Custom Localization of Workspace Desktop Edition

Workspace Desktop Edition 8.5.1

1/4/2022

Table of Contents

Custom Localization of Workspace

Introduction

The objective of this document is to provide guidelines to Genesys customers that would like to leverage localization capabilities of **Genesys Workspace Desktop Edition** (Workspace) to build their own localization of the product, either by translating in a brand new language, or by modifying the terms used in out-of-the-box product localization. **Legal Notice: Customer enabled language translation functionality is provided on an as-is basis for internal use only. Outside distribution of this functionality and/or any translation(s) created using such functionality is not permitted except under a separate agreement negotiated with Genesys specifically for the purpose of distributing Genesys-related translations.**

General Principles of Localization in Workspace

Definition of the Language Pack

A Language Pack is a set of XML files that define the string that is used to display the Workspace User Interface in a given language. The language packs are stored in a directory named Languages that is located at the same level as the Workspace executable (actual directory structure might vary depending on the installation mode that was used). This directory contains a set of XML dictionary files for one or several languages. The string-definitions of a specific language are the compilation of keys that are declared in the set of XML files that have the same language-code/country-code extension (for example: `en-US`). Language codes are based on ISO 639-1, and Country codes are based on ISO 3166-1. The file name specifies the language in the following format: `module_name.language_code-country_code.xml` (for example: `Genesyslab.Desktop.Modules.Core.en-US.xml`)

Description of the content of a Language Pack

The table **Workspace Language Pack Description** displays the content of the out-of-the-box Workspace Language Pack for English-US.

Workspace Language Pack Description

Dictionary File Name	Privileges that control the use of the dictionary file	Notes
Genesyslab.Desktop.Modules.Contacts.en-US.xml	InteractionWorkspace.Contacts.*	
Genesyslab.Desktop.Modules.Core.en-US.xml	Any	Core dictionary

Dictionary File Name	Privileges that control the use of the dictionary file	Notes
Genesyslab.Desktop.Modules.OpenMedia.en-US.xml	InteractionWorkspace.Email.* InteractionWorkspace.Chat.* InteractionWorkspace.SMS.* InteractionWorkspace.WorkItem.* InteractionWorkspace.Workbins.*	
Genesyslab.Desktop.Modules.Outbound.en-US.xml	InteractionWorkspace.Outbound.*	
Genesyslab.Desktop.Modules.SIPEndpointCommunication.en-US.xml	InteractionWorkspace.SIP.*	
Genesyslab.Desktop.Modules.StandardResponse.en-US.xml	InteractionWorkspace.SRL.*	
Genesyslab.Desktop.Modules.Statistics.en-US.xml	InteractionWorkspace.ObjectStatistics.* InteractionWorkspace.GadgetStatistics.* InteractionWorkspace.KPI.*	
Genesyslab.Desktop.Modules.TeamCommunicator.en-US.xml	InteractionWorkspace.TeamCommunicator.*	
Genesyslab.Desktop.Modules.Voice.en-US.xml	InteractionWorkspace.Voice.* InteractionWorkspace.IM.*	
Genesyslab.Desktop.Modules.Windows.en-US.xml	Any	Core dictionary

Depending on the Privileges that you assign to agents, you can avoid translating the dictionary files that correspond to modules that you have not installed. This reduces the effort of building your language pack.

Adding a new language

Goal

Deliver to agents a translation of Workspace application that is not yet available from Genesys.

Procedures

Creating a new Language Pack

Use the following steps to create a new language pack for Workspace.

1. Duplicate all of the XML Dictionary files of the existing language pack that you want to use as a reference (for example, English-US) and rename them by using the language and country codes of the new language.

2. Edit each dictionary file of the duplicated file set and translate to the new language the text that is contained in the following attributes: `Name`, `String`, `Header`, `ToolTip`, `Text`, `Content`, `Title`, `ToolTipStatus`, `Watermark`, `Error`, and `ActionButtonToolTip`
    - Translation of strings that contain the underscore character "_" must also contain an underscored character. These strings are represented as an Access Key in the User Interface (they are displayed as underlined when the agent presses the ALT key to directly access the control)
    - Translation of strings that contain placeholder character sequences such as `{<digit>}` (for example `{0}`) must also contain those character sequences.
    - Keys that are surrounded by the '<!--' and '-->' strings do not require translation
3. Once all the keys are translated, copy and paste the new XML file set for the new language into the `Languages` directory.

Deploying the new Language Pack

You cannot modify the Language Pack IP to deploy the custom language using the same out-of-the-box method that you use for Genesys-distributed language packs. To deploy in a custom language, you must build a combined package, using both the out-of-the-box Workspace files, as well as a custom Language Pack that you create.

Direct installation on Workstation

Add the new Language Pack file into the structure of an existing deployment so that the structure matches the following example:

```
[distribution_root]

    *.dlls
    interactionworkspace.exe
    Languages

        *.en-US.xml [out-of-the-box language pack]
        *.<new_language_code>-< new_country_code>.xml [custom language pack]

    Miscellaneous out-of-the-box directories and files
```

Installation through ClickOnce

Copy the files of the new Language Pack into the directory structure that was generated when Workspace was installed in "centralized deployment mode". The directory structure must match the following example:

```
[distribution_root]

    *.dlls
    interactionworkspace.exe
    Languages

        *.en-US.xml [out-of-the-box language pack]
        *.<new_language_code>-< new_country_code>.xml [custom language pack]

    Miscellaneous out-of-the-box directories and files
    InteractionWorkspaceDeploymentManager

        interactionworkspacedeploymentmanager.exe
        Misc out-of-the-box deployment manager files
```

Execute Workspace Deployment Manager and follow the steps to deploy Workspace by using the

Wizard interface.

Note: If you have already generated a ClickOnce package that did not contain the new Language Pack files, you must follow the following steps and redeploy Workspace.

1. In the "Package Information" step, check the `Add custom files` box.
2. Click `Next.`
3. Select all of the files (`InteractionWorkspace\Languages\*.<new_language_code>-<new_country_code>.xml`) of the new Language Pack in addition to you regular custom files, if any.
4. Finalize the execution of the wizard

Using the New Language Pack

Follow these steps to start Workspace and set it to start in your new language by default:

1. Restart the updated distribution of Workspace (from direct installation or ClickOnce).<liL
2. The Login window opens and displays a drop-down list that contains the original language(s) as well as the new language that you added.
3. Select an alternative language *immediately* to change the display to the new language. All windows that are subsequently opened are also displayed in the new language.

The Logic of Language Selection at Login Time

Language recognition works in the following way:

- The Languages directory contains a set of XML dictionary files for one or several languages. The file name reflect the language:

`module_name.language_code-country_code.xml' (for example: Genesyslab.Desktop.Modules.Core.en-US.xml)`

- If at least one language XML file has a particular language/country extension (For example: `fr-FR`), the corresponding locale is considered available at run-time.
- All available languages (if number of languages is greater than or equal to 2) are presented in a combo-box on the first login page.
- The default selection in the combo-box is set by using the following logic:
  - If the current user workstation locale (language and country) is part of the available languages, then the default value is set to the user locale
  - If the current user workstation locale (language and country) is not part of the available languages, Workspace searches for an available language that has the language code in common. For example, if `fr-FR` is available but the user locale is `fr-CA`, then the default language is `fr-FR`
  - If no language or country code can be matched, then the default language is set to `English-US`
- The agent can select manually another language from the list of available languages.

- After the initial login, the language that was used during the previous session is automatically restored as the default language. It is always possible for the agent to select another language.

Notes and Recommendations:

- Some strings in the User Interface might be displayed in the original language after the new language is selected even if all keys that are contained in the dictionary files have been translated. The reason for this is that some strings come from the configuration layer and are not pre-configured in the dictionary file.
- Editing the `EnglishName`, `CultureName`, and `Culture` XML headers as shown here is recommended but not mandatory:

```
<Dictionary EnglishName="French" CultureName="Canada" Culture=" fr-CA ">
```

- Right-to-left reading languages, such as Arabic and Hebrew, are not supported.
- Carefully encode the dictionary files that you create to ensure that there is no data loss. Genesys recommends that you save your XML file in UTF8 format.

Customizing Display Names for Configuration Objects

[**Modified:** 8.5.109.16, 8.5.120.05, 8.5.143.08, WSEP 8.5.114.05]

[**Added:** 8.5.101.14]

Workspace uses the `display-format.*` configuration options to enable you to customize how different interface elements are displayed to agents. They are also used to customize the display names of Framework objects, such as Routing Points and Queues, in the agent interface.

> ### Tip
> This article is a work in progress. Additional use cases and examples will be added in the future.

Customizing display names for configuration objects

The `display-format.*` configuration options enable you to specify the data source for the display names of different objects. For example:

1. System attributes of configuration objects (for example, Routing Point Number or Agent Group Name)
2. Custom Display names defined in custom dictionary files
3. Custom Display names defined in the object Annex
4. If none of the defined sources contain an actual string to display, Workspace displays the mandatory

attribute value that is identified as the default for a particular object (for example, 'Number' for a DN).

You can use the `display-format.*` options to define multiple sources, separated by the ‘`|`’ character, which specifies the precedence order of the sources. Precedence order if from left to right. If nothing is defined for the first source, the next one is checked, and so on.

Configuring object display names using custom dictionary files

[**Added:** 8.5.101.14]

Starting with Workspace 8.5.100x.xx, the list of field codes for these options is extended to take local dictionary entries into account. For example, for the `display-format.routing-point.name` option, the following key is supported: `$RoutingPoint.DictionaryValue$`. This key selects the name for the routing point based on the value specified for it in the language dictionary file.

For the display-format.action-code.name option, if the `$ActionCode.DictionaryValue$` key is specified, then the value for the action code will be selected from the dictionary that corresponds to the language that the agent specified at login.

For example, in `custom.en-US.xml`, the action code might be specified as:

```
{code}
<Value Id="ActionCode.Break" Text="Break"/>
{code}
```

And, in `custom.fr-FR.xml`, the same action code might be specified as:

```
{code}
<Value Id="ActionCode.Break" Text="Pause"/>
{code}
```

The following is an example of a custom dictionary that specifies alternate text for configuration objects:

```
<?xml version="1.0" encoding="utf-8" ?>
<Dictionary EnglishName="English" CultureName="English" Culture="en-US">
 <!-- [<Tenant>].<object-type>.[<switch>].<object-identifier> (where [<Tenant>] and
[<switch>] are optional) -->

 <Value Id="defaultTenant.RoutingPoint.LucentG3.122" Text="Routing Point 122" />
 <Value Id="defaultTenant.ACDQueue.LucentG3.80001" Text="ACD Queue 80001" />
 <Value Id="defaultTenant.InteractionQueue.any-queue-to-agent-group-8002" Text="Interaction
Queue for Agent Group 8002" />
 <Value Id="defaultTenant.InteractionQueue.email-routing-queue-inbound" Text="Interaction
Queue for inbound emails" />
 <Value Id="defaultTenant.ActionCode.Break" Text="Coffee Break" />
 <Value Id="defaultTenant.Workbin.email-draft-wb" Text="Rough copy Emails" />
 <Value Id="defaultTenant.BusinessAttribute.DispositionCode" Text="Disposal Code" />
 <Value Id="defaultTenant.BusinessAttribute.DispositionCode.DC_Accepted" Text="Taken" />
 <Value Id="defaultTenant.Skill.Email-QualityConfidencePercentageSkill" Text="Email skill" />
 <Value Id="defaultTenant.AgentGroup.Agent Group 80001" Text="Lucent Agent Group 80001" />
</Dictionary>
```

Configuring display names in the Object Annex

[**Added:** 8.5.109.16] [**Modified:** 8.5.120.05, WSEP 8.5.114.05]

You can add a display name for certain configuration layer objects. This feature enables you to name objects without relying on a local dictionary file. This feature makes localization and centralization more efficient.

This method provides a way to configure the display name centrally instead of locally in a dictionary file by enabling you to specify key-value pairs in object Annex. Key-value pairs can be defined for a default language and for localization in alternative languages.

> ### Important
>
> To store a display name in an object Annex as a string value that does not use the character set that is used by Configuration Server, Configuration Server 8.1.3 or higher must be installed and its multi-language capability must be enabled. For example, using French accented characters in a U.S. English deployment or using Korean characters in a Japanese deployment. Refer to Deploying Genesys for Key Mixed Language Scenarios for information about supporting multi-language deployment.

For each object, the default display name is defined by using the `interaction-workspace/display-name = <value>` key-value pair.

For each object, additional language display names are defined by using the `interaction-workspace/display-name.<ISO_language_code>-<ISO_country_code>` = <value> key-value pair (for example, `display-name.fr-CA` for French Canadian.

The following precedence rules are followed for display names:

- If a particular object type is configured to be displayed through the configuration Annex, Workspace tries at runtime to find a key-value pair that matches the locale that selected by the agent at login time.
- If no match is found, Workspace uses the default `display-name=value` key-value pair.
- If no default key-value pair is found, Workspace the value that is defined by the `display-format.*` options.

The following is an example of a display name in two different languages, the default, and French (France), that could be defined in the object Annex of a Routing Point:

- `interaction-workspace/display-name = 'Technical Support Queue'`: Used if there is no matching language specific entry for the locale selected by the agent in the login window.
- `interaction-workspace/display-name.fr-FR = 'File d'Attente du Support Technique'`: Used if the locale selected by the agent in login window is French (France) (fr-FR)

Use the display-format.folder.name option to specify the display format of folders that are displayed in the Disposition Code and Case Data views. You can localize the folder name using the `$Folder.Name$` parameter. [**Added:** 8.5.120.05, WSEP 8.5.114.05]

Summary of Display Format options

[**Modified:** 8.5.109.16]

The table **Summary of Display Format Options** contains descriptions of all of the new and updated display-format configuration options that use the Dictionary Value keys to support this feature.

Summary of Display Format Options

Object Type	Option Name	Field codes to be used in display-format.* options	Dictionary Key Format to be spcified in the XML dictionary if the $DictionaryValue$ is specified in display-format.* options	Views affected by this option
DN - ACD Queue	display-format.acd-queue.name	$ACDQueue.DictionaryValue$, $ACDQueue.AnnexValue$	[<Tenant>].ACDQueue.<switch>.<dn-number>	Team Communicator Login views
Action Code	display-format.action-code.name	$ActionCode.DictionaryValue$, $ActionCode.AnnexValue$	[<Tenant>].ActionCode.<action-code-name>	Global Agent Status Control (tooltip + drop down menu) My Channels view
Agent Group	display-format.agent-group.name	$AgentGroup.DictionaryValue$, $AgentGroup.AnnexValue$	[<Tenant>].AgentGroup.<agent-group-name>	Team Communicator My Statistics Voice Mail (shared voice mail boxes)
Business Attribute	display-format.business-attribute.name	$BusinessAttribute.DictionaryValue$, $BusinessAttribute.AnnexValue$	[<Tenant>].BusinessAttribute.<business-attribute-name>	Interaction View (Case Information, Disposition Code) Contact Directory Contact Profile Contact History (detail - case data)
Business Attribute	display-format.business-attribute.name	$BusinessAttribute.DictionaryValue$, $BusinessAttribute.AnnexValue$	[<Tenant>].BusinessAttribute.<business-attribute-name>	Interaction View (Case Information, Disposition Code) Contact Directory Contact Profile Contact History (detail - case data)
Business Attribute Value			[<Tenant>].BusinessAttribute.<business-attribute-name>.<business-attribute-value-name>	

Object Type	Option Name	Field codes to be used in display-format.* options	Dictionary Key Format to be spcified in the XML dictionary if the $DictionaryValue$ is specified in display-format.* options	Views affected by this option
				Media type (login, my channels, agent status, and so on)
Folder	display-format.folder.name	$Folder.AnnexValue$, $Folder.DictionaryValue$, $Folder.Name$		Interaction View (Disposition Code, enum-tree KVP in Case Information and Outbound Record)
Script - Interaction Queue	display-format.interaction-queue.name	$InteractionQueue.DictionaryValue$, $InteractionQueue.AnnexValue$	[<Tenant>].InteractionQueue.<script-name>	Team Communicator Workbins view (My Interaction Queues)
DN - Routing Point	display-format.routing-point.name	$RoutingPoint.DictionaryValue$, $RoutingPoint.AnnexValue$	[<Tenant>].RoutingPoint.<switch>.<dn-number>	Team Communicator Login views
Skill	display-format.skill.name	$Skill.DictionaryValue$, $Skill.AnnexValue$	[<Tenant>].Skill.<skill-name>	Team Communicator
DN - Virtual Queues	display-format.virtual-queue.name	$VirtualQueue.DictionaryValue$, $VirtualQueue.AnnexValue$	[<Tenant>].VirtualQueue.<switch>.<dn-number>	Login Views
Script - Interaction Workbin	display-format.workbin.name	$Workbin.DictionaryValue$, $Workbin.AnnexValue$	[<Tenant>].Workbin.<script-name>	Workbins view (My Workbin, My Team Workbins)

Masking a contact's phone number on inbound and outbound interaction views

[**Added:** 8.5.143.08]

Protecting the security and privacy of contacts is an important consideration, especially if you have agents who are working from home. You can use the configuration options in this section to hide the contact's phone number in the Interaction window. You can modify the default values of the configuration options that control the contact information that is displayed in inbound and outbound interactions. Workspace also supports plain text instead of field codes in the values of the following options:

- display-format.caller-name
- display-format.case-name-format
- display-format.customer-name-format
- display-format.interaction-callback-name
- display-format.interaction-outbound-pull-preview-name
- display-format.interaction-voice-name
- display-format.party-name-format
- interaction.window-title

For example, you can modify the default value of display-format.caller-name to replace `$Interaction.MainParty$` with plain text such as XXXXXXXXXXX or `Hidden Phone Number`.

For Outbound interactions, use the following configuration options and replace the `$OutboundRecord.PhoneNumber$` value with plain text to modify what your agents see in the Outbound interaction view:

- display-format.outbound-record-name: Specifies how an Outbound Record from a Record Chain is displayed when presented to an agent. The content is populated based on record attributes by a string that contains the following field codes: `$OutboundRecord.PhoneType$`, `$OutboundRecord.PhoneNumber$`,`$OutboundRecordField.X$`, where `X` is the name of the custom outbound field. [**Added:** 8.5.143.08]
- display-format.caller-name: Specifies the content of the 'Origin' field of the Case Information area. This option is enabled when the value of the interaction.case-data.content option contains the `History` key. This content is typically used when placing an outbound call where the origin contains a string such as "outbound call to xxx". The content is populated based on attached data keys or contact attributes (if there is a contact) or outbound record fields (if there is an outbound record) that are defined by a string that contains the following field codes: `$Interaction.CaseId$`,`$Interaction.MainParty$`, `$Contact.X$`,`$AttachedData.Y$`,`$OutboundRecord.PhoneType$`,`$OutboundRecord.PhoneNumber$`, `$OutboundRecordField.Z$`, where `X` is the name of contact attribute, `Y` is the attached data key name, and `Z` is the name of custom outbound field. If the values of the default field codes are empty, the following field code is used: `$Interaction.MainParty$`.

[**Modified:** 8.5.143.08]

Customizing the Wording of an Existing Language

Goal

Replace the wording of an out-of-the-box Workspace string with a string that more closely matches your business context.

Procedure

Use the following steps to create a new language pack for Workspace that has the same name as an existing Workspace language pack.

1. Create a new XML Dictionary file that uses the same naming policy as the out-of-the-box Workspace language files: `custom _name.language_code-country_code.xml`
2. Set the following XML header to specify that it is a custom dictionary to ensure that the definitions in this file override the default strings of this language: `<Dictionary CustomDictionary="true">`
3. Identify the key that you want to customize in one of the original files of the given language and replicate it in your new custom file.

   For example: `<Value Id="State.Agent.Ready" Text="Ready" String="Ready"/>`
4. Modify the text that is contained in the attributes of the key that was copied into the custom file. You can modify the types of the following attributes:
   - Name
   - String
   - Header
   - ToolTip
   - Text
   - Content
   - Title
   - ToolTipStatus
   - Watermark
   - Error
   - ActionButtonToolTip
5. Copy the resulting XML Dictionary file into the `Languages` directory.
6. Restart Workspace.

Notes and Recommendations:

- The only mandatory XML header for a custom file is `CustomDictionary`.
- The following XML headers are optional: `EnglishName`, `CultureName`, and `Culture`. For example: `<Dictionary EnglishName="English" CultureName="English" Culture="en-US" CustomDictionary="true">`.

Potential Errors and Consequences

The table **Workspace Language Errors** displays the most common errors that you might encounter when you create custom language packs.

Workspace Language Errors

Error Message	Consequence
A particular key is not present in any of the files of the new language	The interface will display the value of the default language (en-US).
Malformed XML formatting	It is not possible to switch to the new language.
Translated string is very long	String might appear truncated in the application when the new language is selected.
Keys surrounded by '<!--' and '-->' are translated	Content is not displayed in the interface.
The characters "{0}" are in the original file but are missing in the translation	Some strings are not displayed.

Upgrade Considerations

When a new version of Workspace is released, the default Language Pack might be updated. An update of the Language Pack might consist of the following changes:

- New keys added to an existing dictionary file
- New dictionary file(s) that correspond to new a privilege(s)
- Removal of a key that existed in a previous version of the product
- Change of a string(s) that is associated with a key that existed in previous version of the product
- Movement of a key that existed in previous version of the product from one dictionary file to another

To avoid errors, an upgrade of custom language packs must be planned to ensure that the Workspace User Interface displays correctly in the new version.

Localizing Custom Content in Workspace

Refer to the Workspace Developer's Guide for information about localizing Workspace.

Custom Localization of Workspace

Introduction

The objective of this document is to provide guidelines to Genesys customers that would like to leverage localization capabilities of **Genesys Workspace Desktop Edition** (Workspace) to build their own localization of the product, either by translating in a brand new language, or by modifying the terms used in out-of-the-box product localization. **Legal Notice: Customer enabled language translation functionality is provided on an as-is basis for internal use only. Outside distribution of this functionality and/or any translation(s) created using such functionality is not permitted except under a separate agreement negotiated with Genesys specifically for the purpose of distributing Genesys-related translations.**

General Principles of Localization in Workspace

Definition of the Language Pack

A Language Pack is a set of XML files that define the string that is used to display the Workspace User Interface in a given language. The language packs are stored in a directory named Languages that is located at the same level as the Workspace executable (actual directory structure might vary depending on the installation mode that was used). This directory contains a set of XML dictionary files for one or several languages. The string-definitions of a specific language are the compilation of keys that are declared in the set of XML files that have the same language-code/country-code extension (for example: `en-US`). Language codes are based on ISO 639-1, and Country codes are based on ISO 3166-1. The file name specifies the language in the following format: `module_name.language_code-country_code.xml` (for example: `Genesyslab.Desktop.Modules.Core.en-US.xml`)

Description of the content of a Language Pack

The table **Workspace Language Pack Description** displays the content of the out-of-the-box Workspace Language Pack for English-US.

Workspace Language Pack Description

Dictionary File Name	Privileges that control the use of the dictionary file	Notes
Genesyslab.Desktop.Modules.Contacts.en-US.xml	InteractionWorkspace.Contacts.*	
Genesyslab.Desktop.Modules.Core.en-US.xml	Any	Core dictionary

Dictionary File Name	Privileges that control the use of the dictionary file	Notes
Genesyslab.Desktop.Modules.OpenMedia.en-US.xml	InteractionWorkspace.Email.* InteractionWorkspace.Chat.* InteractionWorkspace.SMS.* InteractionWorkspace.WorkItem.* InteractionWorkspace.Workbins.*	
Genesyslab.Desktop.Modules.Outbound.en-US.xml	InteractionWorkspace.Outbound.*	
Genesyslab.Desktop.Modules.SIPEndpointCommunication.en-US.xml	InteractionWorkspace.SIP.*	
Genesyslab.Desktop.Modules.StandardResponse.en-US.xml	InteractionWorkspace.SRL.*	
Genesyslab.Desktop.Modules.Statistics.en-US.xml	InteractionWorkspace.ObjectStatistics.* InteractionWorkspace.GadgetStatistics.* InteractionWorkspace.KPI.*	
Genesyslab.Desktop.Modules.TeamCommunicator.en-US.xml	InteractionWorkspace.TeamCommunicator.*	
Genesyslab.Desktop.Modules.Voice.en-US.xml	InteractionWorkspace.Voice.* InteractionWorkspace.IM.*	
Genesyslab.Desktop.Modules.Windows.en-US.xml	Any	Core dictionary

Depending on the Privileges that you assign to agents, you can avoid translating the dictionary files that correspond to modules that you have not installed. This reduces the effort of building your language pack.

Adding a new language

Goal

Deliver to agents a translation of Workspace application that is not yet available from Genesys.

Procedures

Creating a new Language Pack

Use the following steps to create a new language pack for Workspace.

1. Duplicate all of the XML Dictionary files of the existing language pack that you want to use as a reference (for example, English-US) and rename them by using the language and country codes of the new language.

2. Edit each dictionary file of the duplicated file set and translate to the new language the text that is contained in the following attributes: `Name`, `String`, `Header`, `ToolTip`, `Text`, `Content`, `Title`, `ToolTipStatus`, `Watermark`, `Error`, and `ActionButtonToolTip`
    - Translation of strings that contain the underscore character "_" must also contain an underscored character. These strings are represented as an Access Key in the User Interface (they are displayed as underlined when the agent presses the ALT key to directly access the control)
    - Translation of strings that contain placeholder character sequences such as `{<digit>}` (for example `{0}`) must also contain those character sequences.
    - Keys that are surrounded by the '<!--' and '-->' strings do not require translation
3. Once all the keys are translated, copy and paste the new XML file set for the new language into the `Languages` directory.

Deploying the new Language Pack

You cannot modify the Language Pack IP to deploy the custom language using the same out-of-the-box method that you use for Genesys-distributed language packs. To deploy in a custom language, you must build a combined package, using both the out-of-the-box Workspace files, as well as a custom Language Pack that you create.

Direct installation on Workstation

Add the new Language Pack file into the structure of an existing deployment so that the structure matches the following example:

```
[distribution_root]
    *.dlls
    interactionworkspace.exe
    Languages
        *.en-US.xml [out-of-the-box language pack]
        *.<new_language_code>-< new_country_code>.xml [custom language pack]
    Miscellaneous out-of-the-box directories and files
```

Installation through ClickOnce

Copy the files of the new Language Pack into the directory structure that was generated when Workspace was installed in "centralized deployment mode". The directory structure must match the following example:

```
[distribution_root]
    *.dlls
    interactionworkspace.exe
    Languages
        *.en-US.xml [out-of-the-box language pack]
        *.<new_language_code>-< new_country_code>.xml [custom language pack]
    Miscellaneous out-of-the-box directories and files
    InteractionWorkspaceDeploymentManager
        interactionworkspacedeploymentmanager.exe
        Misc out-of-the-box deployment manager files
```

Execute Workspace Deployment Manager and follow the steps to deploy Workspace by using the

Wizard interface.

	Note: If you have already generated a ClickOnce package that did not contain the new Language Pack files, you must follow the following steps and redeploy Workspace.

1. In the "Package Information" step, check the `Add custom files` box.
2. Click `Next.`
3. Select all of the files (`InteractionWorkspace\Languages\*.<new_language_code>-<new_country_code>.xml`) of the new Language Pack in addition to you regular custom files, if any.
4. Finalize the execution of the wizard

Using the New Language Pack

Follow these steps to start Workspace and set it to start in your new language by default:

1. Restart the updated distribution of Workspace (from direct installation or ClickOnce).<liL
2. The Login window opens and displays a drop-down list that contains the original language(s) as well as the new language that you added.
3. Select an alternative language *immediately* to change the display to the new language. All windows that are subsequently opened are also displayed in the new language.

The Logic of Language Selection at Login Time

Language recognition works in the following way:

- The Languages directory contains a set of XML dictionary files for one or several languages. The file name reflect the language:

```
module_name.language_code-country_code.xml’ (for example: Genesyslab.Desktop.Modules.Core.en-
US.xml)
```

- If at least one language XML file has a particular language/country extension (For example: `fr-FR`), the corresponding locale is considered available at run-time.
- All available languages (if number of languages is greater than or equal to 2) are presented in a combo-box on the first login page.
- The default selection in the combo-box is set by using the following logic:
  - If the current user workstation locale (language and country) is part of the available languages, then the default value is set to the user locale
  - If the current user workstation locale (language and country) is not part of the available languages, Workspace searches for an available language that has the language code in common. For example, if `fr-FR` is available but the user locale is `fr-CA`, then the default language is `fr-FR`
  - If no language or country code can be matched, then the default language is set to `English-US`
- The agent can select manually another language from the list of available languages.

- After the initial login, the language that was used during the previous session is automatically restored as the default language. It is always possible for the agent to select another language.

Notes and Recommendations:

- Some strings in the User Interface might be displayed in the original language after the new language is selected even if all keys that are contained in the dictionary files have been translated. The reason for this is that some strings come from the configuration layer and are not pre-configured in the dictionary file.
- Editing the `EnglishName`, `CultureName`, and `Culture` XML headers as shown here is recommended but not mandatory:

```
<Dictionary EnglishName="French" CultureName="Canada" Culture=" fr-CA ">
```

- Right-to-left reading languages, such as Arabic and Hebrew, are not supported.
- Carefully encode the dictionary files that you create to ensure that there is no data loss. Genesys recommends that you save your XML file in UTF8 format.

Customizing Display Names for Configuration Objects

[**Modified:** 8.5.109.16, 8.5.120.05, 8.5.143.08, WSEP 8.5.114.05]

[**Added:** 8.5.101.14]

Workspace uses the `display-format.*` configuration options to enable you to customize how different interface elements are displayed to agents. They are also used to customize the display names of Framework objects, such as Routing Points and Queues, in the agent interface.

> ### Tip
> This article is a work in progress. Additional use cases and examples will be added in the future.

Customizing display names for configuration objects

The `display-format.*` configuration options enable you to specify the data source for the display names of different objects. For example:

1. System attributes of configuration objects (for example, Routing Point Number or Agent Group Name)
2. Custom Display names defined in custom dictionary files
3. Custom Display names defined in the object Annex
4. If none of the defined sources contain an actual string to display, Workspace displays the mandatory

attribute value that is identified as the default for a particular object (for example, 'Number' for a DN).

You can use the `display-format.*` options to define multiple sources, separated by the '|' character, which specifies the precedence order of the sources. Precedence order if from left to right. If nothing is defined for the first source, the next one is checked, and so on.

Configuring object display names using custom dictionary files

[**Added:** 8.5.101.14]

Starting with Workspace 8.5.100x.xx, the list of field codes for these options is extended to take local dictionary entries into account. For example, for the `display-format.routing-point.name` option, the following key is supported: `$RoutingPoint.DictionaryValue$`. This key selects the name for the routing point based on the value specified for it in the language dictionary file.

For the display-format.action-code.name option, if the `$ActionCode.DictionaryValue$` key is specified, then the value for the action code will be selected from the dictionary that corresponds to the language that the agent specified at login.

For example, in `custom.en-US.xml`, the action code might be specified as:

```
{code}
<Value Id="ActionCode.Break" Text="Break"/>
{code}
```

And, in `custom.fr-FR.xml`, the same action code might be specified as:

```
{code}
<Value Id="ActionCode.Break" Text="Pause"/>
{code}
```

The following is an example of a custom dictionary that specifies alternate text for configuration objects:

```
<?xml version="1.0" encoding="utf-8" ?>
<Dictionary EnglishName="English" CultureName="English" Culture="en-US">
 <!-- [<Tenant>].<object-type>.[<switch>].<object-identifier> (where [<Tenant>] and
[<switch>] are optional) -->

 <Value Id="defaultTenant.RoutingPoint.LucentG3.122" Text="Routing Point 122" />
 <Value Id="defaultTenant.ACDQueue.LucentG3.80001" Text="ACD Queue 80001" />
 <Value Id="defaultTenant.InteractionQueue.any-queue-to-agent-group-8002" Text="Interaction
Queue for Agent Group 8002" />
 <Value Id="defaultTenant.InteractionQueue.email-routing-queue-inbound" Text="Interaction
Queue for inbound emails" />
 <Value Id="defaultTenant.ActionCode.Break" Text="Coffee Break" />
 <Value Id="defaultTenant.Workbin.email-draft-wb" Text="Rough copy Emails" />
 <Value Id="defaultTenant.BusinessAttribute.DispositionCode" Text="Disposal Code" />
 <Value Id="defaultTenant.BusinessAttribute.DispositionCode.DC_Accepted" Text="Taken" />
 <Value Id="defaultTenant.Skill.Email-QualityConfidencePercentageSkill" Text="Email skill" />
 <Value Id="defaultTenant.AgentGroup.Agent Group 80001" Text="Lucent Agent Group 80001" />
</Dictionary>
```

Configuring display names in the Object Annex

[**Added:** 8.5.109.16] [**Modified:** 8.5.120.05, WSEP 8.5.114.05]

You can add a display name for certain configuration layer objects. This feature enables you to name objects without relying on a local dictionary file. This feature makes localization and centralization more efficient.

This method provides a way to configure the display name centrally instead of locally in a dictionary file by enabling you to specify key-value pairs in object Annex. Key-value pairs can be defined for a default language and for localization in alternative languages.

> ### Important
>
> To store a display name in an object Annex as a string value that does not use the character set that is used by Configuration Server, Configuration Server 8.1.3 or higher must be installed and its multi-language capability must be enabled. For example, using French accented characters in a U.S. English deployment or using Korean characters in a Japanese deployment. Refer to Deploying Genesys for Key Mixed Language Scenarios for information about supporting multi-language deployment.

For each object, the default display name is defined by using the `interaction-workspace/display-name` = <`value`> key-value pair.

For each object, additional language display names are defined by using the `interaction-workspace/display-name.<ISO_language_code>-<ISO_country_code>` = <value> key-value pair (for example, `display-name.fr-CA` for French Canadian.

The following precedence rules are followed for display names:

- If a particular object type is configured to be displayed through the configuration Annex, Workspace tries at runtime to find a key-value pair that matches the locale that selected by the agent at login time.
- If no match is found, Workspace uses the default `display-name=value` key-value pair.
- If no default key-value pair is found, Workspace the value that is defined by the `display-format.*` options.

The following is an example of a display name in two different languages, the default, and French (France), that could be defined in the object Annex of a Routing Point:

- `interaction-workspace/display-name = 'Technical Support Queue'`: Used if there is no matching language specific entry for the locale selected by the agent in the login window.
- `interaction-workspace/display-name.fr-FR = 'File d'Attente du Support Technique'`: Used if the locale selected by the agent in login window is French (France) (fr-FR)

Use the display-format.folder.name option to specify the display format of folders that are displayed in the Disposition Code and Case Data views. You can localize the folder name using the `$Folder.Name$` parameter. [**Added:** 8.5.120.05, WSEP 8.5.114.05]

Summary of Display Format options

[**Modified:** 8.5.109.16]

The table **Summary of Display Format Options** contains descriptions of all of the new and updated display-format configuration options that use the Dictionary Value keys to support this feature.

Summary of Display Format Options

Object Type	Option Name	Field codes to be used in display-format.* options	Dictionary Key Format to be spcified in the XML dictionary if the $DictionaryValue$ is specified in display-format.* options	Views affected by this option
DN - ACD Queue	display-format.acd-queue.name	$ACDQueue.DictionaryValue$, $ACDQueue.AnnexValue$	[<Tenant>].ACDQueue.<switch>.<dn-number>	Team Communicator Login views
Action Code	display-format.action-code.name	$ActionCode.DictionaryValue$, $ActionCode.AnnexValue$	[<Tenant>].ActionCode.<action-code-name>	Global Agent Status Control (tooltip + drop down menu) My Channels view
Agent Group	display-format.agent-group.name	$AgentGroup.DictionaryValue$, $AgentGroup.AnnexValue$	[<Tenant>].AgentGroup.<agent-group-name>	Team Communicator My Statistics Voice Mail (shared voice mail boxes)
Business Attribute	display-format.business-attribute.name	$BusinessAttribute.DictionaryValue$, $BusinessAttribute.AnnexValue$	[<Tenant>].BusinessAttribute.<business-attribute-name>	Interaction View (Case Information, Disposition Code) Contact Directory Contact Profile Contact History (detail - case data)
Business Attribute	display-format.business-attribute.name	$BusinessAttribute.DictionaryValue$, $BusinessAttribute.AnnexValue$	[<Tenant>].BusinessAttribute.<business-attribute-name>	Interaction View (Case Information, Disposition Code) Contact Directory Contact Profile Contact History (detail - case data)
Business Attribute Value			[<Tenant>].BusinessAttribute.<business-attribute-name>.<business-attribute-value-name>	

Object Type	Option Name	Field codes to be used in display-format.* options	Dictionary Key Format to be spcified in the XML dictionary if the $DictionaryValue$ is specified in display-format.* options	Views affected by this option
				Media type (login, my channels, agent status, and so on)
Folder	display-format.folder.name	$Folder.AnnexValue$, $Folder.DictionaryValue$, $Folder.Name$		Interaction View (Disposition Code, enum-tree KVP in Case Information and Outbound Record)
Script - Interaction Queue	display-format.interaction-queue.name	$InteractionQueue.DictionaryValue$, $InteractionQueue.AnnexValue$	[<Tenant>].InteractionQueue.<script-name>	Team Communicator Workbins view (My Interaction Queues)
DN - Routing Point	display-format.routing-point.name	$RoutingPoint.DictionaryValue$, $RoutingPoint.AnnexValue$	[<Tenant>].RoutingPoint.<switch>.<dn-number>	Team Communicator Login views
Skill	display-format.skill.name	$Skill.DictionaryValue$, $Skill.AnnexValue$	[<Tenant>].Skill.<skill-name>	Team Communicator
DN - Virtual Queues	display-format.virtual-queue.name	$VirtualQueue.DictionaryValue$, $VirtualQueue.AnnexValue$	[<Tenant>].VirtualQueue.<switch>.<dn-number>	Login Views
Script - Interaction Workbin	display-format.workbin.name	$Workbin.DictionaryValue$, $Workbin.AnnexValue$	[<Tenant>].Workbin.<script-name>	Workbins view (My Workbin, My Team Workbins)

Masking a contact's phone number on inbound and outbound interaction views

[**Added:** 8.5.143.08]

Protecting the security and privacy of contacts is an important consideration, especially if you have agents who are working from home. You can use the configuration options in this section to hide the contact's phone number in the Interaction window. You can modify the default values of the configuration options that control the contact information that is displayed in inbound and outbound interactions. Workspace also supports plain text instead of field codes in the values of the following options:

- display-format.caller-name
- display-format.case-name-format
- display-format.customer-name-format
- display-format.interaction-callback-name
- display-format.interaction-outbound-pull-preview-name
- display-format.interaction-voice-name
- display-format.party-name-format
- interaction.window-title

For example, you can modify the default value of display-format.caller-name to replace `$Interaction.MainParty$` with plain text such as XXXXXXXXXXX or `Hidden Phone Number`.

For Outbound interactions, use the following configuration options and replace the `$OutboundRecord.PhoneNumber$` value with plain text to modify what your agents see in the Outbound interaction view:

- display-format.outbound-record-name: Specifies how an Outbound Record from a Record Chain is displayed when presented to an agent. The content is populated based on record attributes by a string that contains the following field codes: `$OutboundRecord.PhoneType$`, `$OutboundRecord.PhoneNumber$,$OutboundRecordField.X$`, where `X` is the name of the custom outbound field. [**Added:** 8.5.143.08]
- display-format.caller-name: Specifies the content of the 'Origin' field of the Case Information area. This option is enabled when the value of the interaction.case-data.content option contains the `History` key. This content is typically used when placing an outbound call where the origin contains a string such as "outbound call to xxx". The content is populated based on attached data keys or contact attributes (if there is a contact) or outbound record fields (if there is an outbound record) that are defined by a string that contains the following field codes: `$Interaction.CaseId$,$Interaction.MainParty$`, `$Contact.X$,$AttachedData.Y$,$OutboundRecord.PhoneType$,$OutboundRecord.PhoneNumber$`, `$OutboundRecordField.Z$`, where `X` is the name of contact attribute, `Y` is the attached data key name, and `Z` is the name of custom outbound field. If the values of the default field codes are empty, the following field code is used: `$Interaction.MainParty$`.

[**Modified:** 8.5.143.08]

Customizing the Wording of an Existing Language

Goal

Replace the wording of an out-of-the-box Workspace string with a string that more closely matches your business context.

Procedure

Use the following steps to create a new language pack for Workspace that has the same name as an existing Workspace language pack.

1. Create a new XML Dictionary file that uses the same naming policy as the out-of-the-box Workspace language files: `custom _name.language_code-country_code.xml`
2. Set the following XML header to specify that it is a custom dictionary to ensure that the definitions in this file override the default strings of this language: `<Dictionary CustomDictionary="true">`
3. Identify the key that you want to customize in one of the original files of the given language and replicate it in your new custom file.

   For example: `<Value Id="State.Agent.Ready" Text="Ready" String="Ready"/>`
4. Modify the text that is contained in the attributes of the key that was copied into the custom file. You can modify the types of the following attributes:
   - Name
   - String
   - Header
   - ToolTip
   - Text
   - Content
   - Title
   - ToolTipStatus
   - Watermark
   - Error
   - ActionButtonToolTip
5. Copy the resulting XML Dictionary file into the `Languages` directory.
6. Restart Workspace.

Notes and Recommendations:

- The only mandatory XML header for a custom file is `CustomDictionary`.
- The following XML headers are optional: `EnglishName`, `CultureName`, and `Culture`. For example: `<Dictionary EnglishName="English" CultureName="English" Culture="en-US" CustomDictionary="true">`.

Potential Errors and Consequences

The table **Workspace Language Errors** displays the most common errors that you might encounter when you create custom language packs.

Workspace Language Errors

Error Message	Consequence
A particular key is not present in any of the files of the new language	The interface will display the value of the default language (en-US).
Malformed XML formatting	It is not possible to switch to the new language.
Translated string is very long	String might appear truncated in the application when the new language is selected.
Keys surrounded by '<!--' and '-->' are translated	Content is not displayed in the interface.
The characters "{0}" are in the original file but are missing in the translation	Some strings are not displayed.

Upgrade Considerations

When a new version of Workspace is released, the default Language Pack might be updated. An update of the Language Pack might consist of the following changes:

- New keys added to an existing dictionary file
- New dictionary file(s) that correspond to new a privilege(s)
- Removal of a key that existed in a previous version of the product
- Change of a string(s) that is associated with a key that existed in previous version of the product
- Movement of a key that existed in previous version of the product from one dictionary file to another

To avoid errors, an upgrade of custom language packs must be planned to ensure that the Workspace User Interface displays correctly in the new version.

Localizing Custom Content in Workspace

Refer to the Workspace Developer's Guide for information about localizing Workspace.