PureConnect®

2023 R2

Generated:

30-May-2023

Content last updated:

29-April-2020

See Change Log for summary of changes.

Secure Input

Technical Reference

Abstract

This document describes how to implement and configure Secure Input, which allows secure submission of sensitive or confidential data such as credit card numbers, for CIC.

For the latest version of this document, see the PureConnect Documentation Library at: http://help.genesys.com/pureconnect.

For copyright and trademark information, see
https://help.genesys.com/pureconnect/desktop/copyright_and_trademark_information.htm.

Table of Contents

Introduction to Secure Input

The *Secure Input Technical Reference* is for technical and management staff who need an overview of CIC Secure Input features; and handler designers who use Secure Input tools in handlers.

Secure Input, which is a component of the Payment Card Industry (PCI) standard, separates and encrypts, or obfuscates, submitted, personal data to protect it from theft or misuse. The PCI Security Standards Council defines the PCI standard as follows "A worldwide information security standard … created to help organizations that process card payments prevent credit card fraud through increased controls around data and its exposure to compromise." The current version of the standard (1.2) specifies 12 requirements for compliance, organized into six logically related groups, called "control objectives." For more information, see http://www.pcisecuritystandards.org.

Example of Secure Input

An external customer makes an inbound call to a contact center and the call routes to an agent. At some point in the interaction, the customer must provide credit card information. The agent clicks **Secure Session** in a CIC client toolbar to start a secure session. At that point, audio between agent and customer separates and the following actions occur:

- The agent hears *on-hold music* to indicate that the agent is still connected to the call. When the call reconnects the agent with the customer, the CIC client alerts the agent and the on-hold music ceases.
- The customer navigates an Interactive Voice Response (IVR) system and provides credit card information. CIC sends the information to a web service within the agent's organization. The web service then sends the information to the credit card issuer for validation. At the end of the IVR session, CIC re-establishes the audio between the customer and the agent. CIC then informs both customer and agent of the outcome (success or failure, with a tracking number or reason for any failure).

System Requirements for Secure Input Feature

The Secure Input feature has the following system requirements:

- CIC
- CIC Secure Input IVR feature license
- Interaction Text-to-Speech license and any applicable languages (required to use the Secure IVR Playback feature of Secure Input, which allows the playback of Dual-Tone Multi-Frequency tones to the external party - CIC 4.0 SU 2 and later)

How CIC Supports Secure Input

CIC provides Interaction Designer tools with which you can implement Secure Input in handlers to assist with PCI compliance. Because of the wide variety of Secure Input needs, CIC does not include a standardized Secure Input feature. Use the handler tools to create Secure Input features that fit the needs of your organization. Configure Secure Input in Interaction Administrator and CIC clients. For more information about configuration, see Secure Input Configuration.

A Secure Input handler starts a separate, secure session inside the customer-agent interaction. It collects the confidential information from the customer, validates it with the appropriate web service, ends the secure session, and returns both customer and agent to the main interaction. Secure Input provides two key security features: Separation and data protection.

Note: PureConnect Secure Input features are not available for conference calls.

Separation

If a customer-agent interaction requires automatic validation of confidential information such as a credit card number, Secure Input separates the customer’s input (audio or text) from other parties in the interaction. That separation protects confidential data from accidental capture (for example, by agents, coaches, and monitors). CIC and its subsystems do not capture and do not record the customer's input during these separations.

For more information about Secure Pause features (needed for PCI compliance) related to call recordings, see "Secure Recording Pause" in the PureConnect Security Precautions Technical Reference. You must have the appropriate logon credentials to view this document.

Data protection

Along with separating user input from the agent and others participating in the interaction, CIC protects the data from access by other users of the system in the following manner:

- **No tracing:** CIC processes involved in a secure session (for example, Telephony Server, Interaction Media Server, and Interaction Recorder) avoid tracing of sensitive information.
- **Minimal access to sensitive information in memory:** Only a minimal set of authorized processes has access to sensitive data. In particular, only TsServerU.exe and ininmediaserverU-W64.exe have access to secure data within CIC. This safeguard lowers the risk of accidental disclosure.

High-level Steps to Implement Secure Input in CIC

Following are the high-level steps to implement and use Secure Input.

1. Implement a web service to receive secure-session information from CIC and transmit it to a payment gateway.
2. In Interaction Designer, create and publish the Secure Input handler.
3. In Interaction Administrator, do the following:
    a. Click the **CIC server** object in the navigation pane and then double-click the **Configuration** item in the right pane.
    b. Click the **Telephony Parameters** tab.
    c. Click **General** in the list box and then select **the Enable Secure Input feature** check box.
    d. In the navigation pane, click the **Secure Input Forms** object under the **People** container.
    e. Right-click the right pane and click **New** from the resulting shortcut menu.
    f. Define a Secure Input form. For more information about creating a Secure Input form, see Define Secure Input Forms in Interaction Administrator.
    g. In the navigation pane, click the **Workgroups** object under the **People** container.
    h. In the right pane, double-click the workgroup for which you want to allow Secure Input.
    i. In the **Configuration** dialog box for that workgroup, click the **Secure Input Forms** tab and add an available Secure Input form to this workgroup.
    j. In the navigation pane, click the **Users** object under the **People** container.
    k. In the **Configuration** dialog box for that user, click the **Security** tab and then click **Security Rights**.
    l. In the **Security Rights** dialog box, ensure that you select the **Secure Input** and **Initiate Secure Input Interactions** check boxes.
4. In a CIC client, right-click the toolbar, click **Customize**, click **Secure Input** to add the option to the toolbar, and then restart the CIC client.

Download the Sample Secure Input Handler

Genesys provides a sample Secure Input handler to demonstrate how to implement Secure Input support. Genesys doesn’t support the use of sample handler in a production environment. Download the sample handler for secure IVR from the CIC Utilities and Downloads page.

Note: To download the sample handler for Secure IVR, you must have the appropriate credentials.

Interaction Supervisor Secure Input Features

In Interaction Supervisor, if the selected interaction has used Secure Input, the **Interaction Details** view shows a Secure Input icon. Next to the icon is the number of times that the interaction has used Secure Input. For more information, see "Interaction Details View" in the Interaction Supervisor Help.

Secure Input Configuration

Complete the following steps to configure Secure Input:

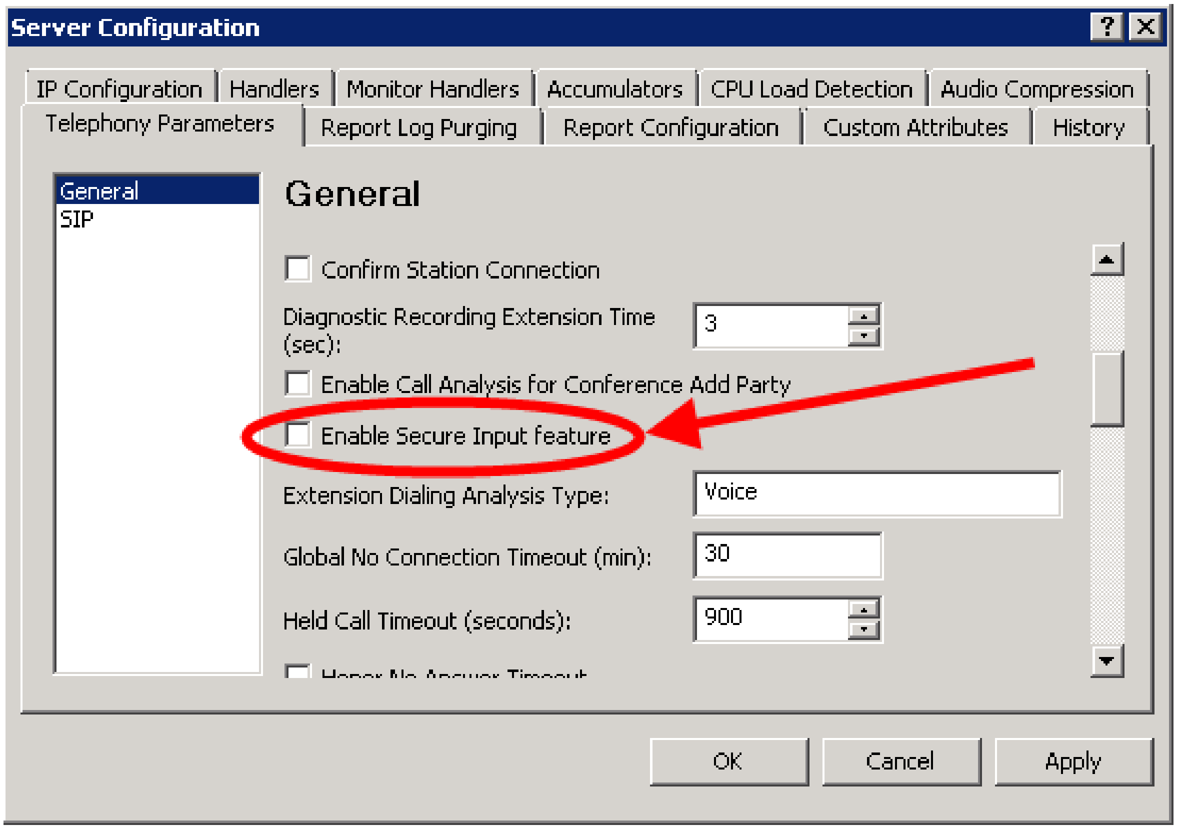
1. Activate Secure Input in Interaction Administrator
2. Define Secure Input Forms in Interaction Administrator
3. Assign Secure Input Form to a Workgroup and Assign Security Rights
4. Add the Secure Input Option to Interaction Desktop
5. Configure Text Filtering for Email Messages and Web Chats (optional)
6. Configure Secure Input to Use HTTPS

Activate Secure Input in Interaction Administrator

Step one for configuring Secure Input is to activate it in Interaction Administrator. For a complete list of the configuration steps, see Secure Input Configuration.

To activate Secure Input in Interaction Administrator

1. Open Interaction Administrator.
2. In the navigation pane of Interaction Administrator, click the **CIC server** object.
3. In the right pane, double-click the Configuration item. The **Server Configuration** dialog box appears.
4. In the **Server Container** dialog box, click the **Telephony Parameters** tab.
5. On the **Telephony Parameter** tab, click the **General** item in the list box on the left side of the dialog box.
6. Select the **Enable Secure Input Feature** check box.



General

□ Confirm Station Connection

Diagnostic Recording Extension Time (sec): 3

□ Enable Call Analysis for Conference Add Party

□ Enable Secure Input feature

Extension Dialing Analysis Type: Voice

Global No Connection Timeout (min): 30

Held Call Timeout (seconds): 900

□ Honor No Answer Timeout

7. Click **OK**. Interaction Administrator activates the Secure Input feature.

Define Secure Input Forms in Interaction Administrator

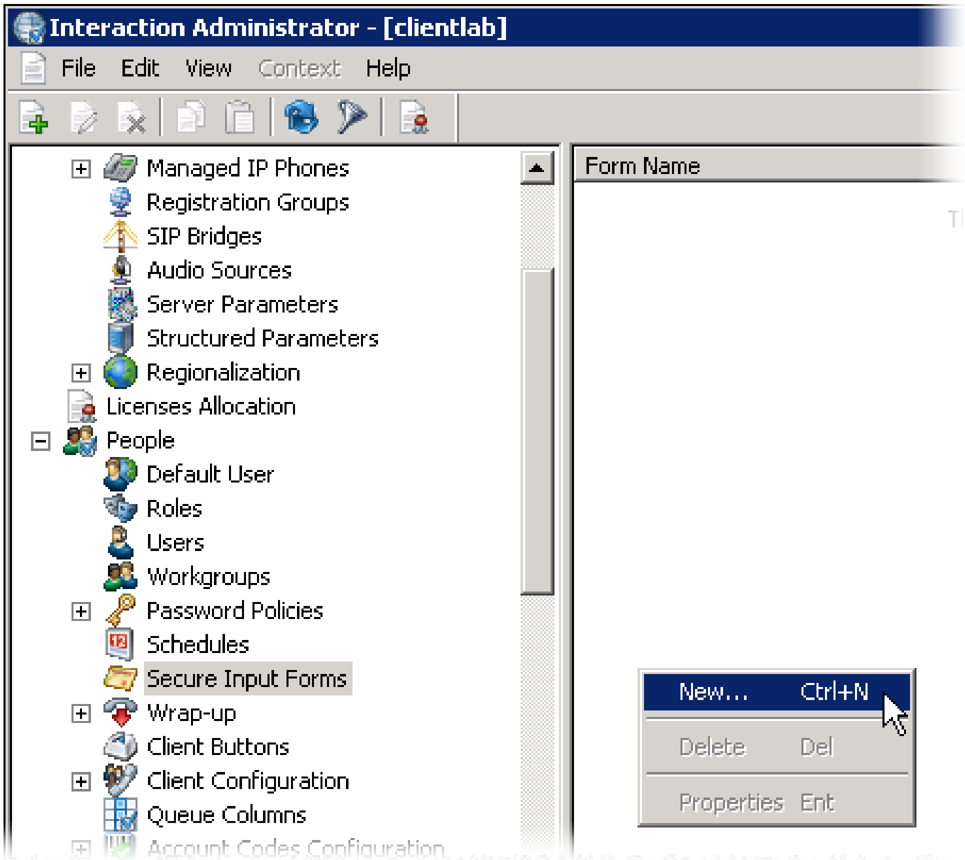
Step two for configuring Secure Input is to define Secure Input forms in Interaction Administrator. CIC displays these forms to the agent when the agent starts a secure session. The agent enters non-confidential information that the customer provided, and then starts the secure session. In the secure session, the customer uses the IVR system to enter confidential information.

> **Note:** Secure Input forms let the agent enter non-secure data (such as customer name and product purchased) in preparation for the secure session.

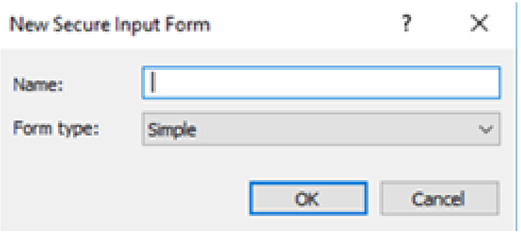
For a complete list of the configuration steps, see Secure Input Configuration.

To define Secure Input forms in Interaction Administrator

1. Open Interaction Administrator.
2. In the navigation pane, click the **Secure Input Forms** object under the **People** container.
3. In the right pane, right-click and then click **New** from the resulting shortcut menu.



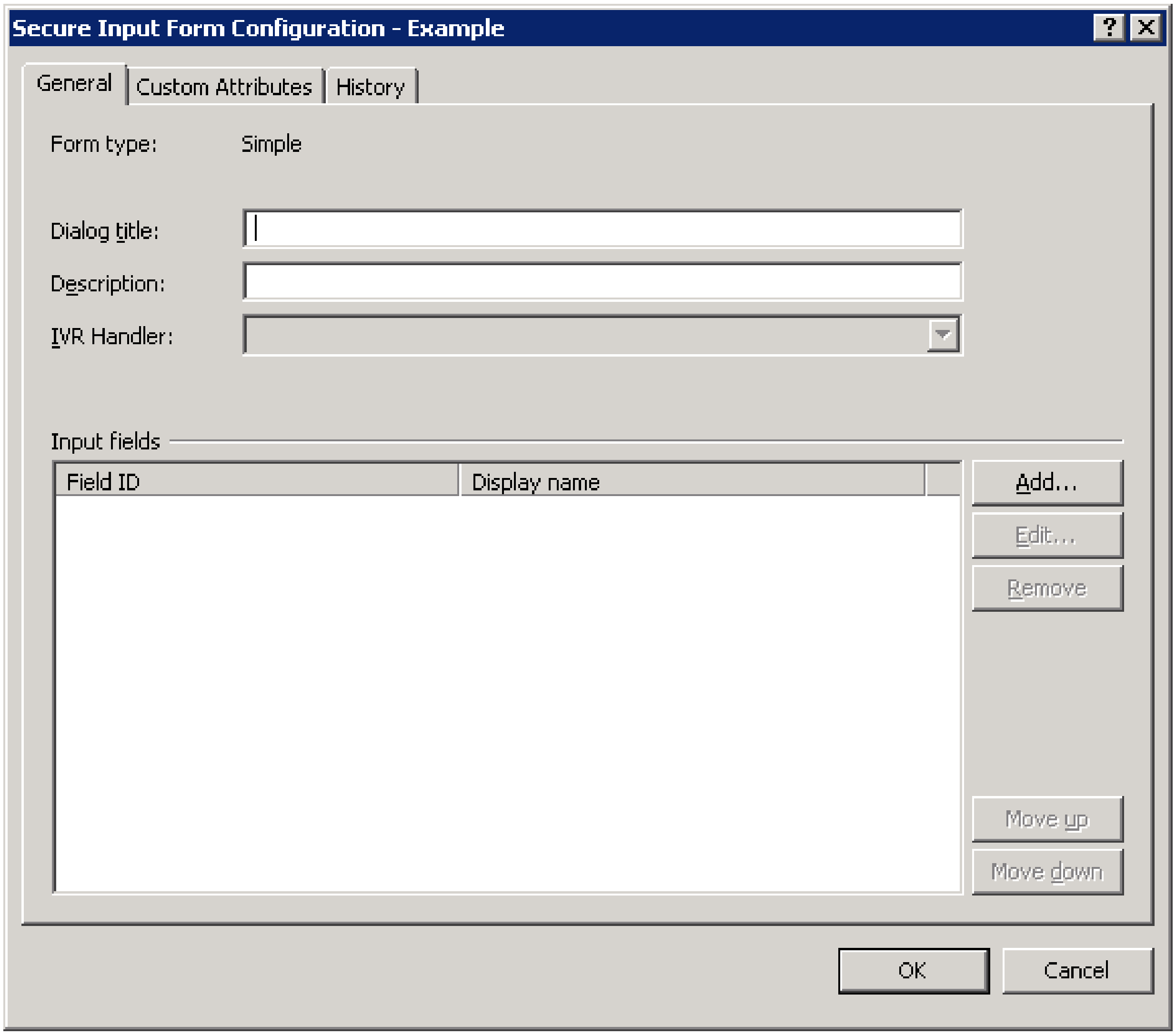
The **New Secure Input Form** dialog box appears.



4. In the **Name** box, enter a name for this **Secure Input Form**.
5. In the **Form type** list box, click one of the following items:
   - **Simple** - Accepts text input from the agent
   - **Custom** - Accepts input of various types with attributes that you specify in an add-on.

> **Note:** Custom forms require you to write an add-on for the CIC client through which you want to use Secure Input. For more information about custom forms, see Appendix: Custom Secure Input forms.

6. Click **OK**. The **Secure Input Form Configuration** dialog box appears.

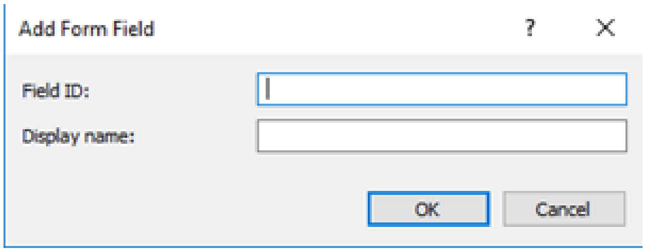


Form type: Simple

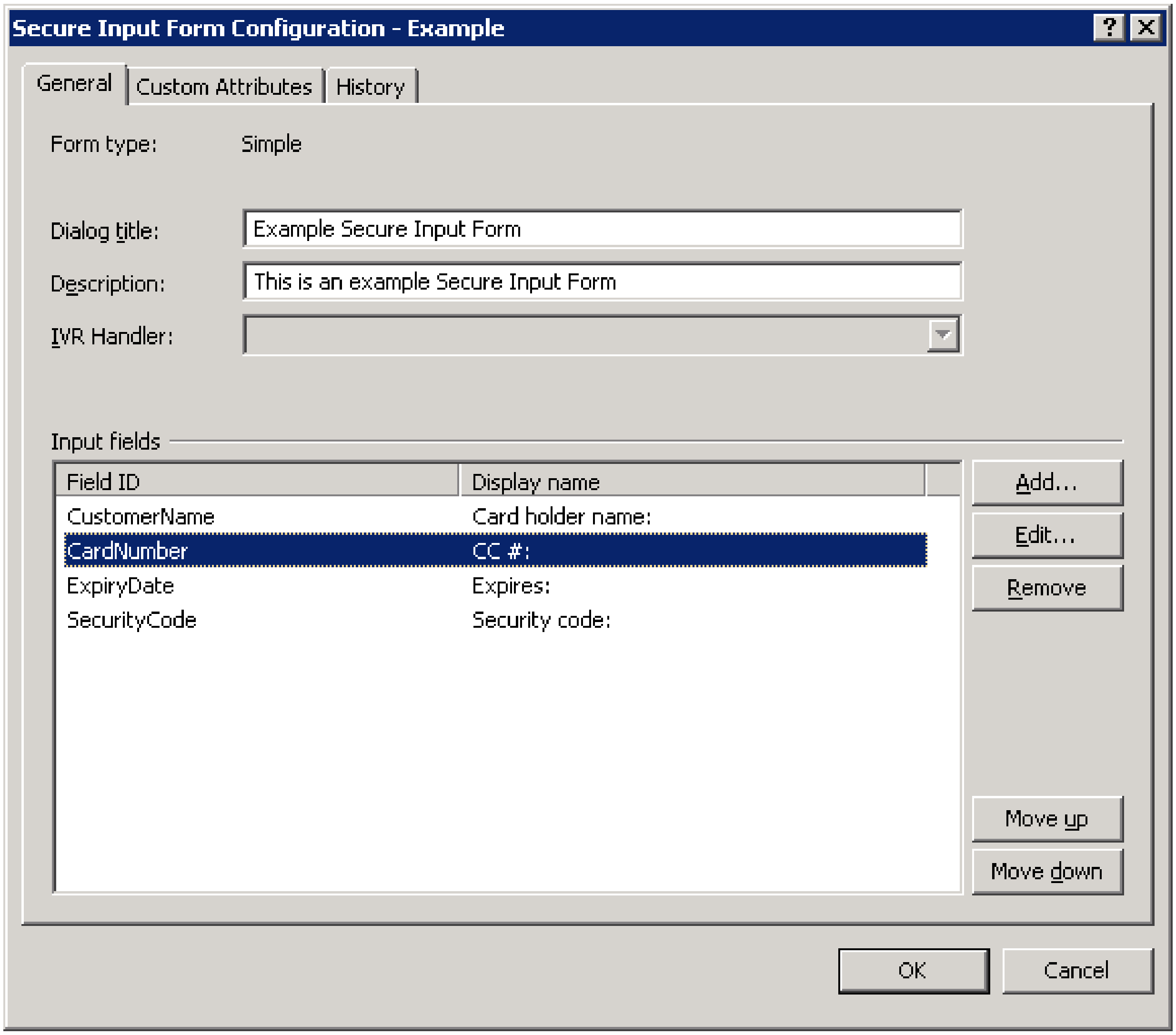
7. On the **General** tab, provide information for the following:
    - **Dialog title** Type the text to display to the agent as the title of the Secure Input form.
    - **Description** - Type the text that describes the form, such as purpose or occasion for usage.CIC clients display this description when the agent selects the Secure Input form.

- **IVR Handler** - Click the handler to use to start the secure session when the agent invokes it. The **IVR Handler** list box displays only those published handlers that start with a **Secure Input** initiator.

8. To add fields to the Secure Input form, click **Add**. The **Add Form Field** dialog box appears.



9. In the **Add Form Field** dialog box, provide the specified information for the following:
    - **Field ID** - Type an identifier that CIC uses for the secure data field. CIC uses the Field ID to identify data items that it sends to the validation service.
    - **Display name** - Type a label to display for the Field ID control of the Secure Input form.
10. Click **OK**. Interaction Administrator adds the field to the new Secure Input form.
11. Continue adding fields as necessary.
12. To edit the configuration, appearance, and order of the fields for this Secure Input form, click **Edit**, **Remove**, **Move Up**, and **Move Down**.



Secure Input Form Configuration - Example

Form type: Simple

Dialog title: Example Secure Input Form

Description: This is an example Secure Input Form

IVR Handler:

Input fields

Field ID	Display name
CustomerName	Card holder name:
CardNumber	CC #:
ExpiryDate	Expires:
SecurityCode	Security code:

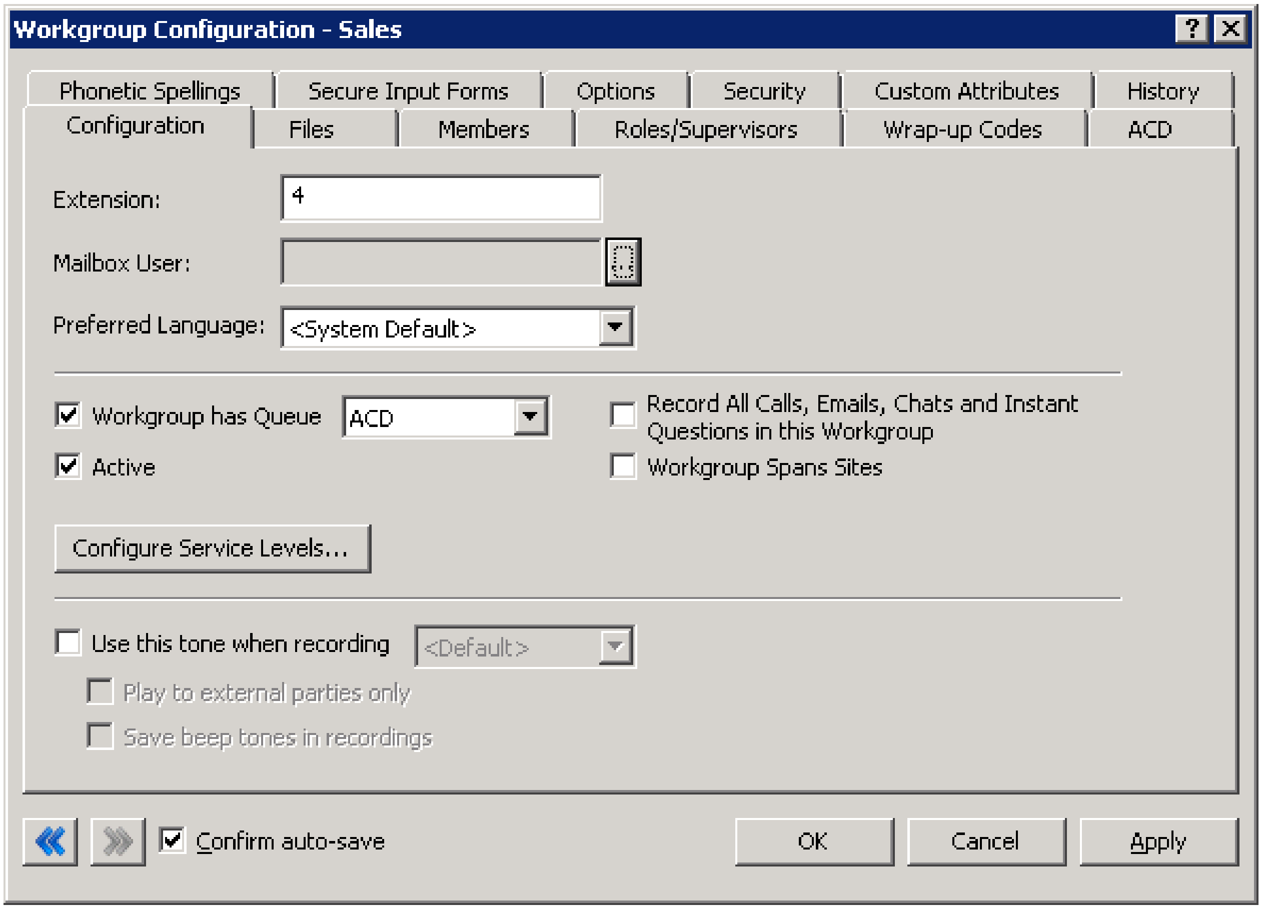
13. To define custom attributes for CIC to use with the Secure Input form, use the **Custom Attributes** tab.
14. To provide any notes for this Secure Input form, use the **History** tab.
15. Click **OK**. Interaction Administrator saves the new Secure Input form and makes it available for interactions.

Assign Secure Input Form to a Workgroup and Assign Security Rights

Step three for configuring Secure Input is to assign a Secure Input form to a workgroup in Interaction Administrator and then assign access rights for that form to members of that workgroup. For a list of the configuration steps, see Secure Input Configuration.

To assign Secure Input forms to a workgroup and assign security rights

1. Open Interaction Administrator.
2. In the navigation pane on the left side of the **Interaction Administrator** window, expand the **People** container and click the **Workgroups** object.
3. In the list of existing workgroups in right pane, double-click the workgroup for which you want to provide access to the Secure Input form. The **Workgroup Configuration** dialog box appears for the selected workgroup.



Extension: 4

Mailbox User:

Preferred Language: <System Default>

☑ Workgroup has Queue ACD

☑ Active

☐ Record All Calls, Emails, Chats and Instant Questions in this Workgroup

☐ Workgroup Spans Sites

☐ Use this tone when recording <Default>

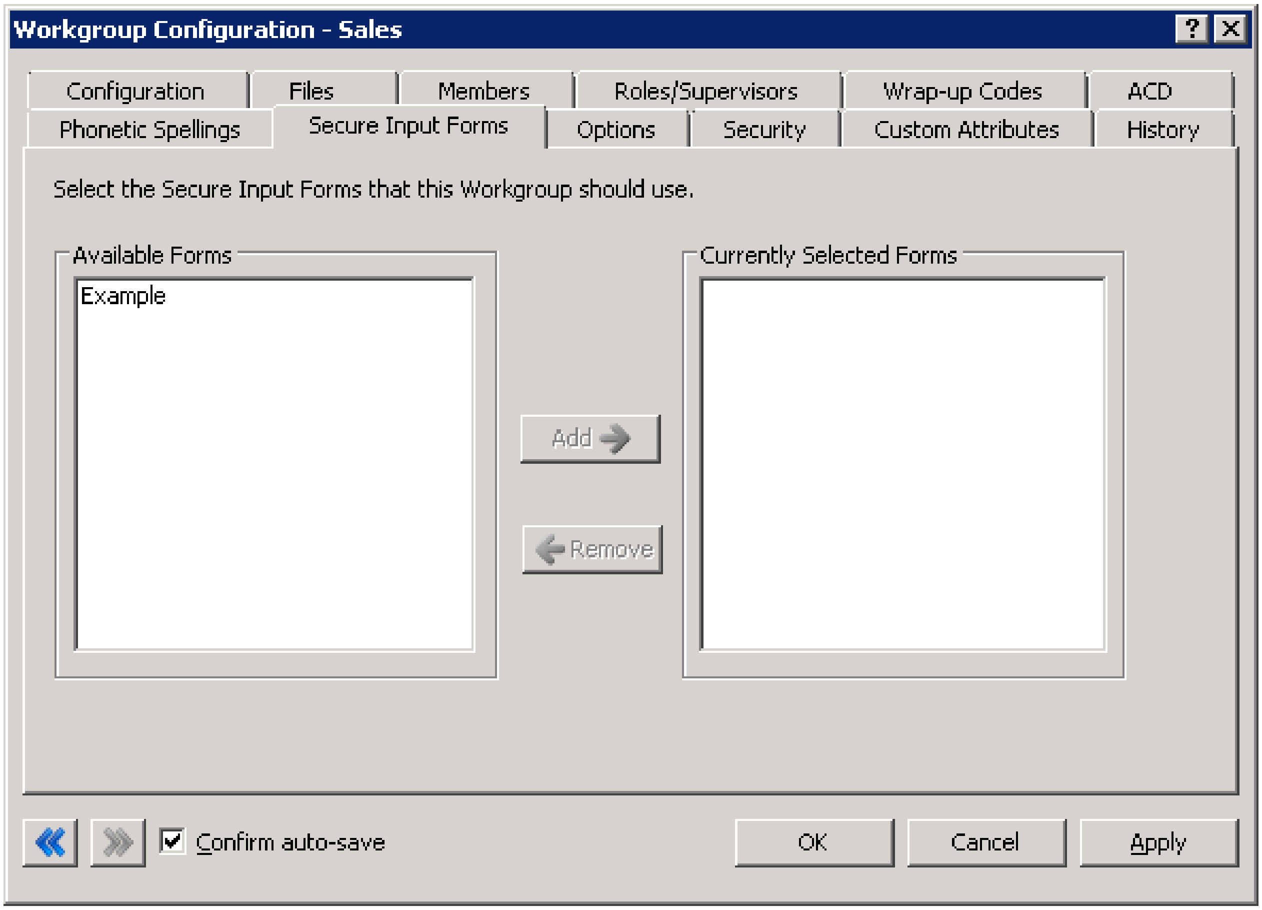
☐ Play to external parties only

☐ Save beep tones in recordings

☑ Confirm auto-save

4. Click the **Secure Input Forms** tab.

> **Note:** If the Workgroup Configuration dialog box does not include the Secure Input Forms tab, activate the Secure Input feature and define at least one Secure Input form.



Select the Secure Input Forms that this Workgroup should use.

Available Forms: Example

☑ Confirm auto-save

5. In the **Available Forms** list box, add one or more Secure Input forms to this workgroup by either double-clicking each form, or selecting multiple forms and then clicking **Add**.

> **Tip:** You can select multiple forms using the following methods:
>
> • Press and hold the **Ctrl** key and click each form.
> • Click a form, press and hold the **Shift** key, and then click the last form in the group to include.
>
> The selected Secure Input forms appear in the **Currently Selected Forms** list box.

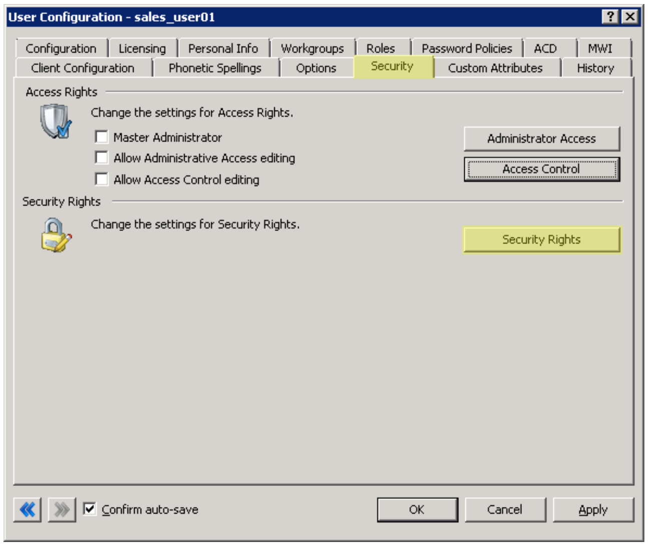
6. Click **OK**. The **Workgroup Configuration** dialog box closes.

> **Important!**
> For users to start a Secure Input session and use the Secure Input form assigned to the workgroup, assign the security rights. You can assign the security rights using the Security Rights option on the Security tab of either the Workgroup Configuration dialog box or the User Configuration dialog box. If you assign the security rights though the Workgroup Configuration dialog box, all users assigned to that workgroup can start a Secure Input session and use the assigned Secure Input form. If you require some, but not all, of the workgroup users to start Secure Input sessions, assign the security rights using the User Configuration dialog box.

7. In the navigation pane of Interaction Administrator, click one of the following objects:

   Workgroups Allows all members of a workgroup can start a Secure Input session and use the assigned Secure Input forms.

   Users Allows a single user to start a Secure Input session and use the assigned Secure Input forms.
8. In the right pane of the Interaction Administrator window, double-click either the workgroup or user for which you want to assign security rights for Secure Input and Secure Input forms. The **Configuration** dialog box appears for the specified entity.
9. Click the **Security** tab.


User Configuration - sales_user01
Configuration | Licensing | Personal Info | Workgroups | Roles | Password Policies | ACD | MWI
Client Configuration | Phonetic Spellings | Options | Security | Custom Attributes | History
Administrator Access
Access Control
Security Rights
OK
Cancel
Apply

Access Rights

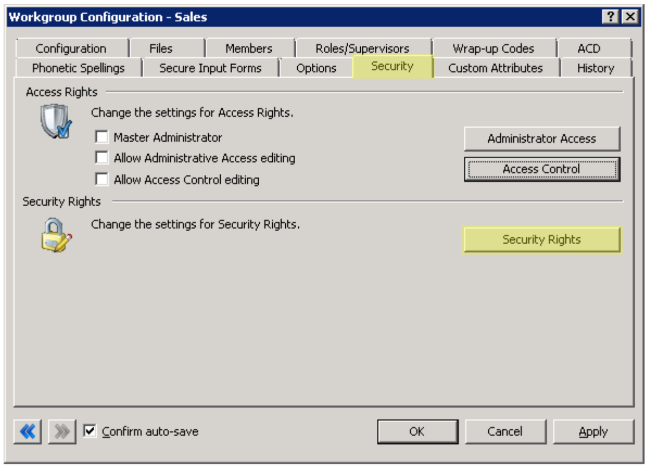
Change the settings for Access Rights.

□ Master Administrator
□ Allow Administrative Access editing
□ Allow Access Control editing

Security Rights

Change the settings for Security Rights.

☑ Confirm auto-save


Workgroup Configuration - Sales
Configuration | Files | Members | Roles/Supervisors | Wrap-up Codes | ACD
Phonetic Spellings | Secure Input Forms | Options | Security | Custom Attributes | History
Administrator Access
Access Control
Security Rights
OK
Cancel
Apply

Access Rights

Change the settings for Access Rights.

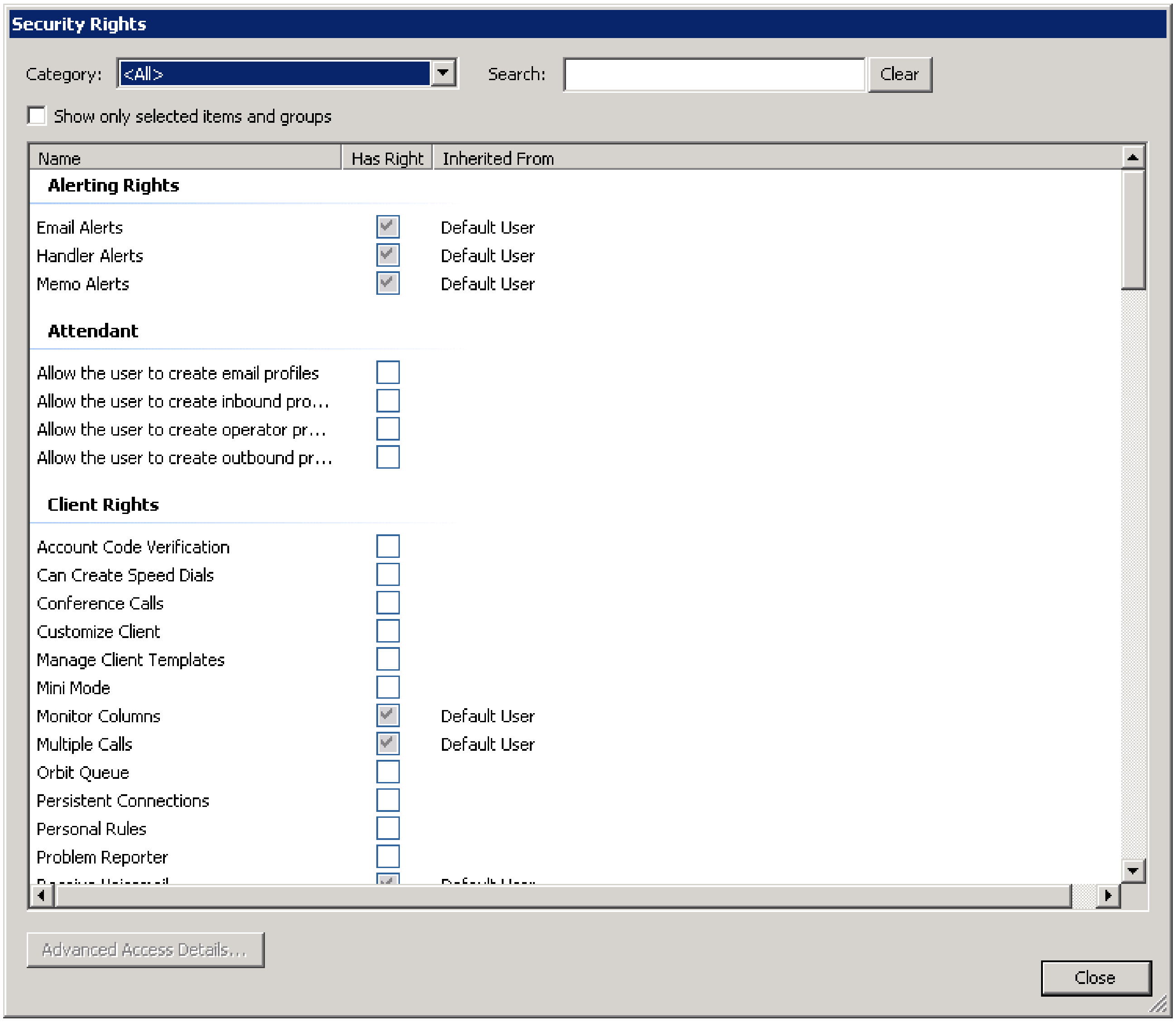
□ Master Administrator
□ Allow Administrative Access editing
□ Allow Access Control editing

Security Rights

Change the settings for Security Rights.

☑ Confirm auto-save

10. On the **Security** tab, click **Security Rights**. The **Security Rights** dialog box appears.



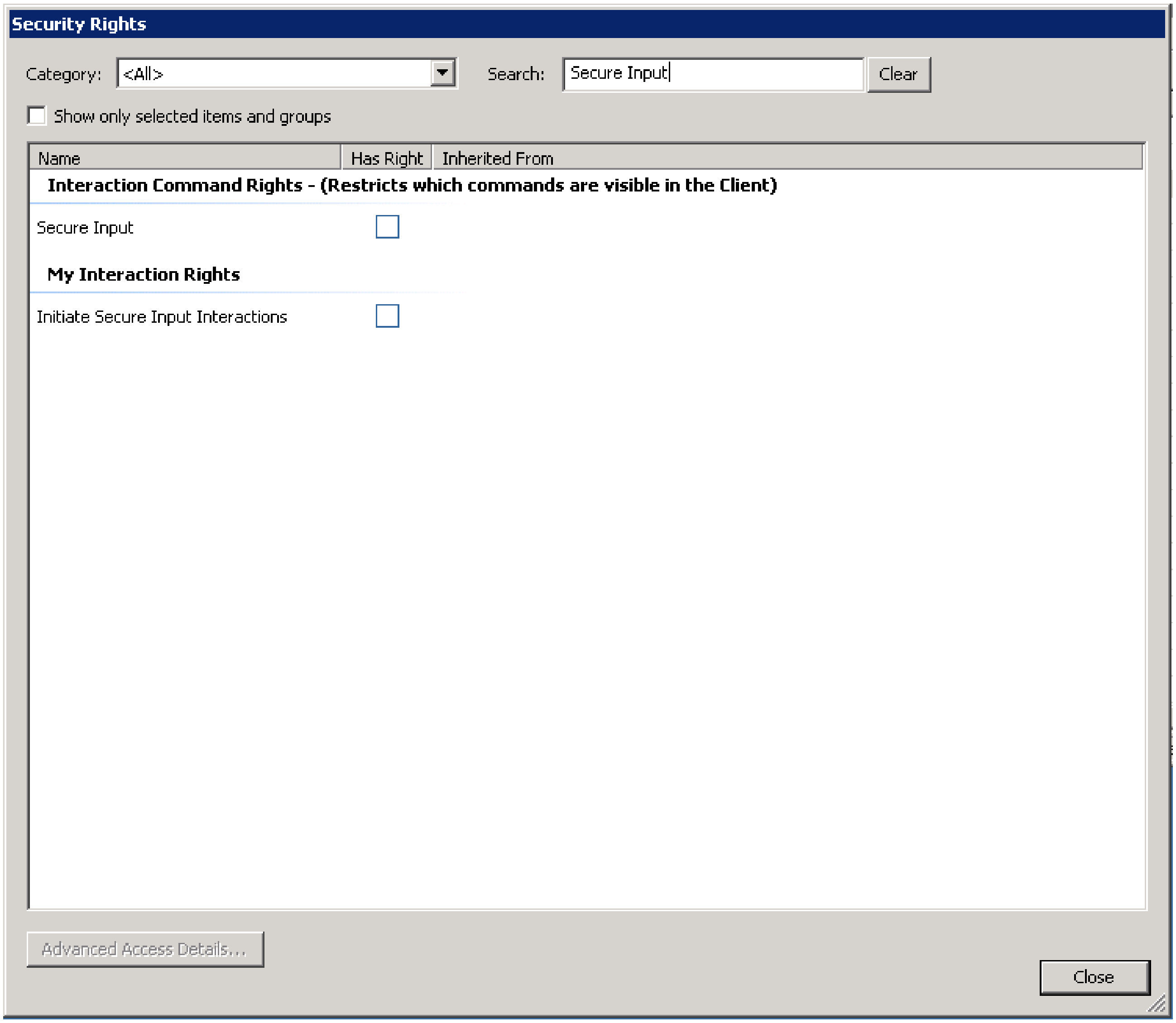
Category: <All>

Search:

□ Show only selected items and groups

Name	Has Right	Inherited From
Alerting Rights		
Email Alerts	☑	Default User
Handler Alerts	☑	Default User
Memo Alerts	☑	Default User
Attendant		
Allow the user to create email profiles	□	
Allow the user to create inbound pro...	□	
Allow the user to create operator pr...	□	
Allow the user to create outbound pr...	□	
Client Rights		
Account Code Verification	□	
Can Create Speed Dials	□	
Conference Calls	□	
Customize Client	□	
Manage Client Templates	□	
Mini Mode	□	
Monitor Columns	☑	Default User
Multiple Calls	☑	Default User
Orbit Queue	□	
Persistent Connections	□	
Personal Rules	□	
Problem Reporter	□	

11. In the **Search** box, type “Secure Input” and wait for Interaction Administrator to filter the list of rights.



Security Rights

Category: <All> Search: Secure Input

Clear

□ Show only selected items and groups

Name	Has Right	Inherited From
Interaction Command Rights - (Restricts which commands are visible in the Client)		
Secure Input	□	
My Interaction Rights		
Initiate Secure Input Interactions	□	

12. Select the **Secure Input** and **Initiate Secure Input Interactions** check boxes.
13. Click **Close**.
14. In the **Configuration** dialog box for the selected entity, click **OK**. The Interaction Administrator main window appears.

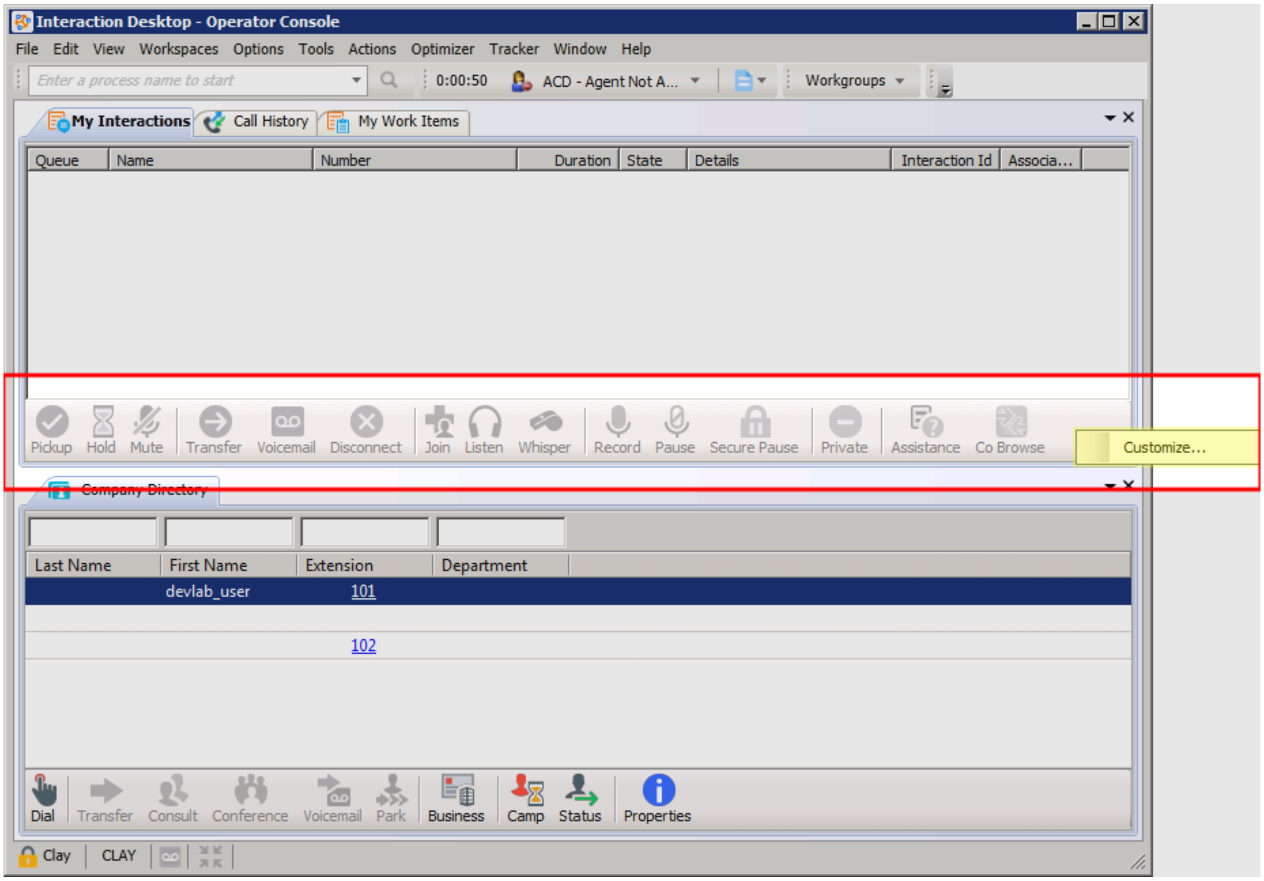
Add the Secure Input Option to Interaction Desktop

Step four for configuring Secure Input is to add the Secure Input option to Interaction Designer. For a list of the configuration steps, see Secure Input Configuration.

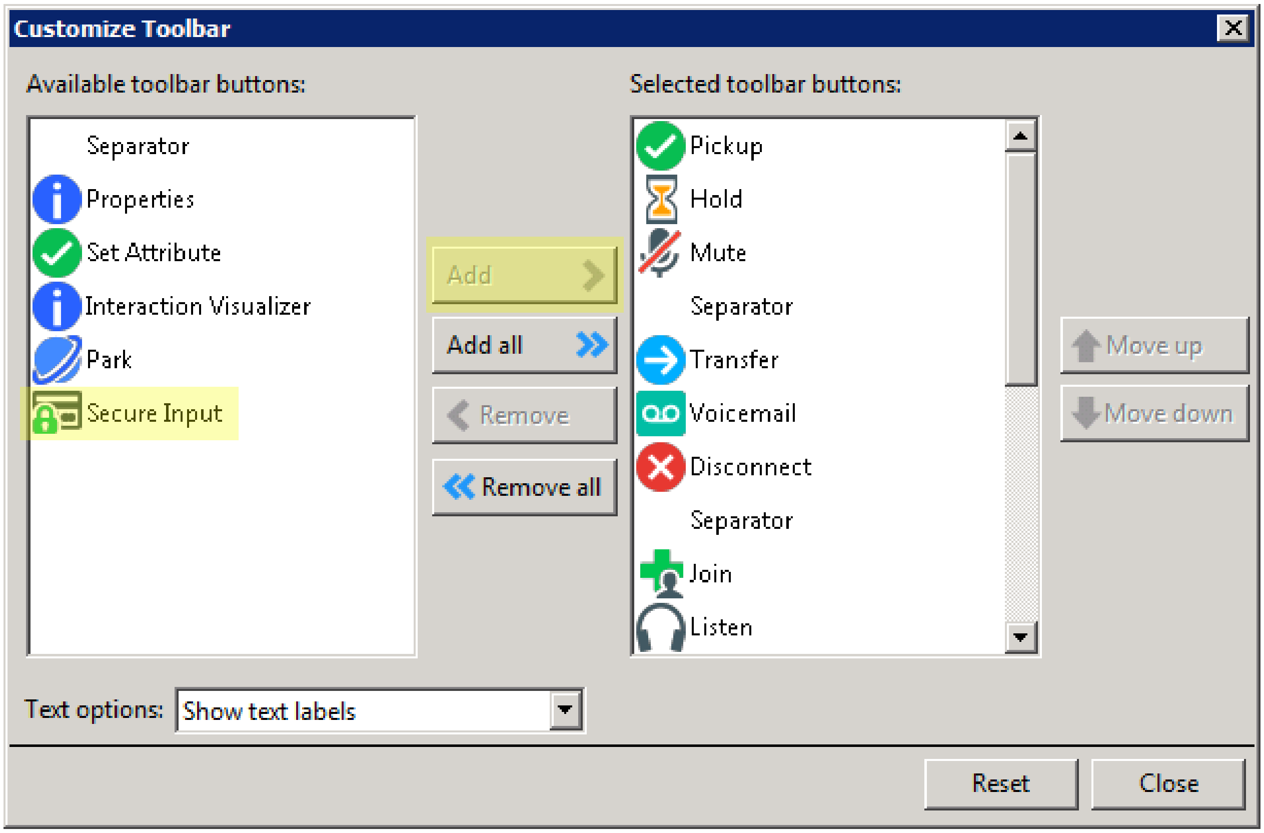
> **Note:** The Secure Input option is available only if you activated the Secure Input feature in Interaction Administrator, defined a Secure Input form, assigned it to a workgroup, and assigned the associated security rights for either workgroups or users.

To add the Secure Input option to Interaction Desktop

1. Open Interaction Desktop.



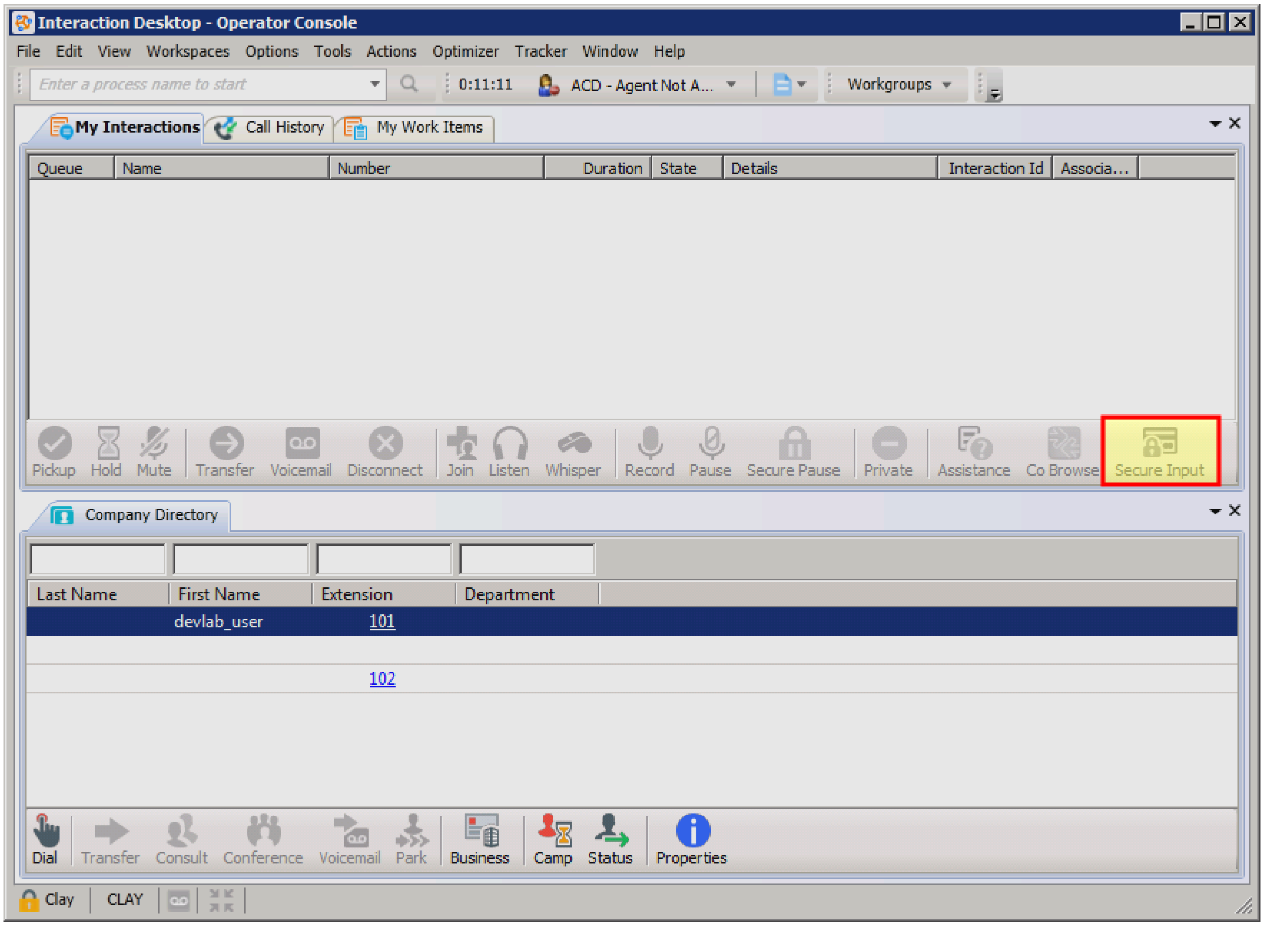
2. Click **Customize...**. The **Customize Toolbar** dialog box appears.



3. In the **Available toolbar buttons** list box of the **Customize Toolbar** dialog box, click the **Secure Input** item and then click **Add**.

Tip: You can change the position of the Secure Input option by clicking Move up and Move down next to the Selected toolbar buttons list box.

4. Click **Close**. The **Secure Input** option appears in the interaction toolbar.



Queue	Name	Number	Duration	State	Details	Interaction Id	Associa...

Last Name	First Name	Extension	Department
	devlab_user	101	
		102	

Configure Text Filtering for Email Messages and Web Chats

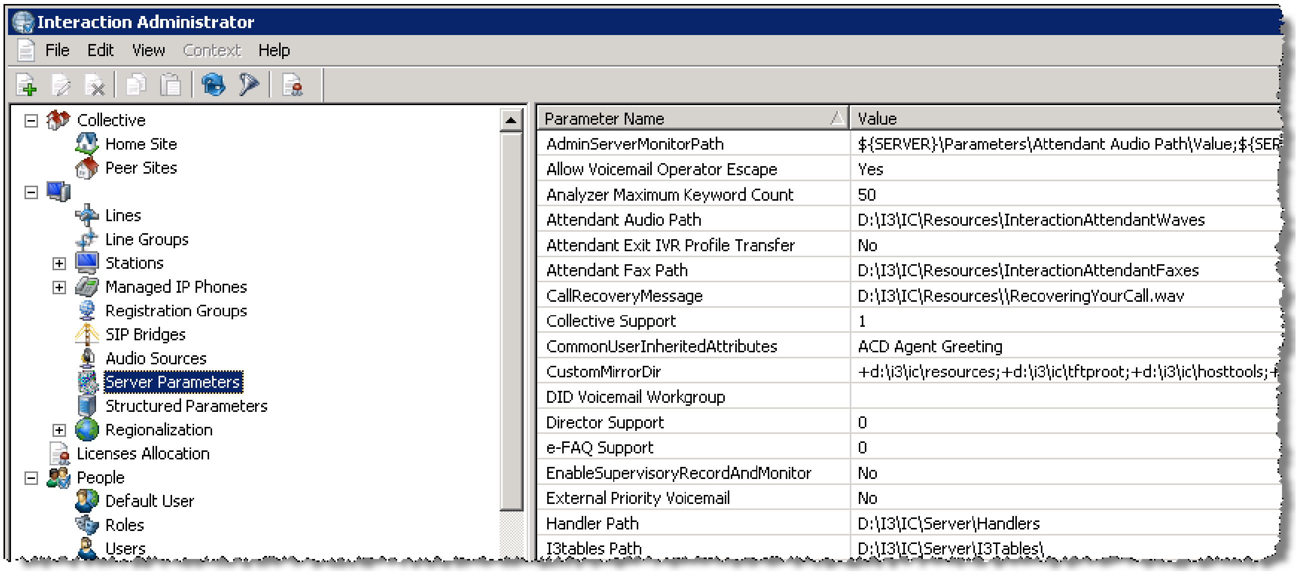
Step five for configuring Secure Input is to configure text filtering for email messages and web chats in Interaction Administrator. This step is optional, and allows you to filter patterns of private information from inbound email messages and web chats so that the system doesn't log the information or display it to agents.

> **Important!**
> Text filtering for inbound email messages requires usage of the Microsoft Exchange Web Services (EWS) connector. Genesys doesn't support text filtering for other mail connectors.

For a list of the configuration steps, see Secure Input Configuration.

To configure text filtering for email messages and web chats

1. Open Interaction Administrator.
2. In the navigation pane of the Interaction Administrator window, click the **Server Parameters** object.



Parameter Name	Value
AdminServerMonitorPath	${SERVER}\Parameters\Attendant Audio Path\Value;${SER
Allow Voicemail Operator Escape	Yes
Analyzer Maximum Keyword Count	50
Attendant Audio Path	D:\I3\IC\Resources\InteractionAttendantWaves
Attendant Exit IVR Profile Transfer	No
Attendant Fax Path	D:\I3\IC\Resources\InteractionAttendantFaxes
CallRecoveryMessage	D:\I3\IC\Resources\\RecoveringYourCall.wav
Collective Support	1
CommonUserInheritedAttributes	ACD Agent Greeting
CustomMirrorDir	+d:\i3\ic\resources;+d:\i3\ic\tftproot;+d:\i3\ic\hosttools;+
DID Voicemail Workgroup	
Director Support	0
e-FAQ Support	0
EnableSupervisoryRecordAndMonitor	No
External Priority Voicemail	No
Handler Path	D:\I3\IC\Server\Handlers
I3tables Path	D:\I3\IC\Server\I3Tables\

3. In the right pane, right-click anywhere and then click **New** from the resulting shortcut menu. The **Entry Name** dialog box appears.
4. In the **Enter Parameter Name** text box of the **Entry Name** dialog box, type “Redact Using Default Expression” and then press **Enter**. The **Parameter Configuration** dialog box appears.
5. On the **Configuration** tab of the **Parameter Configuration** dialog box, type “Yes” in the **Parameter Value** text box and then click **OK**. This server parameter replaces most credit card numbers with the # character using a regular expression.

Create more text filters

To add more regular expressions to prevent the display of personal information in inbound email messages and web chats, do the following steps:

1. Add the “Additional Redaction Expression” server parameter. The **Parameter Configuration - Additional Redaction Expression** dialog box appears.

2. On the **Configuration** tab in the **Parameter Value** text box, type a regular expression to apply to email or web chat text. CIC uses this regular expression and the default regular expression to replace matching character strings with # characters.

   Example regular expression for filtering United States Social Security Numbers:

   ```
   ^(?!000)([0-6]\d{2}|7([0-6]\d|7[012]))([ -]?)(?!00)\d\d\3(?!0000)\d{4}$
   ```

   > **Tip:** For more information about regular expressions, search the Internet for tutorials, references, examples, and testing tools.

   > **Note:** To disable the default regular expression for filtering text, set the Parameter Value of the Redact Using Default Expression server parameter to **No**. If you added the Additional Redaction Expression server parameter, CIC continues to apply its regular expression in that server parameter for filtering text.

Configure Secure Input to Use HTTPS

Step six for configuring Secure Input is to configure Secure Input to use HTTPS. Secure Input doesn't use typical Secure Sockets Layer (SSL) to establish an encrypted link between a web server and a browser. To connect securely to a web service that requires a certificate, configure Secure Input to use HTTPS.

For a list of the configuration steps, see Secure Input Configuration.

To configure Secure Input to use HTTPS

1. Create keys in DS at "\CustomerSite\Production\SERVERNAME\" with the object path of "HttpFactories" and class of "HttpFactories."
2. Under that entry, create a key called "default" using the specified object path and class "HttpFactory.".
3. To the default key, add the "security_ctx.ca_dir" attribute as a string type and specify the location of the folder that contains the certificate.
4. Restart the TsServer subsystem.

> **Important!**
> For a valid switchover pair, restarting the TsServer subsystem causes an immediate switchover. For a stand-alone IC server, restarting the TsServer subsystem renders the TsServer inoperable for up to two minutes while the subsystem restarts.

5. Retrieve and convert the certificate.

   Secure Input requires certificates in a specific format. "CertTrustU.exe" to convert certificates to the required format. Beginning with CIC 2016 R3, "CertTrustU.exe" installs with CIC in the "I3\IC\Server" folder. Always use the version of "CertTrustU.exe" that corresponds to your current CIC version. If you are on a CIC version earlier than 2016 R3, contact PureConnect Customer Care for the required version of "CertTrustU.exe." For more information about "CertTrustU.exe" and its options, see the *Security Features Technical Reference* in the PureConnect Documentation Library.

   Before running "CertTrustU.exe," verify whether the "I3\IC\Certificates\SOAP" folder exists and if it doesn't, create it. "CertTrustU.exe" doesn't create the folder for you.

> **Note:** For a switchover pair, create the folder on both IC servers.

6. From the command line where "CertTrustU.exe" resides, run "CertTrustU.exe -s -y <address> <port>".

Switch	Description
-s	Saves the files in the "I3\IC\Certificates\SOAP" folder.
-y	Answers "yes" to all prompts, rather than waiting on user input.
<address>	URL of the service to which to connect.
<port>	Port number that the service uses. The default value for HTTP is 80 and the default value for HTTPS is 443.
/?	Displays other available tool options.

   For example, if the web service address is "https://www.mysoapservice.com/myService." the command is:

```
CertTrustU.exe -c -y www.mysoapservice.com 443
```

7. Place the generated certificate in the "I3\IC\Certificate\SecureSession" folder. If this folder doesn't exist, create it.

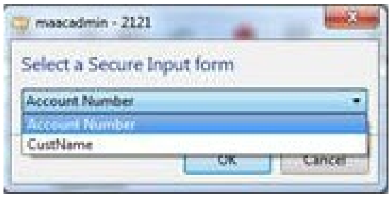
Start a Secure Input Session in Interaction Desktop

When an agent has a connected call and the customer is ready to give confidential information, the agent starts a secure session. The secure session collects data from the customer through a secure, non-recorded channel, and then returns control to the agent to complete the interaction.

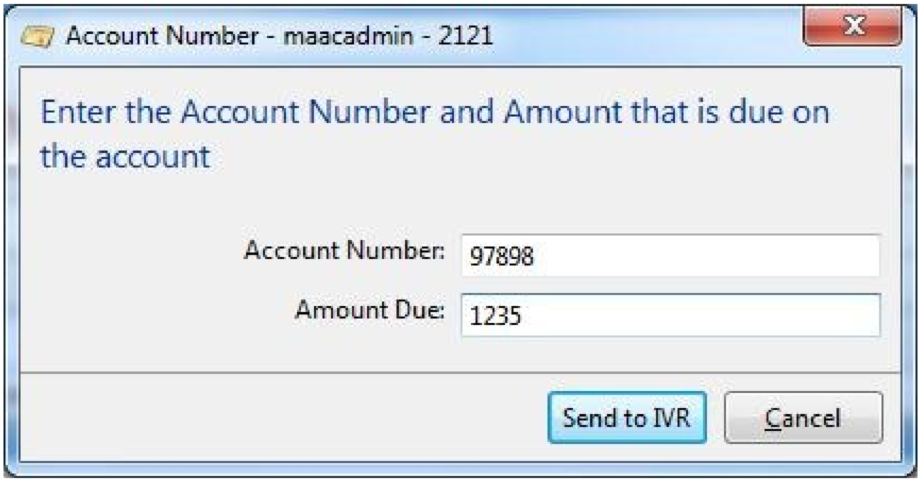
> **Important!**
> This procedure requires you to activate and configure Secure Input. For more information, see Secure Input Configuration.

To start a Secure Input session in Interaction Desktop

1. In Interaction Desktop in the toolbar, click **Secure Input**. A dialog box appears that allows the agent to select a Secure Input form assigned to the workgroup.



2. In the list box, click the Secure Input form and then click **OK**. Interaction Desktop displays a dialog box in which the agent provides non-confidential information to send to the Secure Input IVR session, such as the customer's account number and amount due. The Secure Input IVR session associates this information with the information that the caller provides during the private session. The following dialog box is an example Secure Input form that prompts for the non-confidential information to send to the Secure Input IVR session.

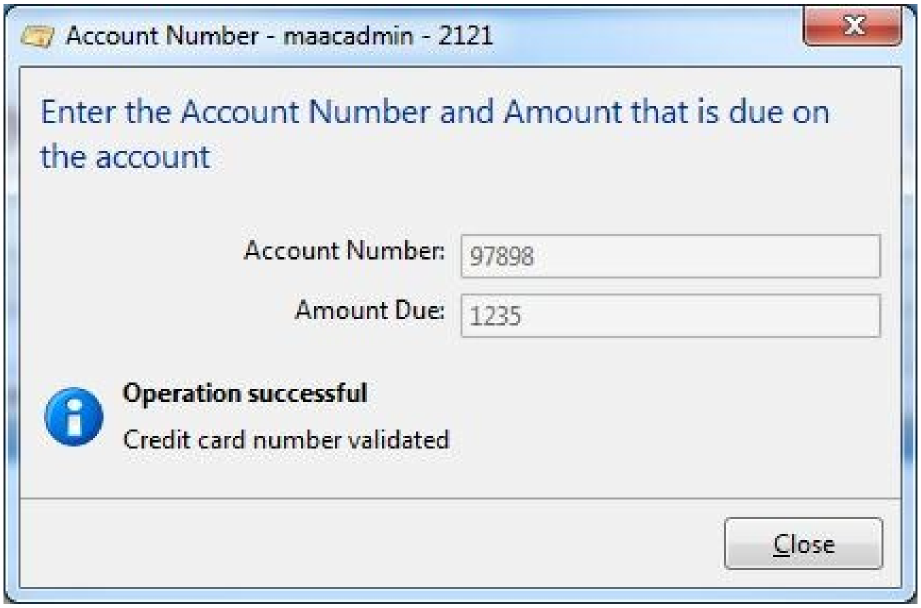


Enter the Account Number and Amount that is due on the account

Account Number: 97898

Amount Due: 1235

3. After you provide the non-confidential customer information, click **Send to IVR**. The system redirects the caller into a Secure Input IVR session where the caller provides confidential information. Interaction Desktop displays **Waiting for caller to complete input …** to the agent. When the caller completes the confidential information, the call reconnects to the agent and displays a result message that indicates the validity of the information.
4. If the confidential information is valid, click **Close**. Otherwise, click **Retry** or **Cancel**.



Enter the Account Number and Amount that is due on the account

Account Number: 97898

Amount Due: 1235

Operation successful

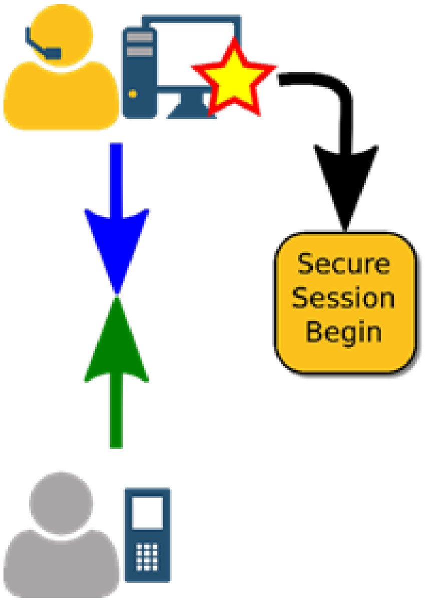
Credit card number validated

Interaction Designer Tools for Secure Input

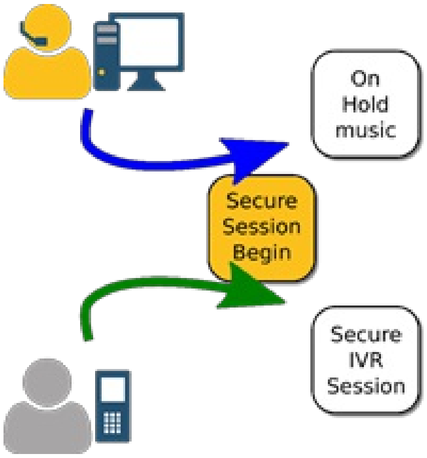
Secure Input is available through CIC handlers, which are customizable flows and extensions for the behavior and functionality of your CIC contact center. To create or modify handlers, use Interaction Designer. For more information, see the *Interaction Designer Help* at https://help.genesys.com/cic/mergedProjects/wh_id/desktop/hid_introduction.htm.

Secure Session Begin

The Secure Session Begin handler tool is the first to use for Secure Input sessions as it has the appropriate functionality in the CIC Telephony Services Application Programming Interface (API). It is the Secure Session Begin handler tool that starts when the agent clicks **Secure Input** in the Interaction Desktop application.

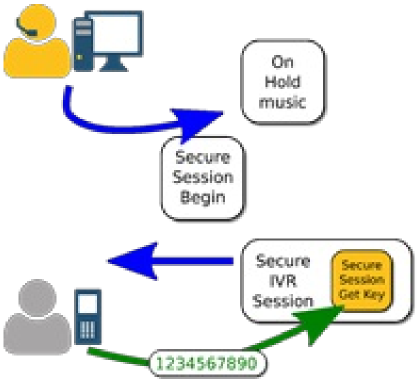


During this process, the system places the agent who started the Secure Input session on hold and transfers the caller to the Secure IVR session.



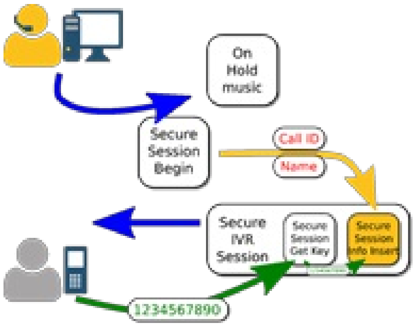
Secure Session Get Key

In the Secure IVR session, the IVR prompts the caller to provide the confidential information, such as a credit card number. The Secure Session Get Key handler tool collects the information that the caller provides, such as digits. The handler tool supports DTMF only, and not ASR.



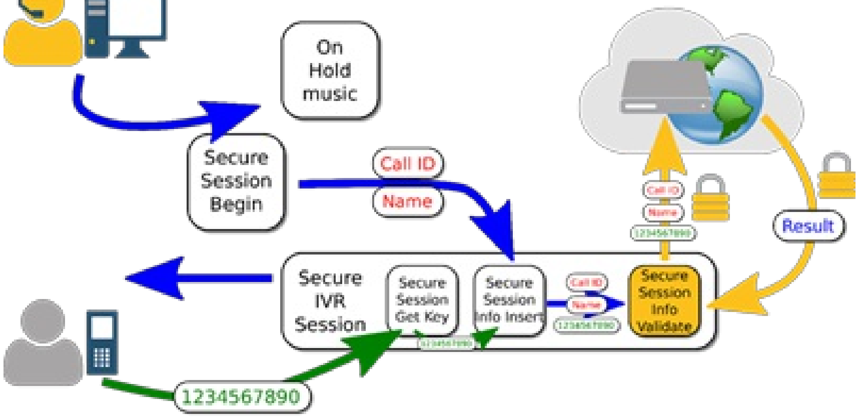
Secure Session Info Insert

The Secure Session Info Insert tool attaches non-confidential information, such as a call ID or name, with the confidential information that the caller provided. This combination of information is for use during the validation process.



Secure Session Info Validate

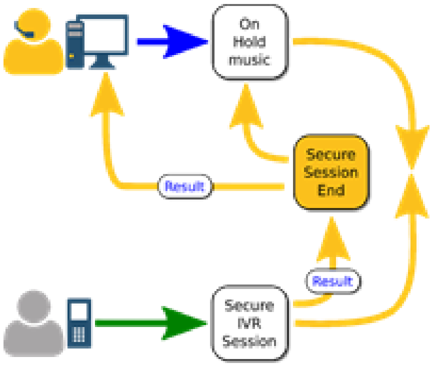
With the combination of the confidential caller information and the non-confidential information from CIC, the Secure Session Info Validate tool submits the information to a web service for validation.



The web service returns a result for failure or success.

Secure Session End

The final handler tool, Secure Session End, provides only the result of the confidential information validation check to the Secure Input form that started the Secure Input session. The caller and agent are again able to communicate.



Interface between the handler and Interaction Desktop

The handler uses the call attribute EIC_SECURE_IVR_RESULT to communicate the results of the secure session to Interaction Desktop.

Syntax

transactionId|*ApprovalValue*|*ApprovalResult* \n

Variables

- *transactionId* is the output of the secure session initiator.
- *ApprovalValue* and *ApprovalResult* are outputs from the secure session info validate tool. If an ApprovalResult contains the | character, remove all instances of that character or replace them with some other value before adding them to the call attribute.
- \n is a newline character.

Interface between CIC and the web service

When the handler calls the Secure Session Info Validate tool, Telephony Services uses the POST method to send the gathered data to the validation web service at the URL specified in the input.

Submission format

```
<?xml version="1.0" encoding='UTF-8' ?>
<params>
<param name="customer name" value="Joe Smith" />
<param name="credit card" value="4012888888881881" />
<param name="customer account" value="343453422" />
<param name=”charge amount” value=”14.55” />
</params>
```

Expected reply format

```
<?xml version="1.0" encoding='UTF-8'?>
<outcome value="accepted” details=”That number was good"/>
```

The reply format requires the following XML declaration:

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

Important!
CIC supports accepted or declined for the value attribute of the outcome element.
The details attribute of the outcome element cannot contain the | (pipe) character because the system reserves it for Interaction Desktop to use to recognize message syntax.

Appendix: Custom Secure Input forms

You can implement custom Secure Input forms using Interaction Desktop Add-in, which allows you to create add-ins including Secure Input forms that you can display in Interaction Desktop. For more information, see the Interaction Desktop Add-in Technical Reference.

The API provides the following interfaces: ISecureInput and ISecureInputForm.

To create a custom Secure Input form

1. Write code that implements the ISecureInput interface.
2. Register that implementation with the ISecureInputService.

Interaction Desktop uses the IsecureInputService to find and display the selected form when the agent clicks **Secure Input** in the toolbar.

For more information, see Example API for Custom Secure Input Forms.

Example API for Custom Secure Input Forms

This API allows you to define the following classes needed for Secure Input:

- **SecureInputAddin.cs:** Defines the Interaction Desktop add-on and implements the IAddIn interface. For more information, see SecureInputAddin class.
- **CustomSecureInput.cs:** Implements the ISecureInput interface. For more information, see CustomSecureInput class.
- **MyForm.cs:** Provides content for the custom Secure Input form (partially implemented). For more information, see MyForm class.

```
namespace ININ.InteractionClient.AddIn

{

/// <summary>

/// Provides access to a custom secure input. Implement this

/// interface and add an instance of the implementation

/// to the <see cref="ISecureInputService"/> to make the custom

/// secure input available.

/// </summary>

/// <remarks>

/// The user selects secure input. If the user selects a custom secure input,

/// the Interaction Desktop uses the matching named form which was added to the

/// <see cref="ISecureInputService"/>.

/// Parameters configured by an administrator dictate what information is passed

/// into the <see cref="GetForm"/> method. In addition, interaction attributes

/// (if available) specified by the <see cref="AdditionalAttributes"/> property is

/// included in the parameter dictionary given to the <see cref="GetForm"/> method.

/// </remarks>

///

public interface ISecureInput

{

/// <summary>

/// Gets the name of this secure input. The name is used to find the secure input and activate
it.

/// </summary>

/// <value>The name of the secure input.</value>

string Name { get; }
```

```
/// <summary>
/// Gets interaction attributes this secure input needs. Gathers these attributes
/// and includes them in the parameter passed to the <see cref="GetForm"/> method
/// in addition to server-defined parameters.
/// </summary>
/// <value>The interaction attributes to retrieve.</value>
IEnumerable<string> AdditionalAttributes { get; }
/// <summary>
/// Gets the secure input form to display.
/// </summary>
/// <param name="parameters">The specified Interaction and secure input
/// parameters.</param>
ISecureInputForm GetForm(IDictionary<string, string> parameters);
}
}
namespace ININ.InteractionClient.AddIn
{
/// <summary>
/// A secure input form is a control (Windows Forms or WPF) that is embedded
/// into a containing window and shown to the user. It collects information
/// before sending a caller into a secure IVR session.
/// </summary>
public interface ISecureInputForm
{
/// <summary>
/// Gets a value indicating the state of the form.
/// </summary>
/// <returns><see langword="true"/> if the data is valid and the user may proceed,
/// <see langword="false"/> if the user is not allowed to proceed with the
/// secure input form.</returns>
bool IsValid { get; }
/// <summary>
/// Occurs when the <see cref="IsValid"/> property is changed.
/// </summary>
event EventHandler IsValidChanged;
/// <summary>
/// Gets a dictionary with the name/value pairs provided to the secure IVR,
/// and to the third-party service processing the secure input session.
/// </summary>
IDictionary<string, string> SecureParameters { get; }
/// <summary>
/// Gets the main content to be displayed in the secure input window displayed to the user.
/// </summary>
/// <returns>
/// Return either a Windows Forms Control or a Windows Presentation Foundation Control.
/// Ignores any other return type.
/// </returns>
object Content { get; }
}
}
```

test

SecureInputAddin class

The SecureInputAddin class defines the Interaction Desktop add-on and implements the IAddIn interface.

When Interaction Desktop loads the add-on, the plug-in architecture calls the Load method. The CustomSecureInputForm class uses the IServiceProvider from the Load method, which provides the following actions:

- Retrieves the ISecureInputService
- Registers a custom implementation of the ISecureInput interface with the ISecureInputService. In this case, the custom implementation is an instance of the CustomSecureInput class.

Interaction Desktop later uses ISecureInputService to access any of these custom Secure Input form implementations.

```
namespace CustomSecureInputForm
{
public class SecureInputAddin: IAddIn
{
public void Load(IServiceProvider serviceProvider)
{
var secureInputService = serviceProvider.GetService(typeof(ISecureInputService)) as
ISecureInputService;
if (secureInputService == null) return;
secureInputService.Add(new CustomSecureInput());
}
public void Unload()
{
}
}
}
```

CustomSecureInput class

The CustomSecureInput class implements the ISecureInput interface.

The Name property defines the name of the Secure Input form and must match the form name used in Interaction Administrator. Use the AdditionalAttributes property to specify any other interaction attributes that the custom Secure Input form requires.

In the final section of code, Interaction Desktop calls the GetForm method when the agent clicks **Secure Input** and clicks the Secure Input form to use. Interaction Desktop gets an IDictionary of key-value pairs corresponding to the custom parameters defined for the custom Secure Input form in Interaction Administrator. The example defines a database connection string in Interaction Administrator as one of a form's **Custom** parameters. The parameter is named database_connection_string.

```
namespace CustomSecureInputForm
{
public class CustomSecureInput: ISecureInput
{
public string Name
{
get
{
return "CustomCreditCardProcessing";
}
}
public IEnumerable<string> AdditionalAttributes
{
get { return new string[0]; }
}
public ISecureInputForm GetForm(IDictionary<string, string> parameters)
{
var formProvider = new MyForm { ConnectionString =
parameters["database_connection_string"] };
return formProvider;
}
}
}
```

MyForm class

This partial implementation of the ISecureInputForm interface uses a WinForms UserControl that provides the content for a custom Secure Input form.

The code defines a mock form that pretends to access a database or web service to retrieve values with which to populate two form fields. Clicking Load causes the following actions:

- Data loads into two labels on the form
- SecureParameters dictionary updates with the new values
- IsValid property changes to true

Those actions inform Interaction Desktop that prerequisite user input or activity is complete and that it can start the Secure IVR session.

```
namespace CustomSecureInputForm
{
public partial class MyForm : UserControl, ISecureInputForm
{
```

```
private bool _isValid;
public MyForm()
{
InitializeComponent();
IsValid = false;
SecureParameters = new Dictionary<string,string>(StringComparer.OrdinalIgnoreCase);
}
public object Content
{
get
{
return this;
}
}
public bool IsValid
{
get
{
return _isValid;
}
private set
{
if (_isValid == value) return;
_isValid = value;
var evt = IsValidChanged;
if (evt != null)
{
evt(this, EventArgs.Empty);
}
}
}
public event EventHandler IsValidChanged;
public string ConnectionString { get; set; }
public IDictionary<string, string> SecureParameters { get; private set; }
private void btnLoad_Click(object sender, EventArgs e)
{
// Hit a web service here, or something, to retrieve the values for the specified customer ID
string amount = "$ 142.12";
string address = String.Format("7601 Interactive Way{0}Indianapolis, IN{0}46278",
Environment.NewLine);
lblAmount.Text = amount;
lblAddress.Text = address;
SecureParameters["amount"] = amount;
SecureParameters["address"] = address;
IsValid = true;
}
}
}
```

Change Log

The following table lists the changes to the *Secure Input Technical Reference* since its initial release.

Date	Change
02-August-2012	Initial release of this document.
14-March-2013	Updated information about sample handler download from Product Information site.
12-April-2013	Added note to Chapter 1 that Secure Input features are not available for conference calls.
13-March-2014	• Added a step in Chapter 2 about server parameters for text filtering. • Updated copyright and trademark.
29-August-2014	Updated documentation to reflect changes required in the transition from version 4.0 SU# to CIC 2015 R1, such as updates to product version numbers, system requirements, installation procedures, references to Genesys Product Information site URLs, and copyright and trademark information.
12-August-2015	• Rewrote content to conform to the switch of Interaction Desktop as the agent client for starting Secure Input sessions • Recreated diagrams for Secure Input handler tools to illustrate functionality more accurately • Rebranded documentation for new corporate logo and color scheme • Edited for clarity, style, topic hierarchy, product naming, and other considerations
29-July-2016	• Updated copyright and trademark information • Added content to reflect that text filtering also applies to email messages that use Microsoft Exchange Web Services (EWS) connector. For more information, see Configure Text Filtering for Email Messages and Web Chats (optional).
02-August-2017	• Added to "Interface between CIC and the web service" section under "Expected reply format" to indicate that reply format requires the XML declaration.
16-August-2017	• Rebranded to Genesys. • Added "Configure Secure Input to Use HTTPS."
09-April-2018	Updated document format.
16-November-2018	Updated *Enable HTTPS support* topic with corrections from Customer Care.
30-April-2019	Reorganized the content only, which included combining some topics and deleting others that just had an introductory sentence such as, "In this section...".
25-September-2019	Added clarification that the Secure Session Get Key handler tool doesn't support ASR input.
16-March-2020	Changed "ininmediaserverU.exe" to "ininmediaserverU-W64.exe" because Genesys only ships the 64-bit Media Server executable.
01-April-2020	Changed path to *Security Precautions Technical Reference* in How CIC Supports Secure Input.
29-April-2020	Update or remove links to "my.inin.com" as appropriate.