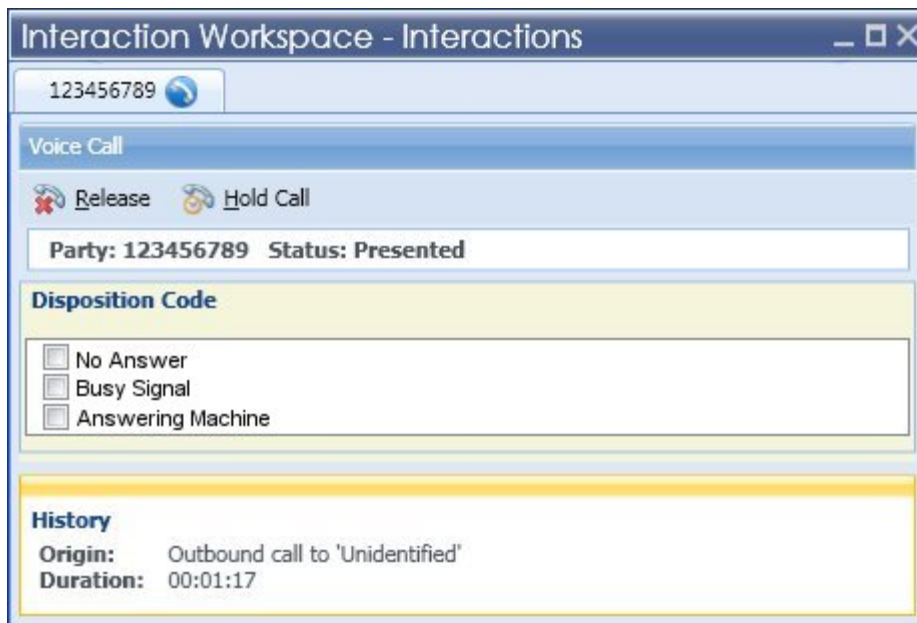
```
// File: DispositionCodeExView.cs
public partial class DispositionCodeExView : UserControl, IDispositionCodeView
{
    public DispositionCodeExView(IDispositionCodeViewModel dispositionCodeViewModel)
    {
        this.viewManager = viewManager;
        this.Model = dispositionCodePresentationModel;
        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 Interaction Workspace Voice Interactions after customization. In the customized view, the radio buttons for disposition codes are replaced with check boxes.



Disposition Code View After Customization

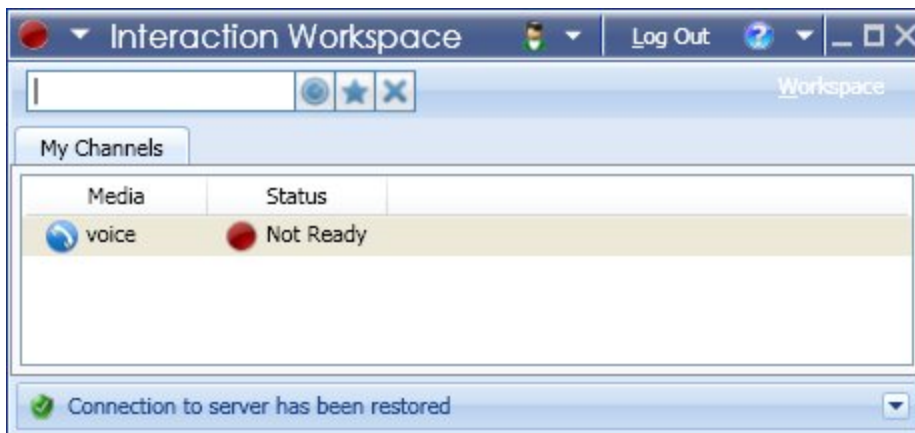
Although the application has a different appearance, it retains its previous behavior.

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 ToolbarWorkplaceRegion

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.

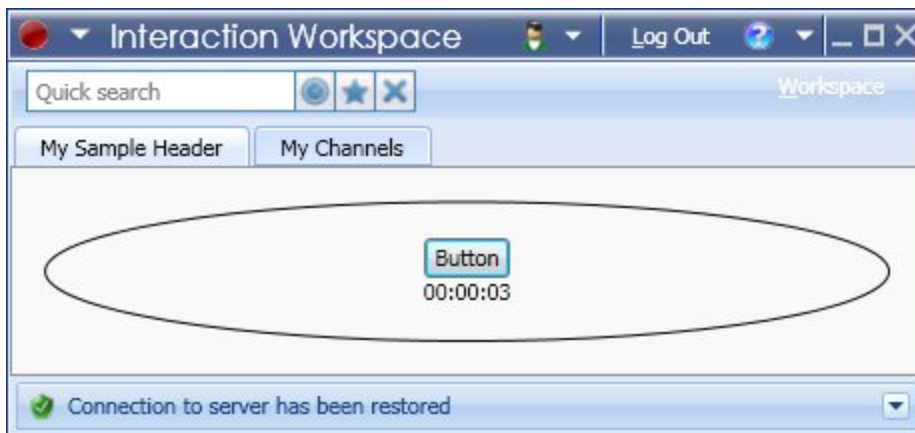
[C#]

```

// 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" });
        ...
    }
}

```

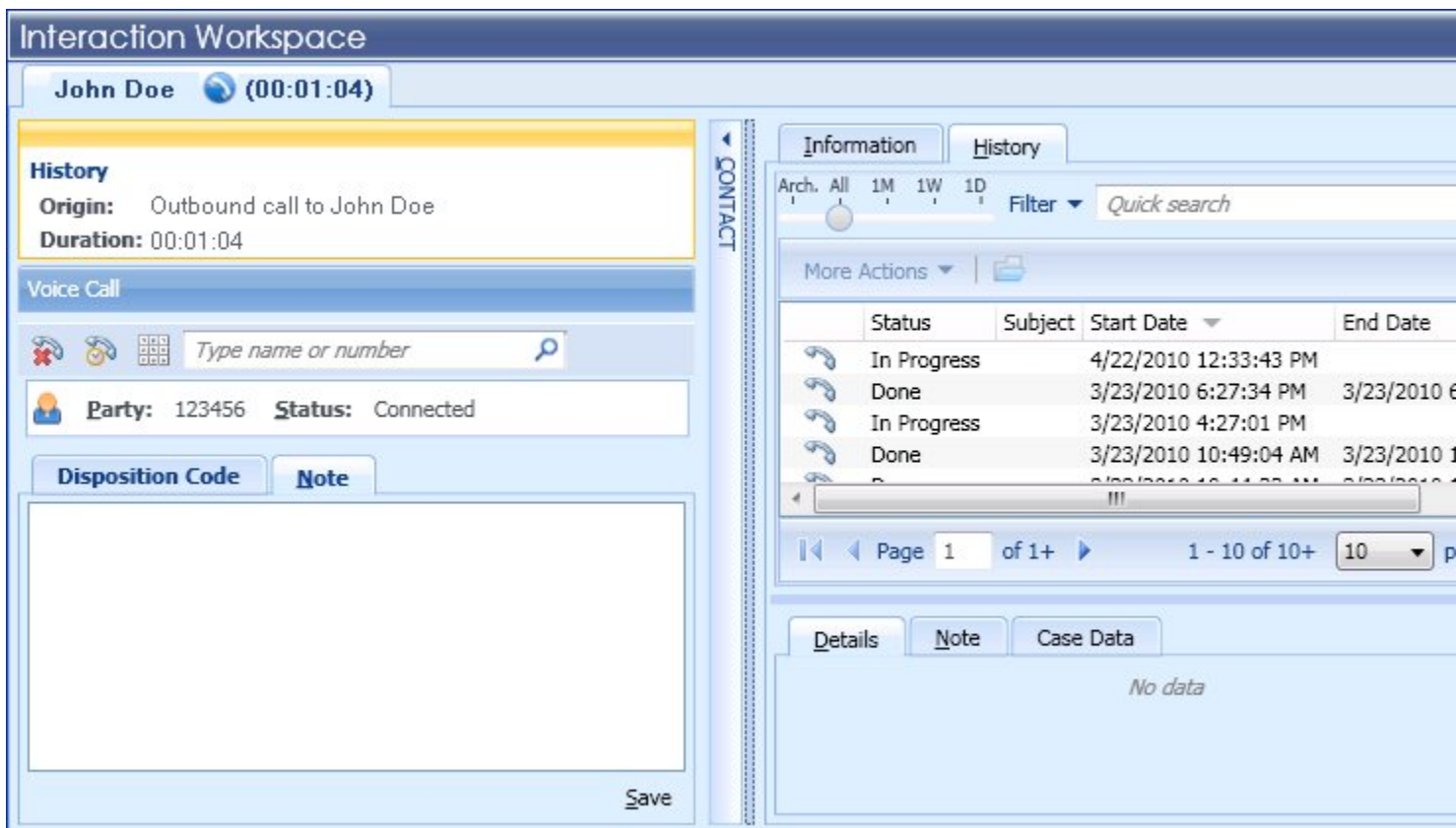
The figure below depicts the `MySampleView` after customization with the second tab (`My Sample Header`) included in the view. In the following example, the `ToolbarWorkplaceRegion` of the view is modified. For a complete list of views and regions, see [How to Customize Views and Their Regions Reference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

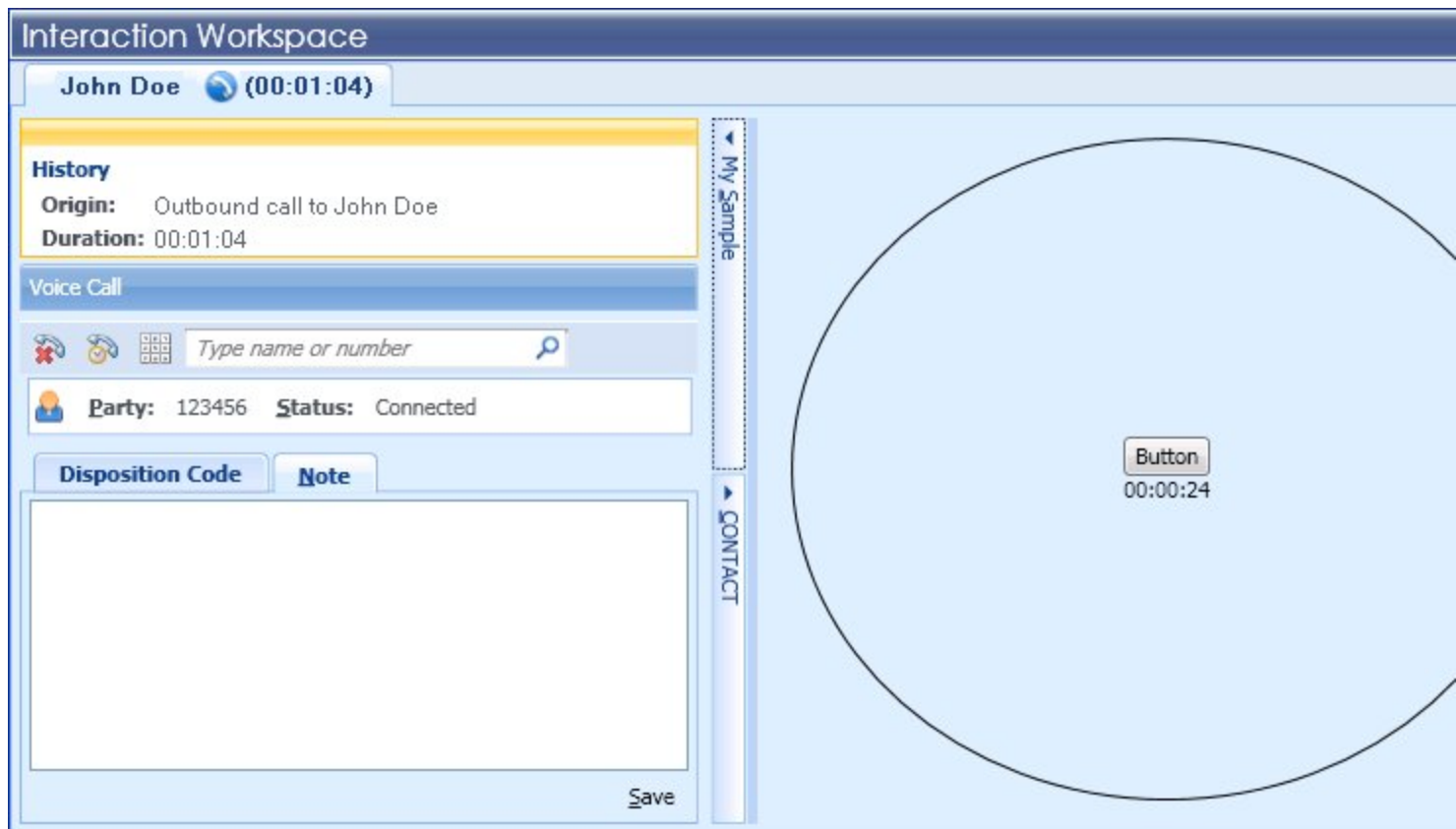
[C#]

```

// 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)
{
```

```
object>    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;
}
```

Read Next

 [Advanced Customization](#)

Best Practices for Views



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

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**--Interaction Workspace can be configured to emit a sound when an unsolicited event occurs.

Branding

To replace trademark logos, icon images and text, modify the `Rebranding.xml` file. The `Rebranding.xml` file is similar to a language dictionary and enables you customize the appearance of your application. For example, you can replace the embedded splashscreen resource, `pack://application:,,,/Genesyslab.Desktop.WPFCommon;component/Images/Splash.png`, with a local image such as `MySplash.png` or `MyImagesFolder/MySplash.png`. The `Rebranding.xml` file is shown in the following example:

[XML]

```
<Dictionary>
  <Value Id="Application.SplashScreen"
Source="pack://application:,,,/Genesyslab.Desktop.WPFCommon;component/Images/Splash.png"/>
  <Value Id="Window.AboutWindow.Logo" Source="MonLogo.png"/>
  <Value Id="Window.LoginWindow.Logo" Source="Branding/Logo2.png"/>
  <Value Id="Windows.AboutWindow.TextBlockInteractionWorkspace" Text="Custom Interaction
```

```
Workspace"/>
</Dictionary>
```

Note: You must include a path that is relative to the application when you replace the splashscreen.

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

Interaction Workspace 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>
  <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.

Loosely-coupled Application Library and Standard Controls

Interaction Workspace 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 SDK, 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 Interaction Workspace topic [to create your own view](#).

Reference for Windows



Purpose: To present the list of windows that your application can customize.

Windows and Views

Available windows and views	Related pages
- Window: ToolbarWindow - Multi-view: InteractionBarCasesRegion - Multi-view: MyWorkplaceContainerView - Multi-view: MainToolbarContainerView - Multi-view: MainToolbarWorkbinsContainerView	Toolbar Window
- Window: GadgetDialerWindow - Window: SystemTrayView - Window: GadgetMyWorkspaceWindow - View: GadgetMyChannelsWindow - View: GadgetMyHistoryWindow - View: GadgetWorkbinsContainerView - View: GadgetMyStatisticsWindow - View: GadgetCallCenterStatisticsWindow - View: GadgetMyMessages - View: GadgetMyCampaignsWindow - View: GadgetContactContainerView - Window: GadgetStatisticsWindow	Gadget Windows
InteractionsWindow	Interactions Window
- Window: ToasterMessageWindow - Window: ToasterWindow	Toaster Window
ContactSearchWindow	Windows Utilities, in Contacts and Favorites

Available windows and views	Related pages
- ContactMergeWindow - ContainerView	
- forward - CancelForwardView	Windows Utilities, in Forward and Cancel Forward
- IWMessageBoxView - AboutWindow - AuthenticationWindow - AuthentificationPasswordWindow - BroadcastMessageWindow - MediaInformationView - PrintPreviewWindow - EndWorkflowOpenMediaView - CalendarWindowView	Window Utilities, in Additional Accessories

References

<references />

Interactions Window



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

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

- **Window:** A window identifier (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:** 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>`

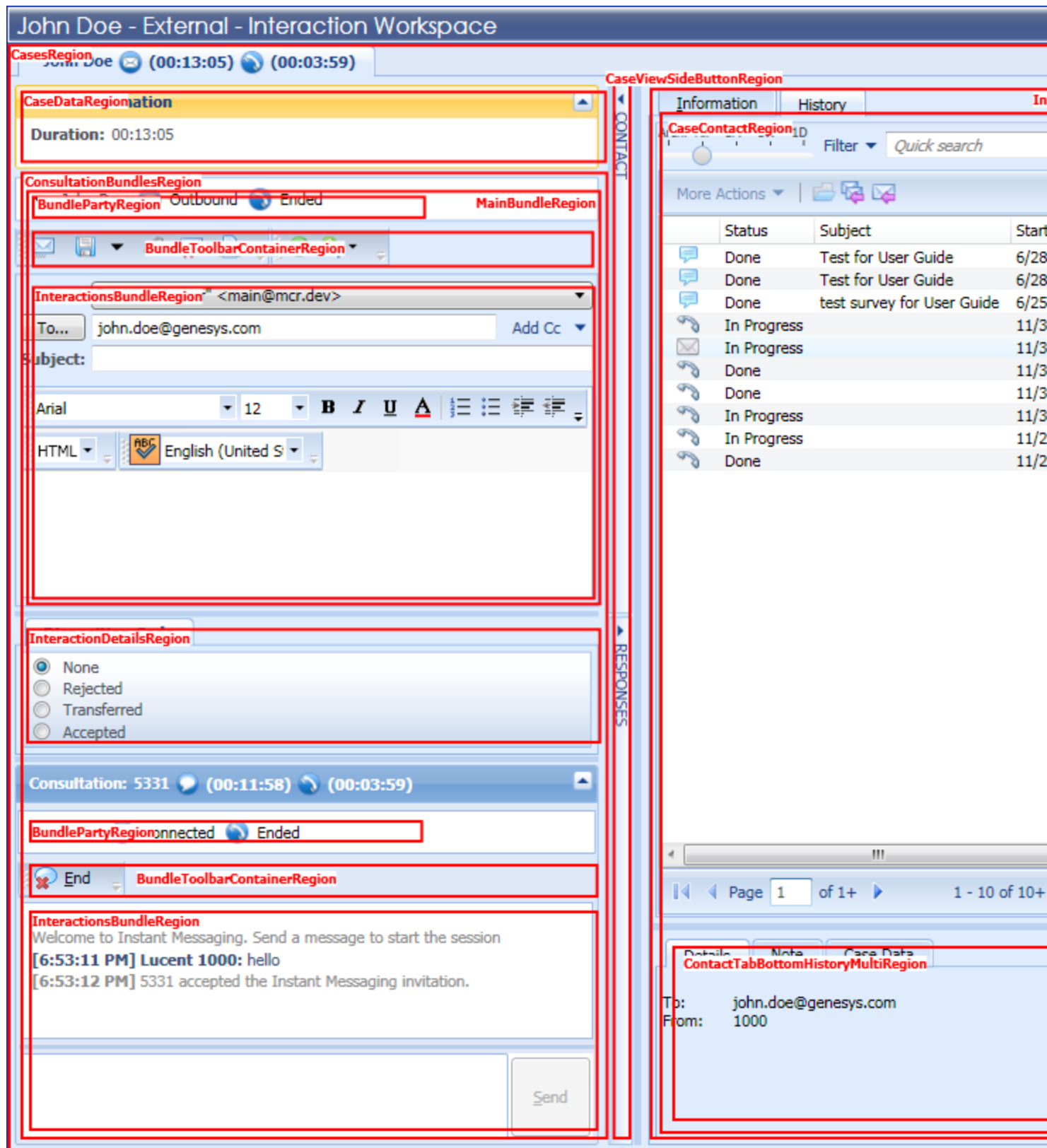
- **Region:** BundlePartyRegion
 - **View:** BundlePartyView (IBundlePartyView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>
 - **Region:** PartyRegion (Multi-view)
 - **View:** PartyView (IPartyView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>
 - **Region:** CustomBundlePartyRegion (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" />
 Available since: 8.1.300.17
- **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>
 Available since: 8.1.200.16
 - **View:** InteractionCallbackRescheduleToolbarView (IInteractionCallbackRescheduleToolbarView) <ref name="Callback">Genesyslab.Desktop.Modules.Callback</ref>
 Available since: 8.1.300.17
 - **View:** InteractionCallbackToolbarView (IInteractionCallbackToolbarView) <ref name="Callback">Genesyslab.Desktop.Modules.Callback</ref>
 Available since: 8.1.300.17
 - **View:** TeamLeadToolbarView (ITeamLeadToolbarView) <ref name="TeamLead">Genesyslab.Desktop.Modules.TeamLead</ref>
 Available since: 8.1.300.17
 - **View:** InteractionSmsToolbarView (IInteractionSmsToolbarView) <ref name="Sms">Genesyslab.Desktop.Modules.Sms</ref>
 Available since: 8.1.300.17

-
- "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>
Available since: 8.1.200.16
 - **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)
 - **View:** SIPMonitor (ISIPMonitor) <ref name="ActiveRecording">Genesyslab.Desktop.Modules.ActiveRecording</ref>
Available since: 8.1.200.16
 - **View:** SIPEndpointSoundView (ISIPEndpointSoundView) <ref name="SIPEndpointCommunication">Genesyslab.Desktop.Modules.SIPEndpointCommunication</ref>
 - **View:** InteractionWorkItemToolBarView (InteractionWorkItemToolBarView) <ref name="OpenMedia">Genesyslab.Desktop.Modules.OpenMedia</ref>
 - **Region:** CustomWorkItemToolBarRegion (Multi-view)
 - **Region:** CustomInfoOnInteractionsBundleRegion (Multi-view)
 - **View:** OutboundRecordView (IOutboundRecordView) <ref name="OpenMedia">Genesyslab.Desktop.Modules.OpenMedia</ref>
 - **Region:** UpdateOutboundFieldsRegion
 - **View:** UpdateOutboundFieldsView (IGenericDataView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>
 - **View:** InteractionCallbackInformationView (InteractionCallbackInformationView) <ref name="Callback">Genesyslab.Desktop.Modules.Callback</ref>
Available since: 8.1.300.17
 - **Region:** CallbackInformationRegion
-

- **View:** CallbackInformationView (IGenericDataView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>
- **Region:** InteractionsBundleRegion (Multi-view)
 - **View:** SmsView (ISmsView) <ref name="OpenMedia">Genesyslab.Desktop.Modules.OpenMedia</ref>
 - **View:** InboundEmailView (IInboundEmailView) <ref name="OpenMedia">Genesyslab.Desktop.Modules.OpenMedia</ref>
 - **Region:** EmailAttachmentsContentControlRegion
 - **View:** AttachmentView
EmailAttachmentsContentControlView (IAttachmentView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>
Updated in: 8.1.300.17
 - **View:** OutboundEmailView (IOutboundEmailView) <ref name="OpenMedia">Genesyslab.Desktop.Modules.OpenMedia</ref>
 - **Region:** OutboundEmailAttachmentsContentControlRegion
 - **View:**
AttachmentViewOutboundEmailAttachmentsContentControlView (IAttachmentView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>
Updated in: 8.1.300.17
 - **View:** ChatView (IChatView) <ref name="OpenMedia">Genesyslab.Desktop.Modules.OpenMedia</ref>
 - **View:** IMView (IIMCallView) <ref name="Voice">Genesyslab.Desktop.Modules.Voice</ref>
 - **View:** WorkItemView (IWorkItemView) <ref name="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>
Updated in: 8.1.300.17
- **View:** BundleView (IBundleView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>

-
- **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
Available since: 8.1.200.16
 - **View:** FormerContactDetailContentView
(IContactDetailContentView|FormerContactDetailContentView)
<ref name="Contacts">Genesyslab.Desktop.Modules.Contacts</ref>
Updated in: 8.1.300.17
 - **Region:** ControlListAttachmentRegion
 - **View:** AttachmentView
ControlListAttachmentView (IAttachmentView)
<ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>
Updated in: 8.1.300.17
 - **View:** OpenMediaContactDetailContentView
(IContactDetailContentView|OpenMediaContactDetailContentView)
<ref name="OpenMedia">Genesyslab.Desktop.Modules.OpenMedia</ref>
Updated in: 8.1.300.17
 - **Region:** ControlListAttachmentRegion

-
- **View:** AttachmentViewControlListAttachmentView (IAttachmentView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref> Updated in: 8.1.300.17
 - **View:** NotepadViewContactHistoryNotepadView (INotepadView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref> Updated in: 8.1.300.17
 - **View:** ContactHistoryCaseDataView (IStaticCaseDataView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>
 - **Region:** StaticCaseDataRegion Available since: 8.1.200.16
 - **View:** StaticCaseDataView (IGenericDataView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>
 - **Region:** StaticDispositionCodeRegion Available since: 8.1.200.16
 - **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> Updated in: 8.1.300.17



The complete view and the main inner views and regions of the `IInteractionWindow`.

References

<references />

Gadget Windows



Purpose: To present the list of gadget windows that your application can customize.
Updated in 8.1.300.17

Description

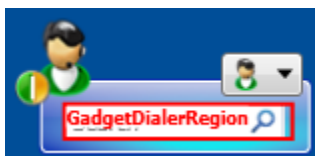
Gadgets display some specific views and regions also used in the main toolbar. This page presents the list of gadgets available.

Window: GadgetDialerWindow

Available since: 8.0.1

This gadget embeds a dialing toolbar.

- **Window:** GadgetDialerWindow (IGadgetWindow) <ref name="Windows" >Module: Genesyslab.Desktop.Modules.Windows</ref>
 - **Region:** GadgetCustomTrayRegion (Multi-view)
Available since: 8.1.200.16
 - **View:** MainToolbarMonitorView (IMainToolbarMonitorView) <ref name="ActiveRecording" >Module: Genesyslab.Desktop.Modules.ActiveRecording</ref>
 - **Region:** GadgetDialerRegion
 - **View:** GadgetDialer (ITeamCommunicatorView) <ref name="TeamCommunicator" >Module: Genesyslab.Desktop.Modules.TeamCommunicator</ref>
 - **Region:** GadgetSplitButtonRegion
Available since: 8.1.200.16
 - **View:** GadgetButtonView (IGadgetButtonView) <ref name="Windows" >Module: Genesyslab.Desktop.Modules.Windows</ref>

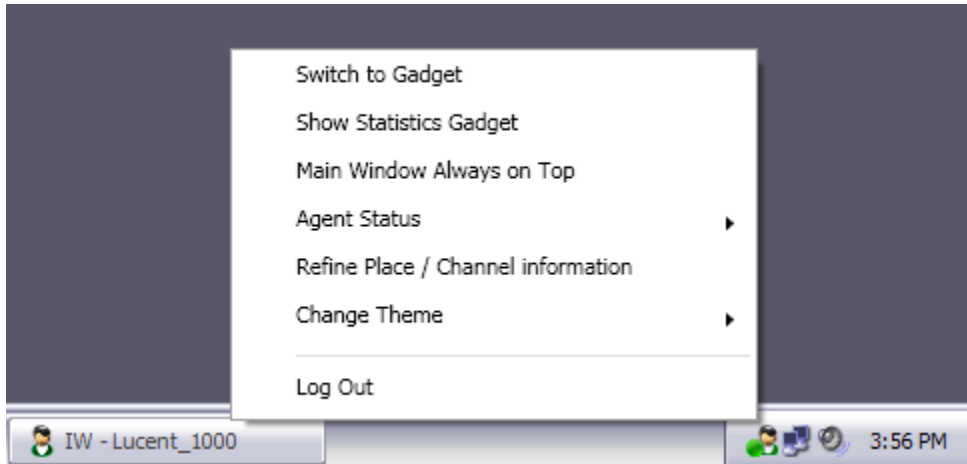


GadgetDialerRegion in the GadgetDialerWindow

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:

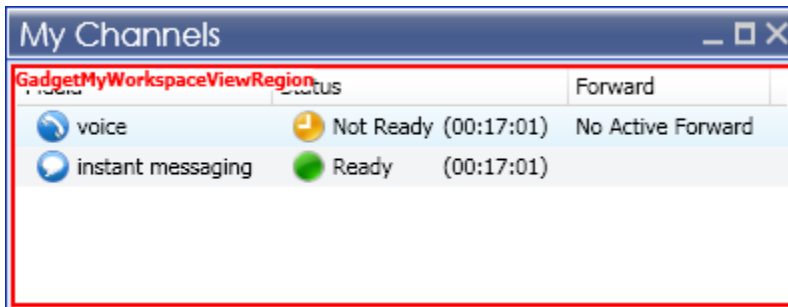
- **View:** [GadgetMyChannelsWindow](#)--The agent channels, with the place status.
- **View:** [GadgetMyHistoryWindow](#)--The agent's history.
- **View:** [GadgetWorkbinsContainerView](#) --The workbins explorer and menu.
- **View:** [MyStatisticsView](#)--The agent's statistics. (deprecated since 8.1.3)
- **View:** [GadgetMyStatisticsWindow](#)--The agent's statistics (since 8.1.3).
- **View:** [MyCallCenterStatisticsView](#)--The call center statistics. (deprecated since 8.1.3)
- **View:** [GadgetCallCenterStatisticsWindow](#)--The call center statistics(since 8.1.3).
- **View:** [GadgetMyMessages](#)--The agent's messages.
- **View:** [GadgetContactContainerView](#)--Displays the contact directory.(since: 8.1.200.16).
- **View:** [GadgetMyCampaignsWindow](#)--The campaign's messages (since: 8.1.200.16).

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

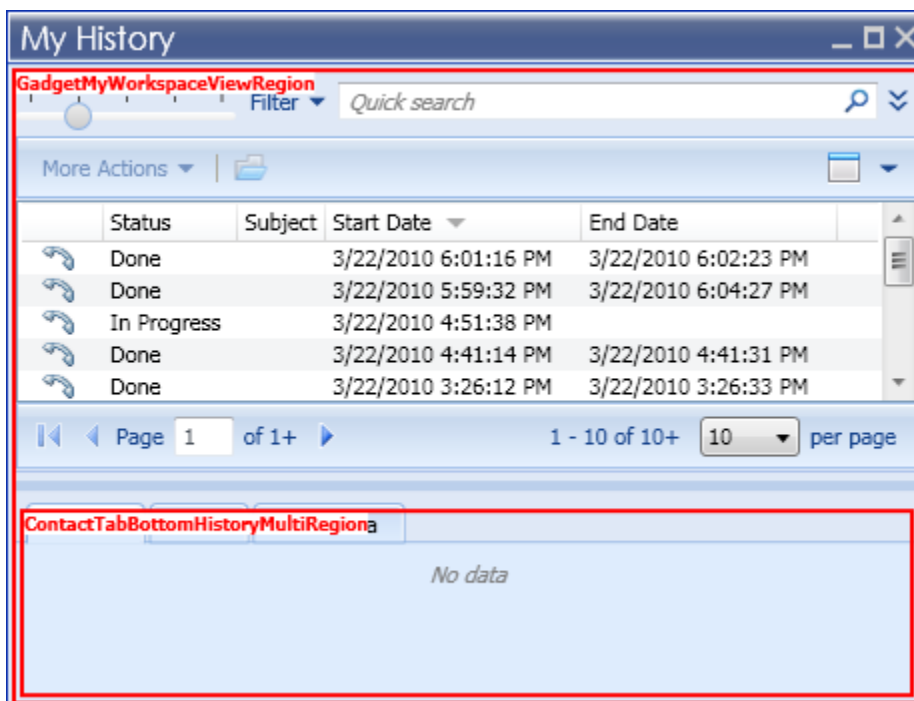
Available since: 8.0.1; updated in: 8.1.2

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
 - Available since: 8.1.200.16
 - **View:** FormerContactDetailContentView (IContactDetailContentView) <ref name="Contacts" > Module: Genesyslab.Desktop.Modules.Contacts</ref>
 - Available since: 8.1.200.16
 - **Region:** ControllistAttachmentRegion
 - **View:** AttachmentView (IAttachmentView) (Module:

Genesyslab.Desktop.Modules.Windows)

- **View:** OpenMediaContactDetailContentView (IContactDetailContentView) <ref name="OpenMedia" > Module: Genesyslab.Desktop.Modules.OpenMedia</ref>
Available since: 8.1.200.16
- **Region:** ControllListAttachmentRegion
 - **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
Available since: 8.1.200.16
 - **View:** StaticCaseDataView (IGenericDataView) <ref name="Windows" > Module: Genesyslab.Desktop.Modules.Windows</ref>
 - **Region:** StaticDispositionCodeRegion
Available since: 8.1.200.16
 - **View:** StaticDispositionCodeView (IGenericDataView) <ref name="Windows" > Module: Genesyslab.Desktop.Modules.Windows</ref>



GadgetMyHistoryWindow in the GadgetMyWorkspaceViewRegion

View: GadgetWorkbinsContainerView

Available since: 8.1.0; updated in: 8.1.3

Description: Displays the workbins views of the selected workbins:

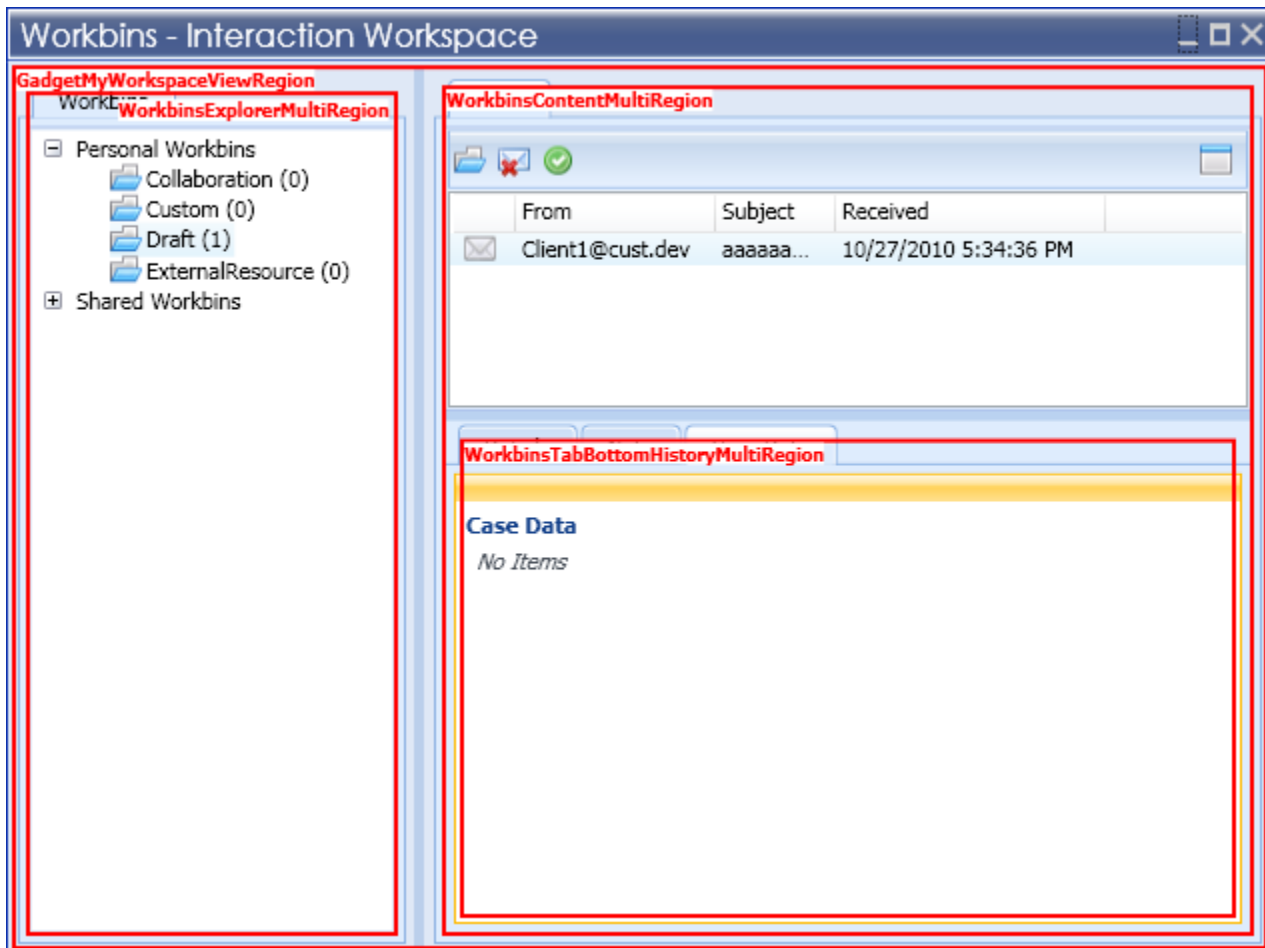
- **Window:** A window identifier (IGadgetMyWorkspaceWindowView) <ref name="Windows" />
- **Region:** GadgetMyWorkspaceViewRegion
 - **View:** GadgetWorkbinsContainerView (IMainToolbarWorkbinsContainerView) <ref name="OpenMedia" > Module: Genesyslab.Desktop.Modules.OpenMedia</ref>
 - **Region:** WorkbinsExplorerMultiRegion (Multi-view)
 - **View:** WorkbinsExplorerView (IWorkbinsExplorerView) <ref name="OpenMedia" />
 - **Region:** WorkbinsExplorerViewErrorRegion
 - **View:** WorkbinsExplorerViewErrorView (ILastAlertErrorView) <ref name="Windows" />
 - **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

Available since: 8.1.200.16

 - **View:** FormerContactDetailContentView (IContactDetailView|FormerContactDetailContentView) <ref name="Contacts" > Module: Genesyslab.Desktop.Modules.Contacts</ref> Updated in:8.1.300.17
 - **Region:** ControlListAttachmentRegion
 - **View:** AttachmentViewControlListAttachmentView (IAttachmentView) <ref name="Windows" > Module: Genesyslab.Desktop.Modules.Windows</ref> Updated in:8.1.300.17
 - **View:** OpenMediaContactDetailContentView (IContactDetailView|OpenMediaContactDetailContentView) <ref name="OpenMedia" > Module: Genesyslab.Desktop.Modules.OpenMedia</ref> Updated in:8.1.300.17
 - **Region:** ControlListAttachmentRegion
 - **View:** AttachmentViewControlListAttachmentView

(IAttachmentView) <ref name="Windows" > Module:
Genesyslab.Desktop.Modules.Windows</ref>
Updated in:8.1.300.17

- **View:** NotepadViewContactHistoryNotepadView (INotepadView) <ref name="Windows" />
Updated in:8.1.300.17
- **View:** ContactHistoryCaseDataView (IStaticCaseDataView) <ref name="Windows" />
 - **Region:** StaticCaseDataRegion
Available since: 8.1.200.16
 - **View:** StaticCaseDataView (IGenericDataView) <ref name="Windows" > Module:
Genesyslab.Desktop.Modules.Windows</ref>
 - **Region:** StaticDispositionCodeRegion
Available since: 8.1.200.16
 - **View:** StaticDispositionCodeView (IGenericDataView) <ref name="Windows" > Module:
Genesyslab.Desktop.Modules.Windows</ref>



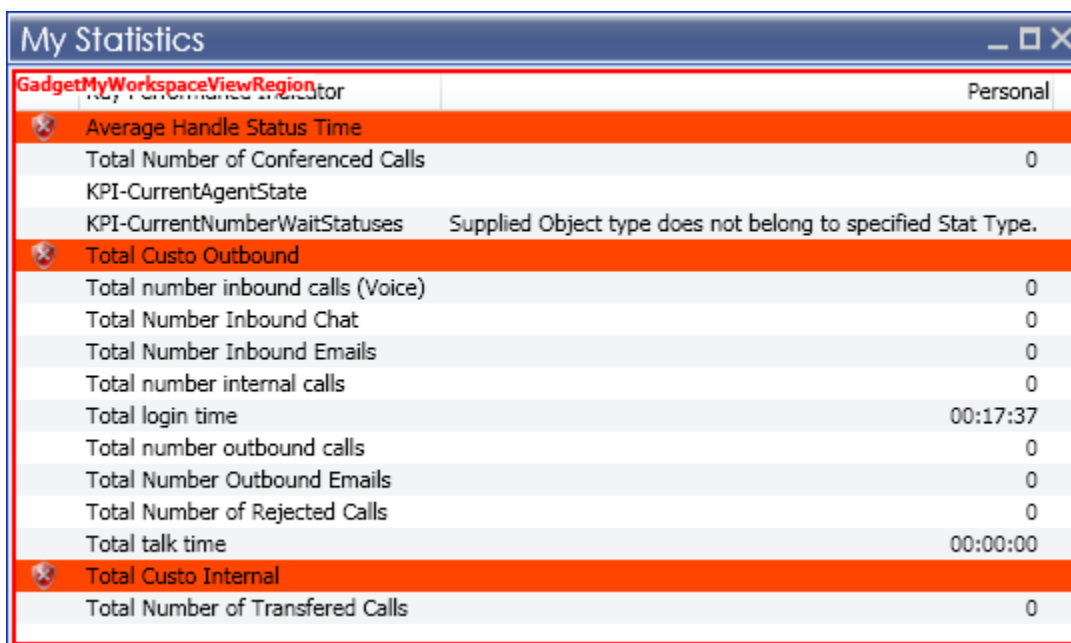
GadgetWorkbinsContainerView in GadgetMyWorkspaceViewRegion

View: MyStatisticsView

Available since: 8.0.1; **deprecated in:** 8.1.3; see [GadgetMyStatisticsWindow](#) instead

Description: Displays the agent's statistics. **Window:** A window identifier (IGadgetMyWorkspaceWindowView) <ref name="Windows" />

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



My Statistics		Personal
GadgetMyWorkspaceViewRegion		
Average Handle Status Time		
Total Number of Conferenced Calls		0
KPI-CurrentAgentState		
KPI-CurrentNumberWaitStatuses	Supplied Object type does not belong to specified Stat Type.	
Total Custo Outbound		
Total number inbound calls (Voice)		0
Total Number Inbound Chat		0
Total Number Inbound Emails		0
Total number internal calls		0
Total login time		00:17:37
Total number outbound calls		0
Total Number Outbound Emails		0
Total Number of Rejected Calls		0
Total talk time		00:00:00
Total Custo Internal		
Total Number of Transferred Calls		0

MyStatisticsView in GadgetMyWorkspaceViewRegion

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 Statistics		
GadgetMyWorkspaceViewRegion		Personal
Average Handle Status Time		
Total Number of Conferenced Calls		0
KPI-CurrentAgentState		
KPI-CurrentNumberWaitStatuses		Supplied Object type does not belong to specified Stat Type.
Total Custo Outbound		
Total number inbound calls (Voice)		0
Total Number Inbound Chat		0
Total Number Inbound Emails		0
Total number internal calls		0
Total login time		00:17:37
Total number outbound calls		0
Total Number Outbound Emails		0
Total Number of Rejected Calls		0
Total talk time		00:00:00
Total Custo Internal		
Total Number of Transferred Calls		0

GadgetMyStatisticsWindow in GadgetMyWorkspaceViewRegion

View: MyCallCenterStatisticsView

Available since: 8.0.1; deprecated in: 8.1.3

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

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

Call Center Statistics		
GadgetMyWorkspaceViewRegion	Description	Value
80001@LucentG3	Average abandoned call time	00:00:00
80001@LucentG3	Abandoned Calls Percentage	0%
122@LucentG3	Abandoned Calls Percentage - RP	0%
80001@LucentG3	Average distribution call time	00:00:00
80001@LucentG3	Max Waiting Time	00:00:00
80001@LucentG3	Current number of interactions waiting in the queue	0
80001@LucentG3	Distrib Calls Percentage	0%
122@LucentG3	Distrib Calls Percentage - RP	0%
80001@LucentG3	Estimated time to handle interaction in the queue	00:00:00
80001@LucentG3	Total number of abandoned calls	0
122@LucentG3	Total number of abandoned calls - RP	0
80001@LucentG3	Total number of distributed calls	0
122@LucentG3	Total number of distributed calls - RP	0

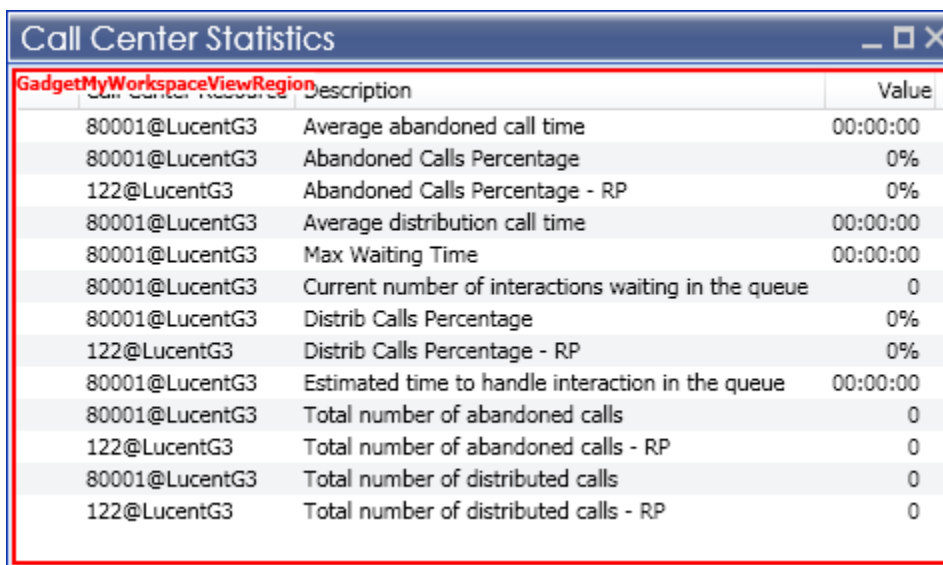
MyCallCenterStatisticsView in GadgetMyWorkspaceViewRegion

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" />



Call Center Resource	Description	Value
80001@LucentG3	Average abandoned call time	00:00:00
80001@LucentG3	Abandoned Calls Percentage	0%
122@LucentG3	Abandoned Calls Percentage - RP	0%
80001@LucentG3	Average distribution call time	00:00:00
80001@LucentG3	Max Waiting Time	00:00:00
80001@LucentG3	Current number of interactions waiting in the queue	0
80001@LucentG3	Distrib Calls Percentage	0%
122@LucentG3	Distrib Calls Percentage - RP	0%
80001@LucentG3	Estimated time to handle interaction in the queue	00:00:00
80001@LucentG3	Total number of abandoned calls	0
122@LucentG3	Total number of abandoned calls - RP	0
80001@LucentG3	Total number of distributed calls	0
122@LucentG3	Total number of distributed calls - RP	0

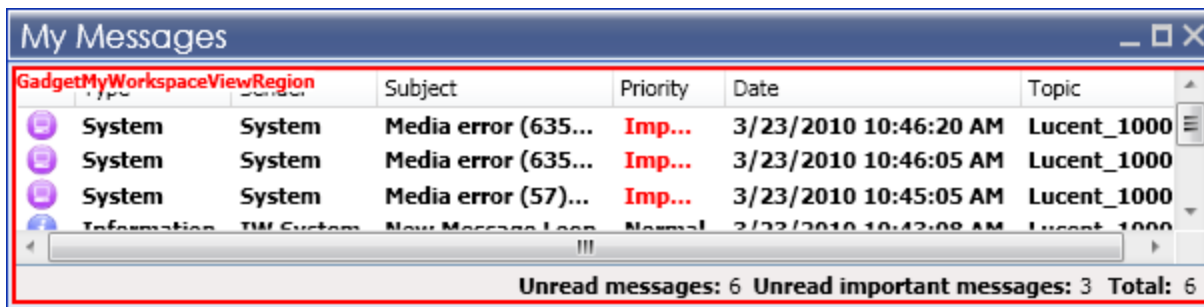
GadgetCallCenterStatisticsWindow in GadgetMyWorkspaceViewRegion

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" />

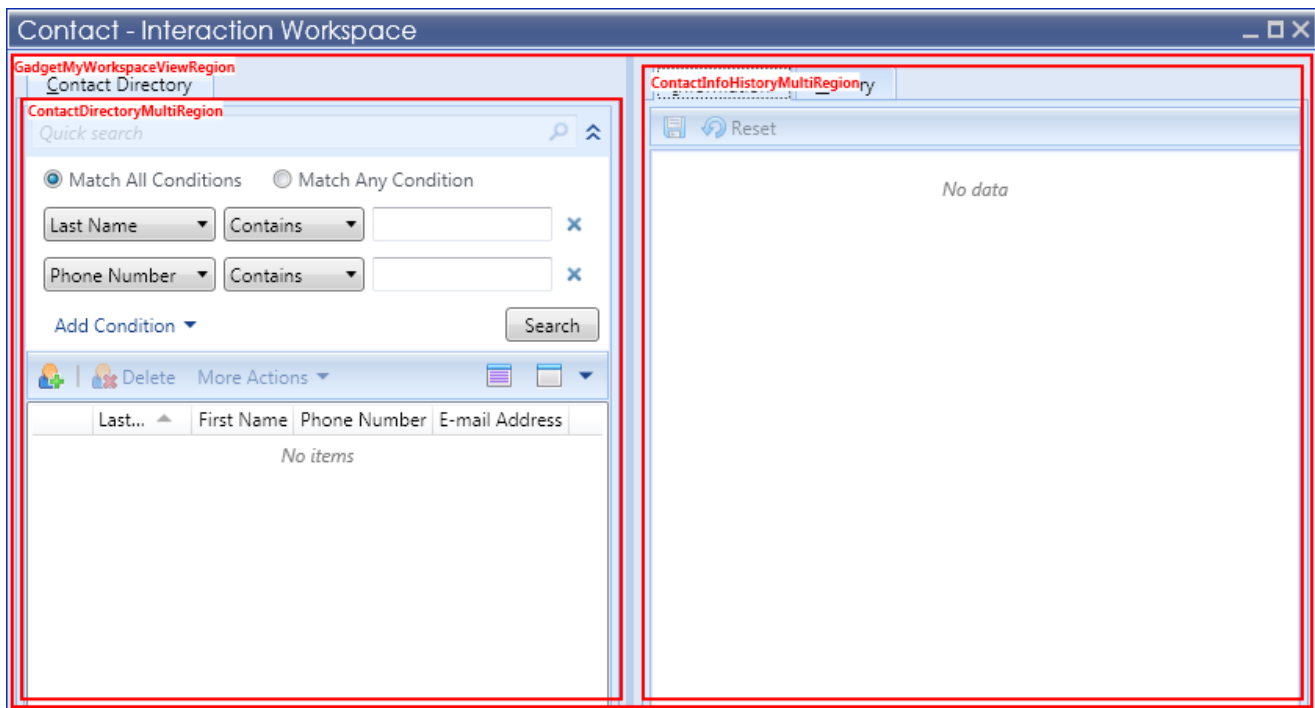
View: GadgetContactContainerView

Available since: 8.1.200.16; updated in 8.1.300.17

Description: Displays the contact directory.

- **Window:** A window identifier (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:** 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 (IContactDetailViewIFormerContactDetailContentView) <ref name="Contacts">Genesyslab.Desktop.Modules.Contacts</ref>
Updated in: 8.1.300.17
 - **Region:** ControlListAttachmentRegion
 - **View:** AttachmentView ControlListAttachmentView (IAttachmentView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>
Updated in: 8.1.300.17
 - **View:** OpenMediaContactDetailContentView (IContactDetailViewIOpenMediaContactDetailContentView) <ref name="OpenMedia">Genesyslab.Desktop.Modules.OpenMedia</ref>
Updated in: 8.1.300.17
 - **Region:** ControlListAttachmentRegion
 - **View:** AttachmentView ControlListAttachmentView (IAttachmentView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>
Updated in: 8.1.300.17
 - **View:** NotepadViewContactHistoryNotepadView (INotepadView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>
Updated in: 8.1.300.17
 - **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>
-

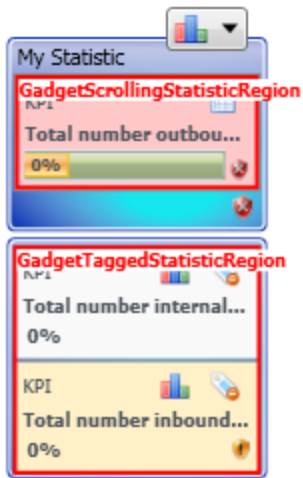


Window: GadgetStatisticsWindow

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

<references />

Toaster Window



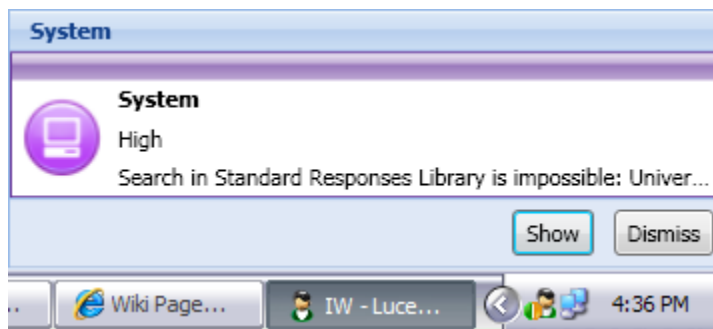
Purpose: To present for each window, the list of default views and regions that your application can customize.

Window: ToasterMessageWindow

Available since: 8.1.0

Description: The Toaster Message Window displays the business-type or system-type messages that your application receives.

- **Window:** A window identifier (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

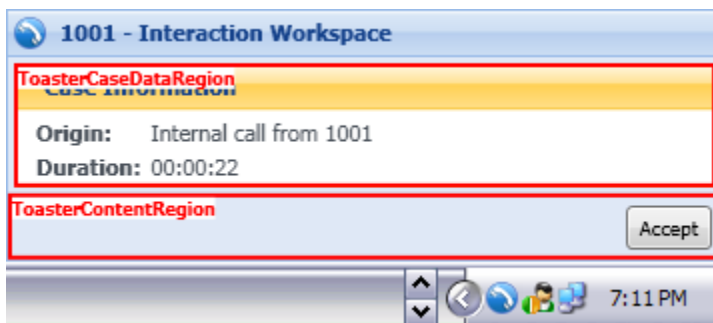
Window: ToasterWindow

Available since: 8.1.0

Description: The toaster window displays light information about the current case in progress.

- **Window:** A window identifier (IToasterWindow) `<ref name="Windows" />`
 - **Region:** ToasterCaseDataRegion (Multi-view)

- **View:** CaseDataView (ICaseDataView) <ref name="Windows" />
 - **Region:** UpdateCaseDataRegion
Available since: 8.1.200.16
 - **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

<references />

Toolbar Window



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

The ToolbarWindow (IToolbarWindow) <ref name="Windows" >Module: Genesyslab.Desktop.Modules.Windows</ref> is the top-level window which contains predefined buttons in the ToolbarWorksheetButtonRegion associated with specific working views.



The ToolbarWindow; no working view is deployed.

This toolbar contains views and regions which are activated depending on the user's input by clicking expandable button for instance. The components available in this toolbar are the following views and regions:

- **Window:** ToolbarWindow (IToolbarWindow) <ref name="Windows" />
 - **Region:** MainSplitButtonRegion (Multi-view)
Available since: 8.1.200.16
 - **View:** MainButtonView (IMainButtonView) <ref name="Windows" >Module: Genesyslab.Desktop.Modules.Windows</ref>
 - **View:** MainToolbarMonitorView (IMainToolbarMonitorView) <ref name="ActiveRecording" >Module: Genesyslab.Desktop.Modules.ActiveRecording</ref>
Available since: 8.1.300.17
 - **Region:** HelpSplitButtonRegion
Available since: 8.1.200.16
 - **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:** ToolbarWorksheetButtonRegion (Multi-view)

- **View:** MyWorkplaceButtonView (IMyWorkplaceButtonView) <ref name="Windows" />
 - **Region:** WorkspaceMenuRegion
 - **View:** PlaceStatusMenu (IPlaceStatusMenuView)
 - **View:** MyCampaignsMenu (IMyCampaignsMenuView) <ref name="Outbound" >Module: Genesyslab.Desktop.Modules.Outbound</ref>
Available since: 8.1.200.16
 - **View:** MyHistoryMenu (IHistoryMenuView) <ref name="Contacts" />
 - **View:** MyStatisticsMenu (IMyStatisticsMenuView) <ref name="Statistics" />
 - **View:** CallCenterStatisticsMenu (ICallCenterStatisticsMenuView) <ref name="Statistics" >Module: Genesyslab.Desktop.Modules.Statistics</ref>
- **View:** MainToolbarWorkbinsContainerButtonView (IMainToolbarWorkbinsContainerButtonView) <ref name="OpenMedia">Module: Genesyslab.Desktop.Modules.OpenMedia</ref>
- **View:** MainToolbarContainerButtonView (IMainToolbarContainerButtonView) <ref name="Contacts" >Module: Genesyslab.Desktop.Modules.Contacts</ref>
- **View:** MainToolbarContainerAlertMessageButtonView (IAlertMessageButtonView) <ref name="Windows" />

Available since: 8.1.100.14

- **Region:** ToolbarInteractionBarRegion

Available since: 8.1.100.14

- **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
 - **View:** AlertMessageView (IAlertMessageView) <ref name="Windows" />
 - **Region:** MyMessageRegion
 - **View:** MyMessagesView (IMyBroadcastMessageView) <ref name="Windows" />
- **Region:** ToolbarWorksheetRegion
 - **View:** MyWorkplaceContainerView (IMyWorkplaceContainerView) <ref name="Windows" />
 - **View:** MainToolbarContainerView (IMainToolbarContainerView) <ref name="Contacts" />
 - **View:** MainToolbarWorkbinsContainerView (IMainToolbarWorkbinsContainerView) <ref name="OpenMedia" />

The following subsections detail the multi-views which are part of the components listed above.

Multi-view: InteractionBarCasesRegion

Available since: 8.1.1. **Updated in:** 8.1.2, 8.1.3

The multi-view InteractionBarCasesRegion is part of the InteractionBar region and includes the following regions and views:

- **Region:** InteractionBarCasesRegion (Multi-view)
 - **View:** InteractionBarCase (IInteractionBarCaseView) <ref name="InteractionBar"/>
 - **Region:** InteractionBarMainBundleRegion
 - **View:** InteractionBarMainBundle (IInteractionBarMainBundleView) <ref name="InteractionBar">Module: Genesyslab.Desktop.Modules.InteractionBar</ref>
 - **Region:** InteractionBarInteractionsRegion (Multi-view)
 - **View:** InteractionBarInteraction (IInteractionBarInteractionView) <ref name="InteractionBar"/>
 - **Region:** BundleToolbarContainerRegion (Multi-view)
 - **View:** InteractionCallbackToolBarView (IInteractionCallbackToolBarView) <ref name="Callback">Module: Genesyslab.Desktop.Modules.Callback</ref>
Available since: 8.1.300.17
 - **View:** TeamLeadToolBarView (ITeamLeadToolBarView) <ref name="TeamLead">Module: Genesyslab.Desktop.Modules.TeamLead</ref>
Available since: 8.1.300.17
 - **View:** InteractionPreviewToolBarView (IInteractionPreviewToolBarView) <ref name="Outbound">Module: Genesyslab.Desktop.Modules.Outbound</ref>
 - **View:** InteractionSmsToolBarView (IInteractionSmsToolBarView) <ref name="OpenMedia">Module: Genesyslab.Desktop.Modules.OpenMedia</ref>
 - **View:** InteractionInboundEmailToolBarView (IInteractionInboundEmailToolBarView) <ref name="OpenMedia" />
 - **View:** InteractionOutboundEmailToolBarView (IInteractionOutboundEmailToolBarView) <ref name="OpenMedia" />
 - **View:** BundleToolBarView (IBundleToolBarView) (Module: Genesyslab.Desktop.Modules.Windows)
Available since: 8.1.300.17
 - **Region:** BundleCustomButtonRegion (Multi-view)
 - **View:** DonePullPreviewToolBarView (IDonePullPreviewToolBarView) <ref name="Outbound">Module: Genesyslab.Desktop.Modules.Outbound</ref>
 - **View:** RescheduleToolBarView (IRescheduleToolBarView) <ref name="Outbound">Module: Genesyslab.Desktop.Modules.Outbound</ref>
 - **View:** IxnMonitor (IxnMonitor) <ref name="

-
- "ActiveRecording">Module:
Genesyslab.Desktop.Modules.ActiveRecording</ref>
 - **View:** InteractionCallbackRescheduleToolBarView (IInteractionCallbackRescheduleToolBarView) <ref name="Callback">Module:
Genesyslab.Desktop.Modules.Callback</ref>
 - **View:** InteractionChatToolBarView (IInteractionChatToolBarView) <ref name="OpenMedia" />
 - **View:** InteractionChatConsultToolBarView (IInteractionChatConsultToolBarView) <ref name="OpenMedia">Module:
Genesyslab.Desktop.Modules.OpenMedia</ref>
Available since: 8.1.200.16
 - **View:** InteractionIMToolBarView (IInteractionIMToolBarView) <ref name="Voice">Module: Genesyslab.Desktop.Modules.Voice</ref>
 - **View:** InteractionPreviewToolBarView (IInteractionPreviewToolBarView) <ref name="Outbound">Module:
Genesyslab.Desktop.Modules.Outbound</ref>
 - **View:** InteractionVoiceToolBarView (IInteractionVoiceToolBarView) <ref name="Voice" />
 - **Region:** InteractionVoiceCustomButtonRegion (Multi-view)
 - **View:** SIPMonitor (ISIPMonitor) <ref name="ActiveRecording">Module:
Genesyslab.Desktop.Modules.ActiveRecording</ref>
Available since: 8.1.200.16
 - **View:** SIPEndpointSoundView (ISIPEndpointSoundView) <ref name="SIPEndpointCommunication">Module:
Genesyslab.Desktop.Modules.SIPEndpointCommunication</ref>
 - **View:** InteractionWorkItemToolBarView (IInteractionWorkItemToolBarView) <ref name="OpenMedia" />
 - **Region:** CustomWorkItemToolBarRegion (Multi-view)
 - **Region:** InteractionBarBundleToolBarRegion (Multi-view)
 - **View:** TeamLeadToolBarView (ITeamLeadToolBarView) <ref name="TeamLead">Module: Genesyslab.Desktop.Modules.TeamLead</ref>
Updated in:8.1.300.17
 - **View:** BundleToolBarView (IBundleToolBarView) <ref name="Windows" />
 - **Region:** BundleCustomButtonRegion (Multi-view)
 - **View:** DonePullPreviewToolBarView (IDonePullPreviewToolBarView) <ref name="Outbound" />
 - **View:** RescheduleToolBarView (IRescheduleToolBarView) <ref name="Outbound" />
 - **View:** IxnMonitor (IIxnMonitor) <ref name="ActiveRecording">Module:
Genesyslab.Desktop.Modules.ActiveRecording</ref>
-

Available since: 8.1.200.16

- **View:** InteractionCallbackRescheduleToolBarView (InteractionCallbackRescheduleToolBarView) (Module: Genesyslab.Desktop.Modules.Callback)
Available since: 8.1.300.17
- **Region:** InteractionBarConsultationBundlesRegion (Multi-view)
 - **View:** InteractionBarConsultationBundle (InteractionBarConsultationBundleView) <ref name="Interactionbar" >Module: Genesyslab.Desktop.Modules.InteractionBar</ref>
Available since: 8.1.200.16
 - **Region:** InteractionBarInteractionsRegion (Multi-view)
 - **Region:** InteractionBarBundleToolBarRegion (Multi-view)

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" />
Available since: 8.0.1
 - **View:** myCampaignsListView (IMyCampaignsView) <ref name="Outbound" >Module: Genesyslab.Desktop.Modules.Outbound</ref>
- Available since: 8.1.100.14
 - **Region:** CampaignsErrorRegion
 - **View:** campaignsErrorView (ILastAlertErrorView) <ref name="Windows" />
 - **View:** MyContactHistory (IMyContactHistoryView) <ref name="Contacts" />
Available since: 8.1.0
 - **Region:** ContactHistoryErrorRegion
 - **View:** ContactHistoryErrorView (ILastAlertErrorView) <ref name="Windows" />
 - **Region:** ContactTabBottomHistoryMultiRegion (Multi-view)
 - **View:** ContactDetailView (IContactDetailView) <ref name="Contacts" />
 - **Region:** ContactDetailRegion
Available since: 8.1.200.16
 - **View:** OpenMediaContactDetailContentView (IContactDetailView)(IOpenMediaContactDetailContentView) <ref name="Contacts">Module: Genesyslab.Desktop.Modules.Contacts</ref>
Updated in: 8.1.300.17

- **Region:** ControlListAttachmentRegion
 - **View:** AttachmentView ControlListAttachmentView (IAttachmentView) <ref name="Windows" />
- **View:** FormerContactDetailContentView (IContactDetailContentView|FormerContactDetailContentView) <ref name="Contacts">Module: Genesyslab.Desktop.Modules.Contacts</ref>
Available since: 8.1.200.16, updated in: 8.1.300.17
 - **Region:** ControlListAttachmentRegion
 - **View:** AttachmentView ControlListAttachmentView (IAttachmentView) <ref name="Windows" />
Updated in: 8.1.300.17
- **View:** ~~NotepadView~~ ContactHistoryNotepadView
Updated in: 8.1.300.17
(INotepadView) <ref name="Windows" />
- **View:** ContactHistoryCaseDataView (IStaticCaseDataView) <ref name="Windows" />
 - **Region:** StaticCaseDataRegion
Available since: 8.1.200.16
 - **View:** StaticCaseDataView (IGenericDataView) <ref name="Windows">Module: Genesyslab.Desktop.Modules.Windows</ref>
 - **Region:** StaticDispositionCodeRegion
Available since: 8.1.200.16
 - **View:** StaticDispositionCodeView (IGenericDataView) <ref name="Windows">Module: Genesyslab.Desktop.Modules.Windows</ref>
- **View:** MyStatisticsView (IMyStatisticsView) <ref name="Statistics" />
- **View:** MyCallCenterStatisticsView (IMyCallCenterStatisticsView) <ref name="Statistics" />

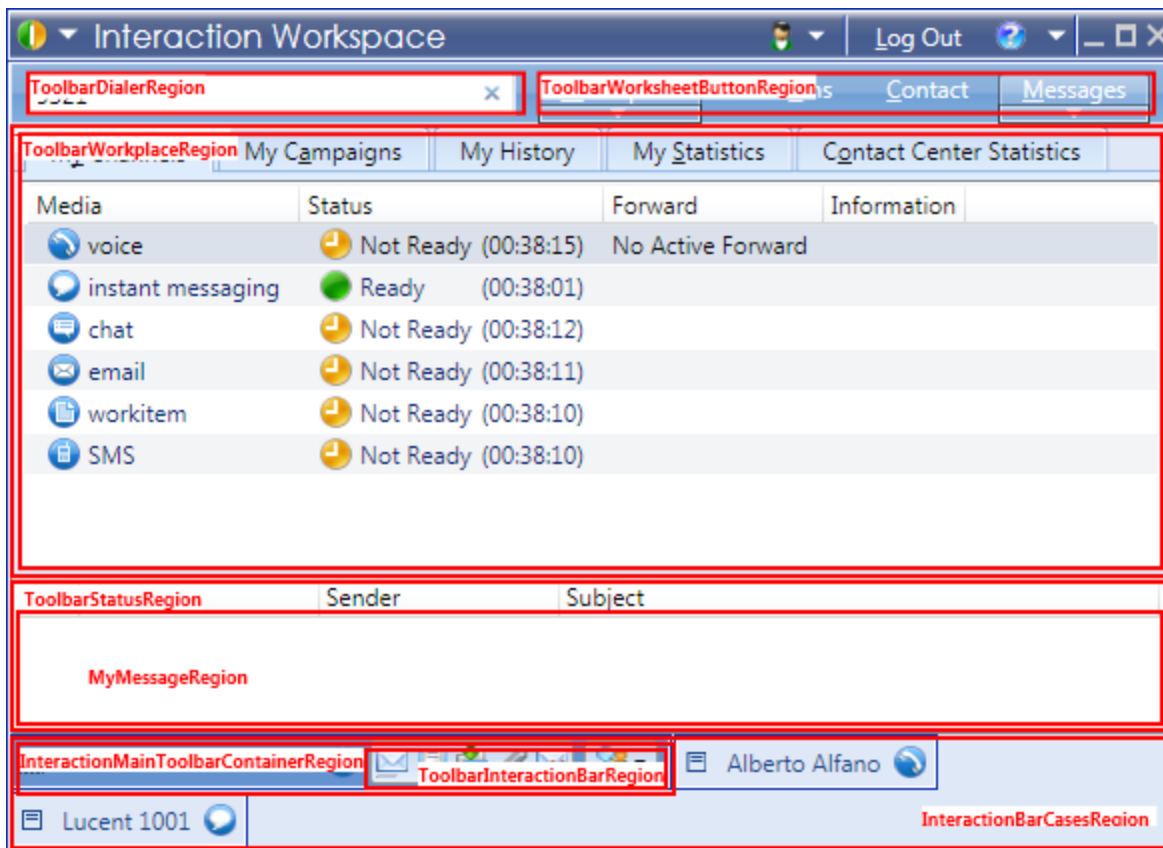
See the **Gadget Windows** for screenshots of **MyStatisticsView** and **MyCallCenterStatisticsView**.

Example: MyPlaceStatusView

Updated in: 8.1.100.14

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 *MyChannel* tab is selected and makes visible the MyPlaceStatusView.



The selection of 'MyChannels' displays the 'PlaceStatus' View (8.1.1).

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

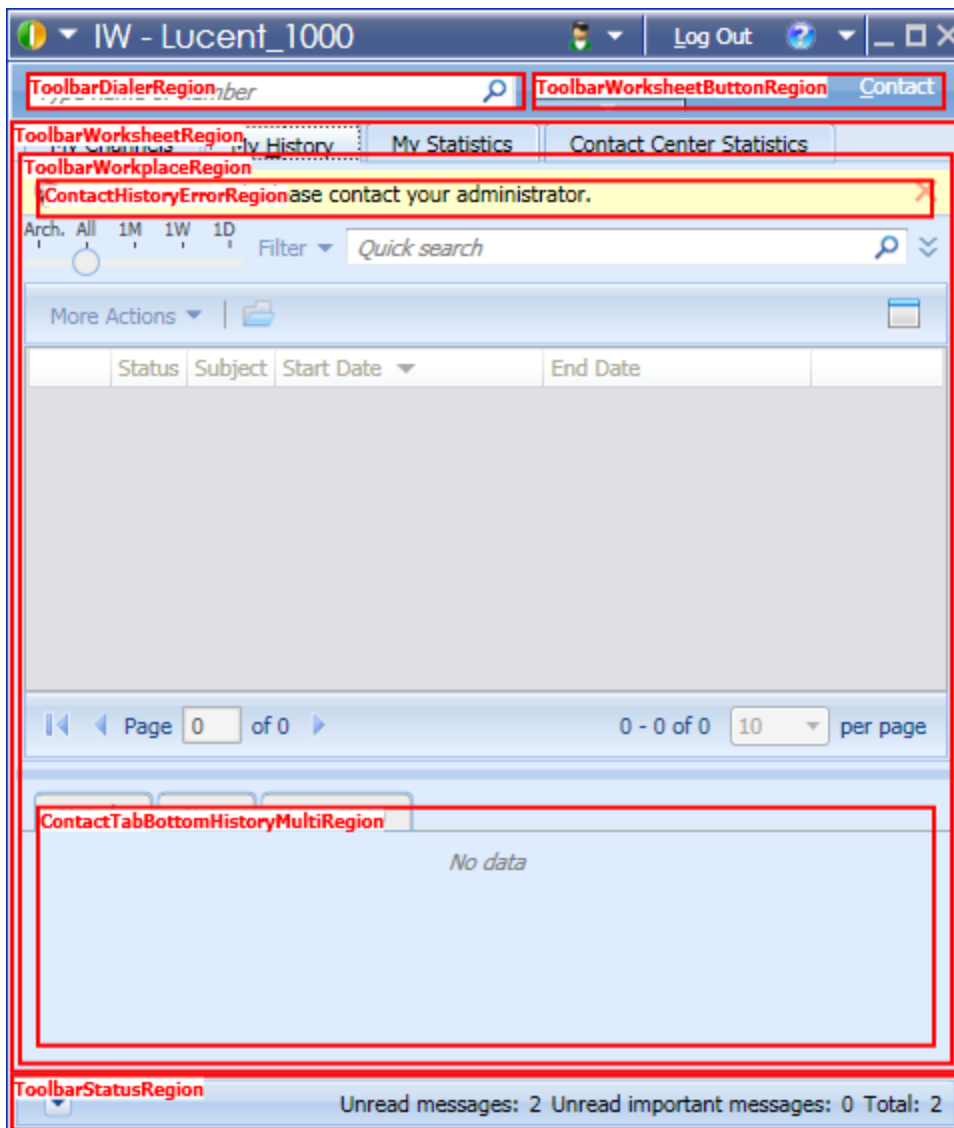
- **Window:** ToolbarWindow (IToolbarWindow) <ref name="Windows" />
 - **Region:** MainSplitButtonRegion (Multi-view)
 - **View:** MainButtonView (IMainButtonView) <ref name="Windows" >Module: Genesyslab.Desktop.Modules.Windows</ref>
 - **Region:** HelpSplitButtonRegion
 - **View:** ButtonHelpView (IButtonHelpView) <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:** PlaceStatusMenu (IPlaceStatusMenuView)
 - **Region:** ToolbarStatusRegion

-
- **View:** AlertMessageView (IAlertMessageView) <ref name="Windows" />
 - **Region:** MyMessageRegion
 - **View:** MyMessagesView (IMyBroadcastMessageView) <ref name="Windows" />
 - **Region:** ToolbarWorksheetRegion
 - **View:** MyWorkplaceContainerView (IMyWorkplaceContainerView) <ref name="Windows" />
 - **Region:** ToolbarWorkplaceRegion (Multi-view)
 - **View: MyPlaceStatusView** (IMyPlaceStatusView) <ref name="Windows" />
 - **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)

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

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 (IContactDetailContentView)(IOpenMediaContactDetailContentView) <ref name="Contacts">Module: Genesyslab.Desktop.Modules.Contacts</ref> Updated in: 8.1.300.17
 - **Region:** ControlListAttachmentRegion
 - **View:** AttachmentView ControlListAttachmentView (IAttachmentView) <ref name="Windows" /> Updated in: 8.1.300.17
 - **View:** FormerContactDetailContentView (IContactDetailContentView)FormerContactDetailContentView) <ref name="Contacts">Module: Genesyslab.Desktop.Modules.Contacts</ref> Available since: 8.1.200.16, updated in: 8.1.300.17
 - **Region:** ControlListAttachmentRegion
 - **View:** AttachmentView ControlListAttachmentView (IAttachmentView) <ref name="Windows" /> Updated in: 8.1.300.17

Multi-view: MainToolbarContainerView

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

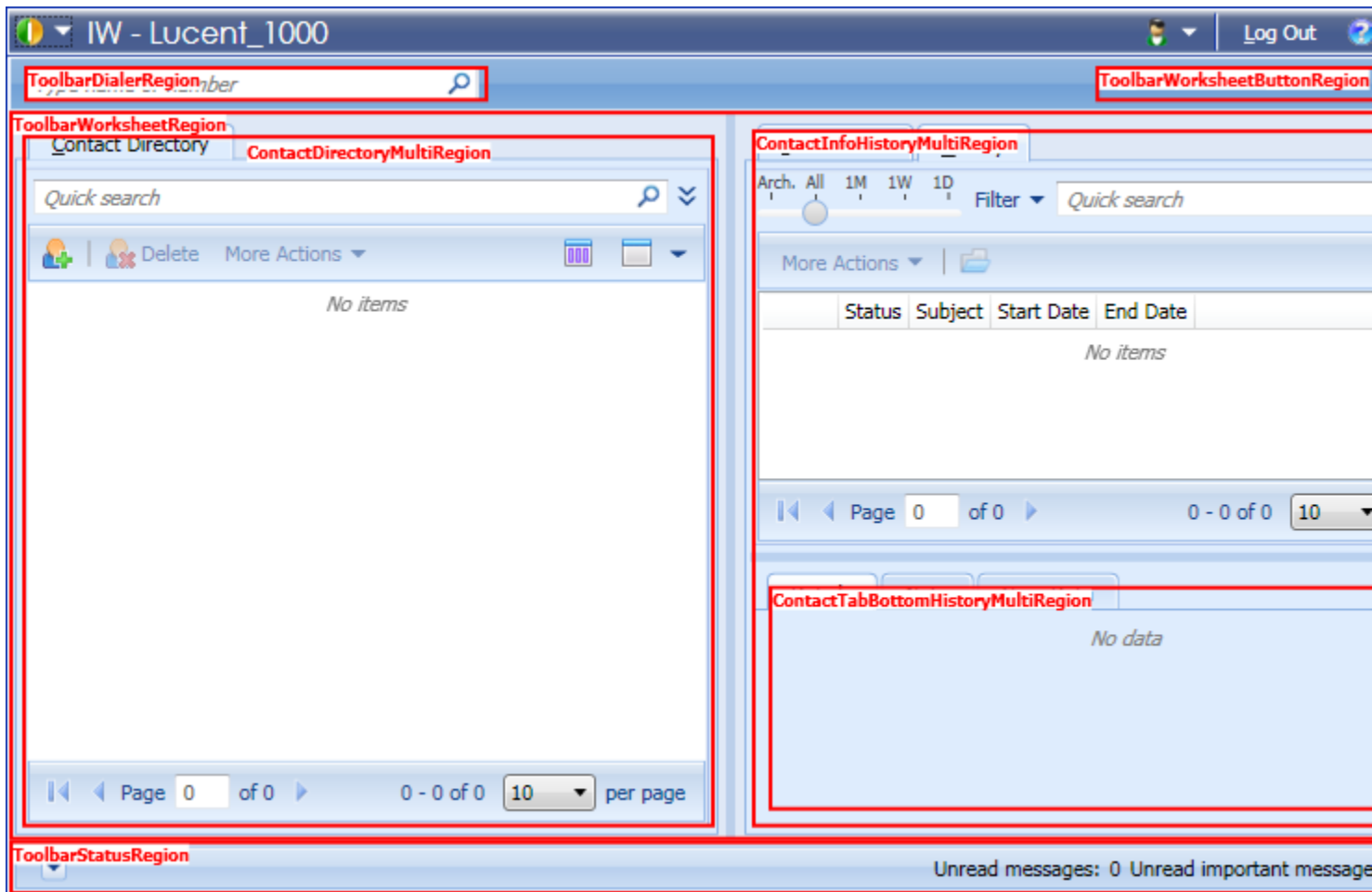
- **View:** MainToolbarContainerView (IMainToolbarContainerView) <ref name="Contacts" />

-
- **Region:** ContactDirectoryMultiRegion (Multi-view)
 - **View:** ContactDirectoryView (IContactDirectoryView) <ref name="Contacts" />
 - **Region:** ContactDirectoryErrorRegion
 - **View:** ContactDirectoryErrorView (ILastAlertErrorView) <ref name="Windows" />
 - **Region:** 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
Available since: 8.1.200.16
 - **View:** FormerContactDetailContentView FormerContactDetailContentView (IContactDetailContentViewFormerContactDetailContentView) <ref name="Contacts">Module: Genesyslab.Desktop.Modules.Contacts</ref>
Available since: 8.1.200.16, updated in: 8.1.300.17
 - **Region:** ControllistAttachmentRegion
 - **View:** AttachmentView ControlListAttachmentView (IAttachmentView) <ref name="Windows" />
Updated in: 8.1.300.17
 - **View:** OpenMediaContactDetailContentView (IContactDetailContentView)(IOpenMediaContactDetailContentView) <ref name="OpenMedia">Module: Genesyslab.Desktop.Modules.OpenMedia</ref>
Available since: 8.1.200.16, updated in: 8.1.300.17
 - **Region:** ControllistAttachmentRegion
 - **View:** AttachmentView AttachmentView ControlListAttachmentView (IAttachmentView) <ref name="Windows" />
Updated in: 8.1.300.17
 - **View:** NotepadView ContactHistoryNotepadView
Updated in: 8.1.300.17
(INotepadView) <ref name="Windows" />
 - **View:** ContactHistoryCaseDataView (IStaticCaseDataView) <ref name="Windows" />
 - **Region:** StaticCaseDataRegion
Available since: 8.1.200.16
 - **View:** StaticCaseDataView (IGenericDataView) <ref name="Windows" />

>Module: Genesyslab.Desktop.Modules.Windows</ref>

- **Region:** StaticDispositionCodeRegion
Available since: 8.1.200.16
- **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):



The Contact Button is selected in the ToolbarWorkSheetButtonRegion.

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:** WorkbinsExplorerView (IWorkbinsExplorerView) <ref name="OpenMedia" />
 - **Region:** WorkbinsExplorerViewErrorRegion
 - **View:** WorkbinsExplorerViewErrorView (ILastAlertErrorView) <ref name="Windows" />
 - **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

Available since: 8.1.200.16

 - **View:** FormerContactDetailContentView (IContactDetailContentView)(IOpenMediaContactDetailContentView) <ref name="Contacts">Module: Genesyslab.Desktop.Modules.Contacts</ref> Updated in: 8.1.300.17
 - **Region:** ControlListAttachmentRegion
 - **View:** AttachmentView ControlListAttachmentView (IAttachmentView) <ref name="Windows" /> Updated in: 8.1.300.17
 - **View:** OpenMediaContactDetailContentView (IContactDetailContentView) <ref name="OpenMedia">Module: Genesyslab.Desktop.Modules.OpenMedia</ref> Available since: 8.1.200.16
 - **Region:** ControlListAttachmentRegion
 - **View:** AttachmentView ControlListAttachmentView (IAttachmentView) <ref name="Windows" /> Updated in: 8.1.300.17
 - **View:** ContactHistoryNotepadView <ref name="Windows" /> Updated in: 8.1.300.17 (INotepadView)
 - **View:** ContactHistoryCaseDataView (IStaticCaseDataView) <ref name="Windows" />
 - **Region:** StaticCaseDataRegion

Available since: 8.1.200.16

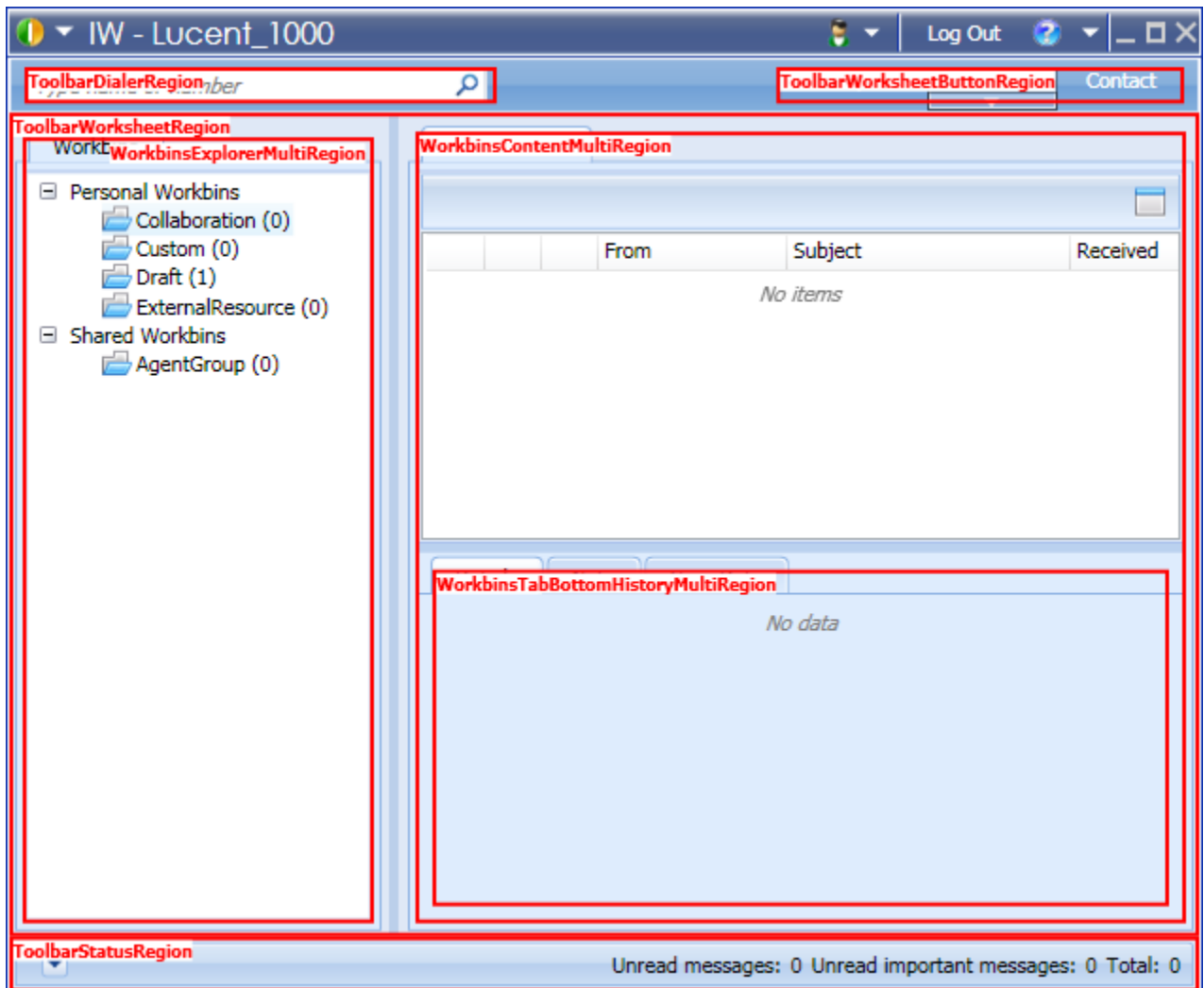
 - **View:** StaticCaseDataView (IGenericDataView) <ref name="Windows" >Module: Genesyslab.Desktop.Modules.Windows</ref>
 - **Region:** StaticDispositionCodeRegion

Available since: 8.1.200.16

 - **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.

References

<references />

Windows Utilities



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

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.

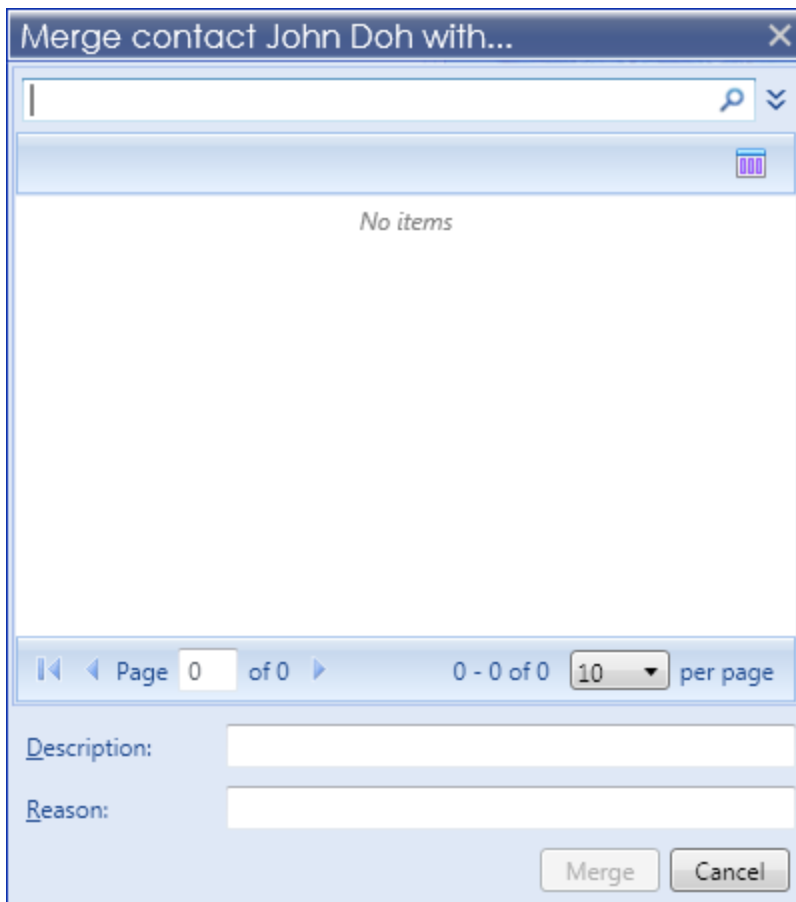
ContactSearchWindow

Available since: 8.1.300.17

- **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

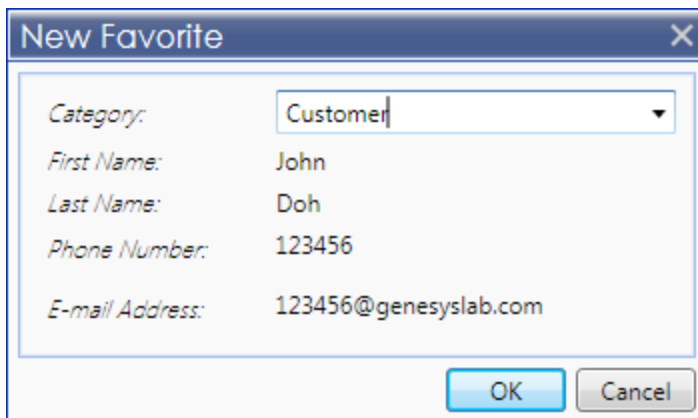
- **Window:** ContactMergeWindow (IContactMergeWindow) (Module: Genesyslab.Desktop.Modules.Contacts)
Available since: 8.1.300.17
- **Region:** MergeContactDirectoryRegion
 - **View:** MergeContactDirectoryView (IContactDirectoryView) (Module: Genesyslab.Desktop.Modules.Contacts)
 - **Region:** ContactDirectoryErrorRegion
 - **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)
Available since: 8.1.300.17
 - **Region:** FavoriteViewRegion
 - **View:** NewContactView (INewContactView) (Module: Genesyslab.Desktop.Modules.TeamCommunicator)
 - **Region:** FavoriteViewErrorRegion
 - **View:** FavoriteViewErrorView (ILastAlertErrorView) (Module: Genesyslab.Desktop.Modules.Windows)



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: 8.1.200.16

- **Window:** forward (IForwardView) <ref name="Windows">Genesyslab.Desktop.Modules.Windows</ref>
 - **Region:** ForwardErrorRegion
 - **View:** ForwardErrorView (ILastAlertErrorView) <ref name="Windows" />
- **Window:** CancelForwardView (ICancelForwardView) (Module: Genesyslab.Desktop.Modules.Windows)
Available since: 8.1.300.17

Additional Accessories

Available since: 8.1.300.17

- **Window:** IWMessageBoxView (IIWMessageBoxView) (Module: Genesyslab.Desktop.Modules.Windows)
- **Window:** AboutWindow (IAboutWindow) (Module: Genesyslab.Desktop.Modules.Windows)
- **Window:** AuthenticationWindow (IAuthenticationWindow) (Module: Genesyslab.Desktop.Modules.Windows)
- **Window:** AuthentificationPasswordWindow (IAuthentificationPasswordWindow) (Module: Genesyslab.Desktop.Modules.Windows)
- **Window:** BroadcastMessageWindow (IBroadcastMessageWindow) (Module: Genesyslab.Desktop.Modules.Windows)
- **Window:** MediaInformationView (IMediaView) (Module: Genesyslab.Desktop.Modules.Windows)
 - **Region:** ForwardErrorRegion

- **View:** ForwardErrorView (ILastAlertErrorView) (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

<references />