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

Frequently Asked Questions

4/16/2025

Frequently Asked Questions

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

Contents

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

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

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

How can I subscribe to/unsubscribe from 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);
// unsubcribe
ContainerAccessPoint.Container.Resolve<IEventAggregator>().GetEvent<AlertEvent>().Unsubscribe(AlertEventHandler
```