GENESYS™

Interaction Scripter .NET Client

Printed help

PureConnect powered by Customer Interaction Center® (CIC)

2018 R3

Last updated May 11, 2018

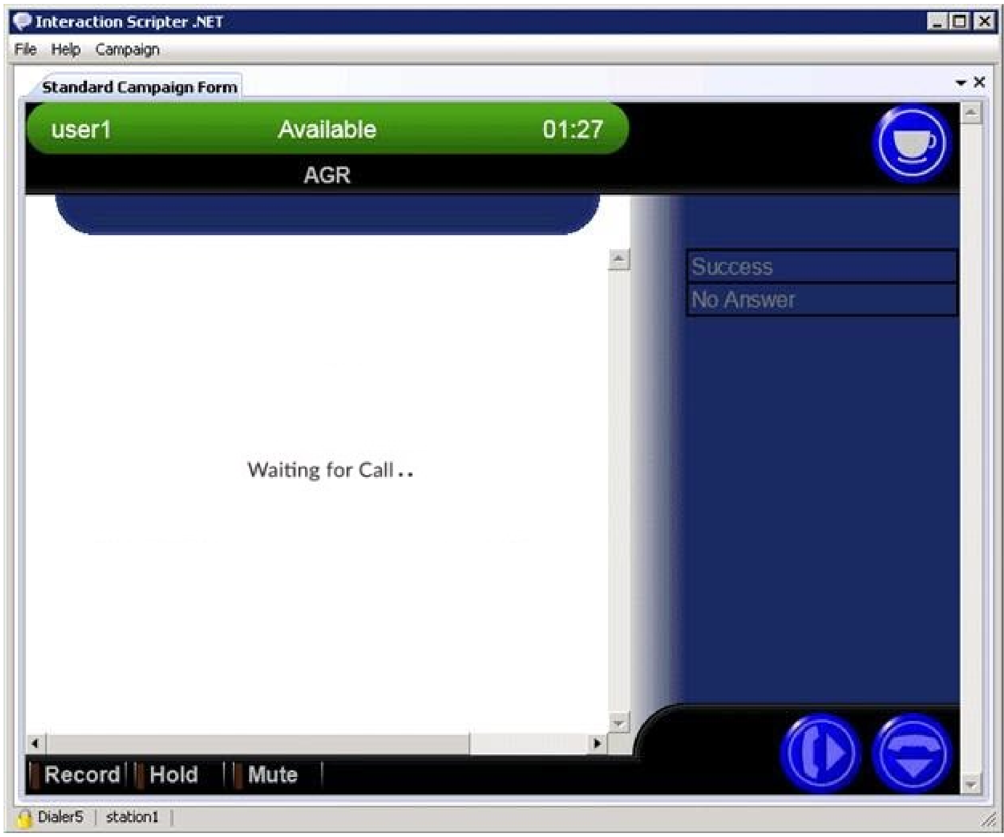
Abstract

This documentation is for call center agents who use the Interaction Scripter .NET client to disposition campaign telephone calls. Scripter .NET populates the agent's display with information pertaining to the call, the customer, and the campaign, based on behavior defined in custom campaign scripts or standard campaign forms.

Table of Contents

Interaction Scripter .NET Client Help

This publication is for contact center agents who process outbound calls placed by Interaction Dialer. It explains how to log in and interact with campaign scripts using **Interaction Scripter .NET Client**. As an alternative to *Interaction Scripter .NET Client*, you can use *Scripter in Interaction Connect* instead. For information about that application, see *Interaction Connect Help.*



Interaction Scripter .NET Client is sometimes called the *Dialer client*, since it is bundled with Interaction Dialer. It is also called *Scripter Client.* A *campaign* is a collection of properties that tell Interaction Dialer how to process the database table that contains telephone numbers and other information. Every campaign has a script that "pops" on screen when a call is sent to you. Scripts define what you see on the screen during a call. Interaction Scripter .NET Client is the application that runs scripts.

Interaction Scripter .NET Client populates your display with information pertaining to the call, the customer, and the campaign, based on behavior defined in a script. All information that you collect or modify during the call is routed back to database tables. Scripts can be simple or highly elaborate, depending upon customer needs. Scripts are written in HTML and JavaScript—the markup languages of the Internet. Campaign scripts run in Interaction Scripter Client. You cannot execute scripts in a web browser, but navigating a campaign script is as easy as surfing the Internet. To interact with a script, you use simple point-and-click mouse movements, and enter information using your keyboard.

Last revision: May 11,2018

Script Types

There are two types of campaign scripts: *custom scripts* and *base scripts* (also known as standard forms). Interaction Scripter .NET Client runs either type of script interchangeably. You are likely to encounter either type, depending upon the level of scripting that your company has licensed, and the type of script that the current campaign is using.

Terminology Note—when the client displays a base script, it is called the **Dialer Client**. When it displays a custom script, it is called the **Scripter Client**. These terms are not really significant outside of Marketing departments, because the same client application is always used. To avoid confusion, the term **Interaction Scripter .NET Client** is used in documentation.

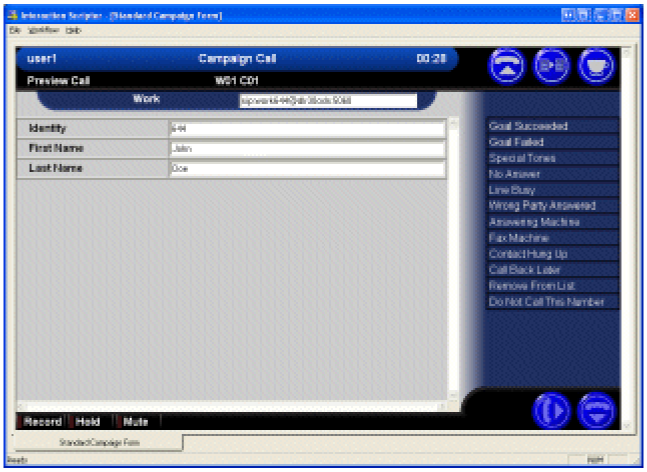
When a base script pops on your screen, it is ready to be completed by you. If the contact is already on the line when the script pops, you can easily see what information needs to be collected. The script may indicate what to say. If information is presented before a call is placed, you can preview the contact record before placing a call to the contact, or you can skip to the next contact.

Base Scripts

Custom Scripts

Base Scripts (Standard forms)

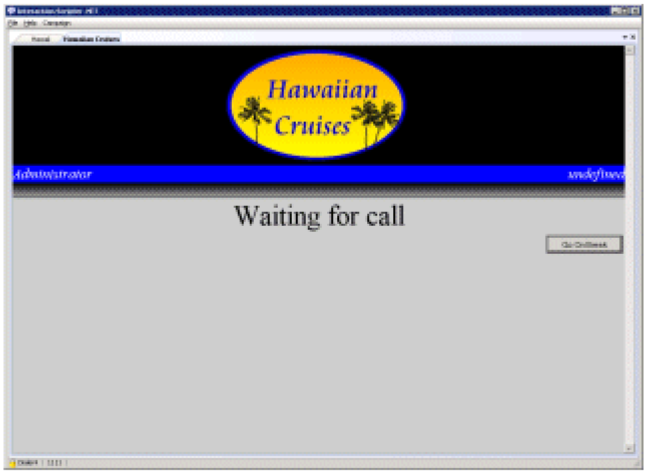
Some contact centers don't wish to rely on programmers. These call centers use base scripts to conduct campaigns. Base scripts are also called *standard forms*. Both terms are interchangeable, and refer to the on-screen forms that your Call Center Administrator created to help you process a call.



Identity	644
First Name	John
Last Name	Doe

Custom Scripts

If your call center licensed Advanced Interaction Scripter, each campaign can have a unique user interface. Custom campaign scripts accommodate any conceivable design requirement, but must be created by web page developers.



Waiting for call

NOTE: Starting with 2018 R3, Interaction Connect supports both base scripts and custom scripts. Due to differences in the way that Scripter .NET and Interaction Connect execute JavaScript, custom scripts must be programmed to target one client or the other. See Interaction Scripter Developer's Guide for details.

Agent Types

Interaction Scripter .NET Client supports two types of call center agents—*Regular Agents* and *Finishing Agents*.

Note: Base scripts do not support Finishing Agents, because base scripts do not provide any mechanism to transfer a call to a finishing agent. Custom campaign scripts can support finishing agents, by calling the Scripter action named IS_Action_MarkCallForFinishing() to transfer the call. See the Interaction Scripter Developer's Guide for details.

Regular Agents

Finishing Agents

Regular Agents

Regular Agents receive new outbound calls. These agents pitch a product, idea, or service, or perform some other front-end process. Once this up-front work is complete, the campaign's custom script transfer calls to an ACD queue for follow-up processing by Finishing Agents. A custom campaign script must perform the actual call transfer.

Finishing Agents

Finishing Agents receive agent-owned callbacks and calls transferred from regular agents, via custom scripts. Finishing agents perform a back-end process, such as validation of credit cards, surveys, or data entry—after a regular agent has finished speaking with the targeted party. Finishing agents receive screen pops exactly like regular agents. The difference is that finishing agents do not receive regular campaign calls.

Related Topics

Work as a Finishing Agent

Work as a Finishing Agent

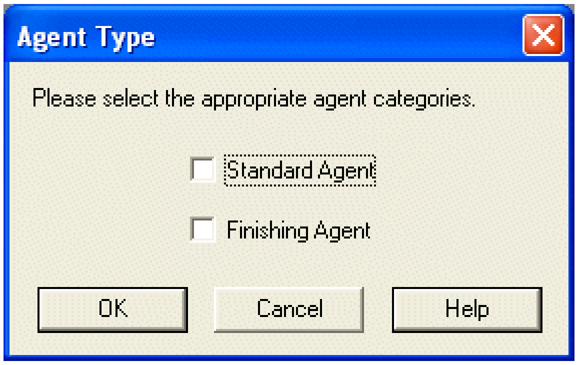
As mentioned in the discussion of Agent Types, Interaction Scripter supports two types of call center agents—Regular Agents and Finishing Agents. Finishing Agents are a second group of agents who complete calls for Regular Agents, by performing a back-end process, such as validation of credit cards, surveys, or data entry—after the regular agent has finished speaking with the contact.

When Finishing Agents participate in a campaign, its custom script transfers calls from Regular Agents to a group of Finishing Agents who complete calls. The campaign must use a custom campaign script to transfer calls to an ACD queue for follow-up processing by Finishing Agents, since base scripts do not provide support for Finishing Agents.

When you work as a Finishing Agent, you will receive screen pops exactly like regular agents, and you are expected to complete the call by assigning reason/finish codes. The main difference is that finishing agents do not receive regular first-line calls.

To work as a finishing agent, follow these steps:

1. Add **/agent=menu** to the shortcut used to start Interaction Scripter. This startup parameter causes a menu to appear at startup, so that you can identify yourself as a Finishing Agent, a Regular Agent, or both. See Set Agent Type at Startup for more details.



Please select the appropriate agent categories.

□ Standard Agent

□ Finishing Agent

2. When a call is delivered to you as a Finishing Agent, interact with the script to collect information from the contact, place an order, or take other action necessary to disposition the call.

   If no Finishing Agents are logged in when a Regular Agent transfers a call, the contact is placed on hold until an agent becomes available or the contact hangs-up. In the contact hangs up, the call is considered an abandoned call. More sophisticated interactions are also possible, if the call is routed to an Attendant Profile. In that scenario, Attendant could play audio or even route the call back to a regular agent (manipulating the categories again) after an elapsed period of time.

Optional Command Line Parameters

You can start Interaction Scripter Client using a command file or shortcut that runs InteractionScripter.NET.exe with optional command line parameters. These parameters allow you to access special features or automate your connection operation.

These parameters may be delimited using a forward slash (/) or a dash (-).

The syntax is:

```
InteractionScripter.NET.exe [parameters]
```

The following optional command line parameters are available:

Parameter	Description
Agent	Identifies the type of agent you want to log on as.
Autoload	Automatically loads the specified HTML page(s) at startup.
Autologin	Bypasses the logon dialog box and automatically log on if a switchover occurs.
Debug	Debugs custom scripts while in development.
ForceScriptUrl	Forces Scripter to switch to the new active campaign when a campaign transition occurs-regardless of the ReadyState status.
Help	Displays a complete list of all the available command line parameters.
InitialStatusKey	Sets status before log on.
Initiate	Places an initial call to the agent in order to establish a connection up front.
LogNameSuffix	Specifies a custom suffix to the application log files.
LogoutDialogOnClose	Configures Scripter to prompt the agent to log out before the application can be closed.
NoConfirmClose	Prevents Scripter to from prompting you for confirmation before exiting the application.
NoStartReceiving	Allows an agent to log on to Scripter, set their status to available, but not have Dialer place calls for them until the IS_Action_StartReceivingCalls is called.

Parameter	Description
Notifier	Specifies the name of the Dialer Server to which you want to connect.
Password	Specifies a CIC user's password on the command line.
Persistent	Specifies that a remote connection should be persistent.
Remote	Logs on from a remote workstation or remote number.
Size	Specifies a particular size for the Scripter window.
Standalone	Disables the Dialer-related functions in scripts.
User	Specifies a CIC user name on the command line.
Workstation	Specifies the name of the workstation defined in Interaction Administrator to which the user is logging on.

Agent

You can use the Agent parameter to identify the type of agent you want to log on as. If the /agent parameter is omitted, the agent is considered to be a regular agent by default. For details and examples, see Set Agent Type at Startup.

Syntax

```
/agent= <[menu];[standard];[finish | finishing];[all];[ro |
readonly];[noprompt]>
```

Options

You can use more than one option on a command line; just be sure and separate each option with a semicolon (;).

menu

> Presents the Agent menu from which the user can select which type of agent they want to log on as.

standard

> Logs the agent on as a standard (opener) agent.

finish | finishing

> Logs the agent on as a finishing agent.

all

Logs the agent on as a standard agent (opener) and a finish agent.

ro | readonly

Displays the agent type menu in Scripter as readonly, user cannot change type.

noprompt

Tells Scripter client not to show the agent type selection dialog during log on.

Examples

To log on as a standard agent:

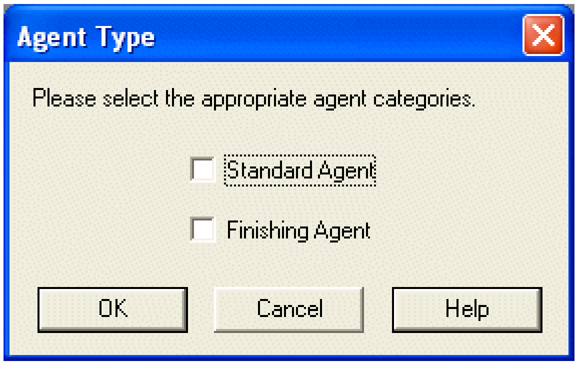
```
InteractionScripter.NET.exe /notifier=myDialerServer /user=craigd
/password=1234 /workstation=Station1 /agent=standard
```

To log on as a finishing agent:

```
InteractionScripter.NET.exe /notifier=myDialerServer /user=craigd
/password=1234 /workstation=Station1 /agent=finishing
```

To log on using the menu:

```
InteractionScripter.NET.exe /agent=menu
```



AllowMultipleInstances

You can use the AllowMultipleInstances parameter to prevent Scripter from prompting you for confirmation before opening additional instances of the application.

Syntax

```
/allowmultipleinstances
```

Example

```
InteractionScripter.NET.exe /allowmultipleinstances
```

Note: The AllowMultipleInstances command line parameter is not built into the InteractionScripter.NET.exe executable, so you won't see it listed in the display generated by the /help parameter. Instead, the AllowMultipleInstances parameter is inherited from the Common Application Framework Environment that forms the basis of many Genesys applications, such as Interaction Scripter.

Autoload

You can use the Autoload parameter to automatically load the specified HTML page(s) at startup. Multiple pages are separated by semicolons (;). Pages are displayed on separate tabs. This option is used when running Interaction Scripter in a blended environment. If the URL or file URL has spaces in the name, enclose the string in quotes to avoid invalid parsing.

Examples

```
InteractionScripter.NET.exe /notifier=myDialerServer /user=fredg
/password=1234 /workstation=Station1
/autoload=http://webServer/Script/page1.htm

InteractionScripter.NET.exe /notifier=myDialerServer /user=fredg
/password=1234 /workstation=Station1
/autoload=http://webServer/Script/page1.htm;C:\Scripts\page2.htm

InteractionScripter.NET.exe
/autoload=http://www.mysite.com;http://www.ibm.com;file://\\Server1\campaign2
\default.htm
```

Autologin

You can use the Autologin parameter to configure Scripter to bypass the logon dialog box and automatically log on if a switchover occurs. This parameter requires the name of the campaign to which Scripter should log on. The Autologin parameter can accept more than one campaign name

Syntax

```
/autologin=<campaignname>; [campaignname2]; [campaignname3]
```

Example

```
InteractionScripter.NET.exe /notifier=myDialerServer /user=craigd
/password=1234 /workstation=Station1 /autologin=HawaiinCruises;
AlaskanCruises
```

Debug

You can use the Debug parameter for debugging custom scripts while in development as it will allow you to monitor a script for errors. if an error is detected, it will be displayed in a separate dialog box.

Syntax

```
/debug
```

Example

```
InteractionScripter.NET.exe /debug
```

ForceScriptUrl

You can use the ForceScriptUrl parameter to make Scripter switch to the new active campaign when a campaign transition occurs-regardless of the ReadyState status. This parameter will force Scripter to navigate to the active campaign custom script URL when a dialer interaction is received in the event the browser location has changed. This behavior essentially ensures that Scripter loads the correct script for a given campaign.

Syntax

```
/forcescripturl
```

Example

```
InteractionScripter.NET.exe /notifier=myDialerServer /user=fredg
/password=1234 /workstation=Station1 /forcescripturl
```

Help

You can use this parameter to view a complete list of all the available command line parameters.

Syntax

```
/help
```

```
/?
```

Example

```
InteractionScripter.NET.exe /help
```

Note: When you use this parameter, you will have to log on before Scripter displays the list of available command line parameters.

InitialStatusKey

You can use the InitialStatusKey parameter to set your status before you log on. Normally the value for this option is the word Available; however, you can use any status value that matches those defined in Interaction Administrator.

Syntax

```
/initialstatuskey=<statusname>
```

Example

```
InteractionScripter.NET.exe /notifier=myDialerServer /user=fredg
/password=1234 /workstation=Station1 /initialstatuskey=available
```

Initiate

You can use the Initiate parameter to enable Dialer to place an initial call to the agent in order to establish a connection up front rather than waiting until the agent receives their first call.

Syntax

```
/initiate
```

Example

```
InteractionScripter.NET.exe /notifier=myDialerServer /user=craigd
/password=1234 /workstation=Station1 /initiate
```

LogNameSuffix

You can use the LogNameSuffix parameter to specify a custom suffix to the application log files. If you don't use this parameter, then the agent's user name appears as the suffix.

Syntax

```
/lognamesuffix=<suffixname>
```

Example

```
InteractionScripter.NET.exe /lognamesuffix=FirstRun
```

LogoutDialogOnClose

You can use the LogoutDialogOnClose parameter to configure Scripter to prompt the agent to log out before the application can be closed.

Syntax

```
/logoutdialogonclose
```

Example

```
InteractionScripter.NET.exe /logoutdialogonclose
```

NoConfirmClose

You can use the NoConfirmClose parameter to prevent Scripter to from prompting you for confirmation before exiting the application.

Syntax

```
/noconfirmclose
```

Example

```
InteractionScripter.NET.exe /notifier=myDialerServer /user=craigd
/password=1234 /workstation=Station1 /noconfirmclose
```

NoStartReceiving

If you are using a custom script that uses the IS_Action_StartReceivingCalls Predictive action, you can use the NoStartReceiving parameter to allow an agent to log on to Scripter, set their status to available, but not have Dialer place calls for them until the IS_Action_StartReceivingCalls is called. This parameter is most commonly used with Preview Campaigns.

Syntax

```
/nostartreceiving
```

Example

```
InteractionScripter.NET.exe /notifier=myDialerServer /user=fredg
/password=1234 /workstation=Station1 /nostartreceiving
```

Note: For the /nostartreceiving command-line parameter to function properly, you must also use the Dialer StartReceivingCalls Per Campaign server parameter with the value set to 1. For more information, see the Server Parameters topic in the Dialer Manager Help.

Notifier

You can use the Notifier parameter to specify the name of the Dialer Server to which you want to connect.

Syntax

```
/notifier=<server>
```

```
/n=<server>
```

Example

```
InteractionScripter.NET.exe /notifier=myDialerServer
```

Password

You can use the Password parameter to specify a CIC user's password on the command line.

Syntax

```
/password=<password>
```

```
/p=<password>
```

Example

```
InteractionScripter.NET.exe /notifier=myDialerServer /user=craigd
/password=1234
```

Persistent

You can use the Persistent parameter to specify that a remote connection should be persistent.

Syntax

```
/persistent
```

```
/persist
```

Example

```
InteractionScripter.NET.exe /notifier=myDialerServer /user=craigd
/password=1234 /remote /persistent
```

Remote

You can use the Remote parameter to log on from a remote workstation or remote number. If you are logging on from a remote number, you must specify the number as part of the parameter. If you are logging on from a remote workstation, you only need to use the base parameter.

Syntax

```
/remote[=remotenumber]

/r[=remotenumber]
```

Example

```
InteractionScripter.NET.exe /notifier=myDialerServer /user=fredg
/password=1234 /r

InteractionScripter.NET.exe /notifier=myDialerServer /user=fredg
;/password=1234 /remote=555-1212
```

Size

If you need to specify a particular size for the Scripter window, you can use the Size parameter along with height and width values in pixels.

Syntax

```
/size=<x,y>
```

Example

```
InteractionScripter.NET.exe /notifier=myDialerServer /user=fredg
/password=1234 /workstation=Station1 /size=500,500
```

Note: When the window size is specified using the Size parameter, the size of the window becomes static; meaning that agents will be unable to resize the Scripter window once the application is running.

Standalone

if you want to disable the Dialer-related functions in scripts, you can use the Standalone parameter. When you do, Scripter does not to load the PredictiveDialCOM2 interface.

Syntax

```
/standalone
```

Example

```
InteractionScripter.NET.exe /standalone
```

User

You can use the User parameter to specify a CIC user name on the command line.

Syntax

```
user=<user id>

/u=<user id>
```

Example

```
InteractionScripter.NET.exe /notifier=myDialerServer /user=craigd
/password=1234
```

Workstation

You can use the Workstation parameter to specify the name of the workstation defined in Interaction Administrator to which the user is logging on. This may or may not be the name of the physical PC that the user is sitting at. If the user is a remote agent and is also using the /remote switch, a telephone number can be substituted for the name of the station.

Syntax

```
/workstation=<station>

/w=<station>
```

Examples

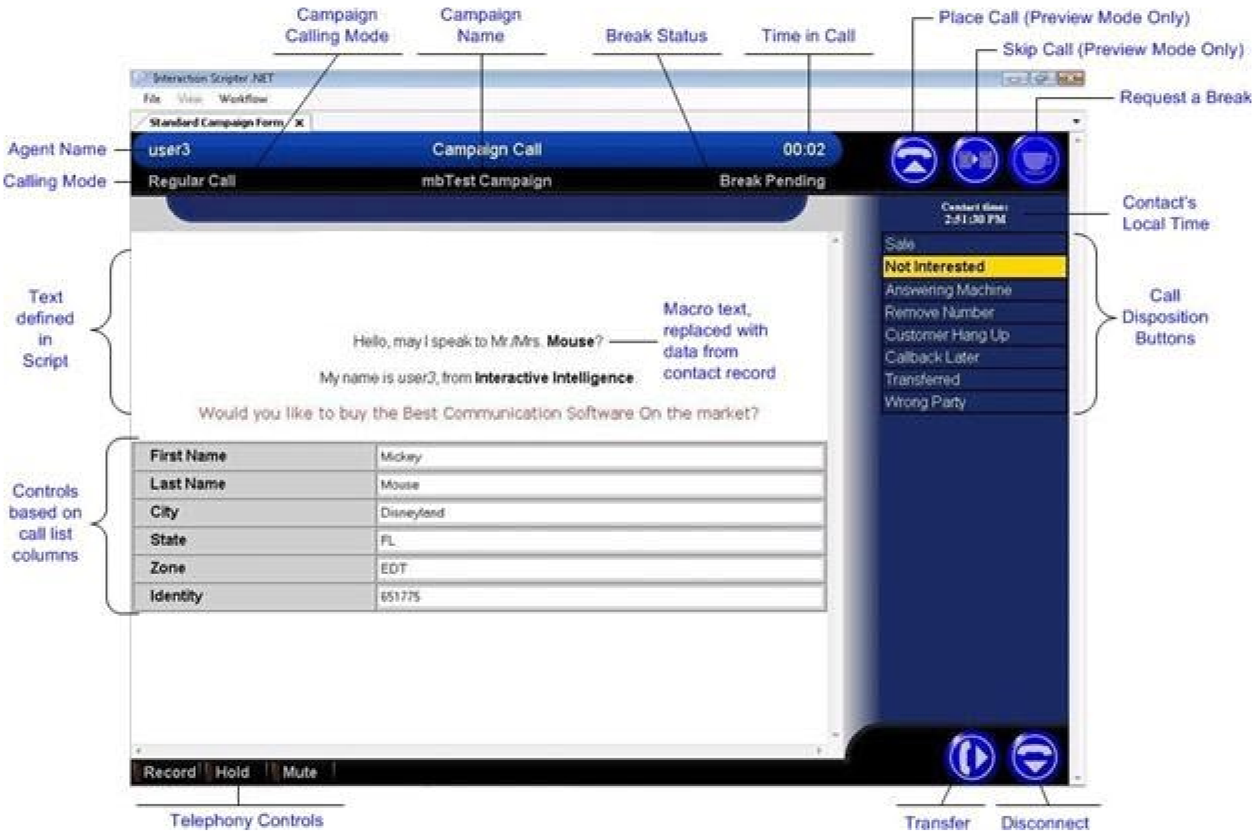
```
InteractionScripter.NET.exe /notifier=myDialerServer /user=fredg
/password=1234 /workstation=Station_001_001

InteractionScripter.NET.exe /notifier=myDialerServer /user=fredg
/password=1234 /w=317 899-9876
```

Work with Standard Forms

Standard forms are also called *base scripts*. The terms are interchangeable. Both refer to scripts created using Interaction Administrator.

Standard forms display columns from the ContactList as fields, radio buttons, or drop-down lists. When a standard form runs in Interaction Scripter, your job is to fill out fields and make selections before using a disposition button to wrap up the call. Standard forms don't tell you what to say to the contact, but are simple to understand, since everything is on one screen.



Hello, may I speak to Mr./Mrs. **Mouse**?

My name is user3, from **Interactive Intelligence**.

Would you like to buy the Best Communication Software On the market?

First Name	Mickey
Last Name	Mouse
City	Disneyland
State	FL
Zone	EDT
Identity	651775

When a Standard Form pops in Interaction Scripter, it is ready to fill in. Interaction Scripter wraps form elements in an interface that provides everything an Agent needs to disposition the call. In preview mode for example, information is presented to the Agent before the call is placed. The Agent can review the contact record and decide whether to call the contact or skip to the next contact. Agents may also request a break. Interaction Scripter provides standard telephony options found in most software phones. Agents can record, mute, transfer, hold, and disconnect calls, for example.

The Title Bar

The title bar displays the ID of the agent, the agent's status, and elapsed time in the current mode. The title bar uses color to indicate the current status of the agent or the call, so that Supervisors can view status information from a distance. The line below the title bar displays the name of the current campaign, "Waiting for Call", or "Agent Break Pending".

Color	Agent Status	Meaning
Blue	Campaign Call	A campaign call is in progress.
Red	On Break	The agent is on break.
Yellow	Follow Up	Indicates that the call is disconnected, but has not been dispositioned by the Agent.
Green	Available	Indicates that the agent is waiting for the next screen pop.

Form Area

The form itself appears below the title bar. Standard forms are customized by the Call Center Administrator to display columns from the campaign Contact List as fields, radio buttons, or drop down lists. Read-only fields have a gray background. Editable fields have a white background. Anything typed or selected in a form is saved in the ContactList database after the call is dispositioned.

Phone Number Type and Telephone Number

In Predictive campaigns, Base Scripter displays the contact's *Phone Number Type* above the form area. *Phone Number Types* are customer-defined strings that identify a type of telephone number. Examples of phone number types might be "Work", "Home", or "Cell".

In Preview mode, the contact's *Phone Number Type* and *telephone number* appear above the form area. This information helps an Agent decide whether to call the contact. For example, Agents might want to skip calls to cellular numbers, for example. In Preview mode, the displayed telephone number is editable if the campaign is configured to allow editing of the number to dial.

Place Call button

This button appears in *Preview* mode only. Preview mode pops information about the contact, but does not automatically call the contact. This gives agents a chance to review information from the database. Agents place a call to the contact by pressing this button.

Skip Call button

This button appears in *Preview* mode only. Agents press this button when they do not wish to call a contact. The agent is immediately returned to an available status.

The Break button

The *Break* button works as a toggle switch. Press this button to request a break, or to resume work after taking a break. Breaks are not always granted immediately. Whenever an agent requests a break, the system checks to see if agents are available to handle any outstanding calls. Scripter displays "Agent Pending Break" below its title bar to indicate that a break request has been made.

When there are enough agents the break request is granted, and the on-screen indicator changes to "Agent on Break". The last agent logged into a campaign is granted a break after pending calls for the agent are completed.

Transfer button

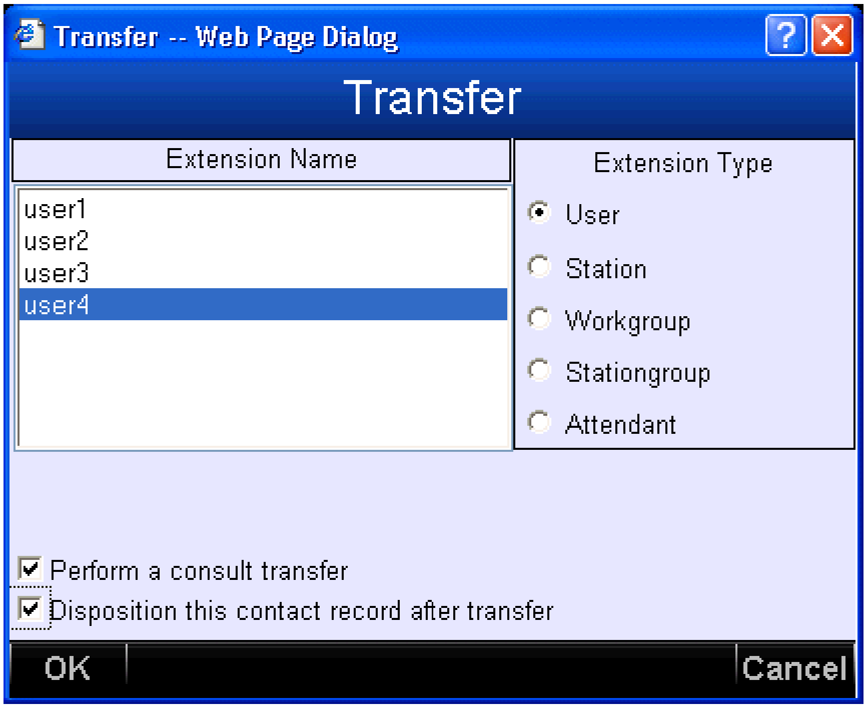
This feature transfers the current call to a user, station, workgroup, station group, or to a profile in Interaction Attendant. When this button is pressed, the Transfer dialog appears so that you can select the destination. You can make both blind and consult transfers.

To blind transfer a call

Transferring a call without speaking to the intended recipient is called a *blind transfer*.

1. Click the *Transfer* button.
2. When the Transfer dialog appears, select an *Extension Type*. This displays a list of recipients to which you can transfer the call.
3. Make a selection from the *Extension Name* list.
4. Make sure the *Perform a consult transfer* check box is not checked.
5. If you are transferring the call to an agent who is not participating in the campaign, check "Disposition this contact record after transfer". The form will remain on the screen until you disposition it by pressing a button on the right side of the display.

The Scripter transfer window adheres to the user's access control rights. An agent can see only the users, workgroups, etc. to which they have modify access rights for a given queue. You must restart Interaction Scripter before a change to access rights takes effect.



Extension Name

- user1
- user2
- user3
- user4

Extension Type

● User
○ Station
○ Workgroup
○ Stationgroup
○ Attendant

☑ Perform a consult transfer
☑ Disposition this contact record after transfer

The ‘Extension Name' list is only populated when the Dialer Server subsystem on the ODS starts up; if a new user, workgroup, etc. is added, the subsystem, CIC service, or server will need to be restarted before the new item will show in the list.

While the ‘Perform a consult transfer' check box is still available when transferring calls to an Attendant profile, consult transfers are not currently supported with Standard Forms in Scripter; even if the check box is enabled, the transfer will be performed as a blind transfer.

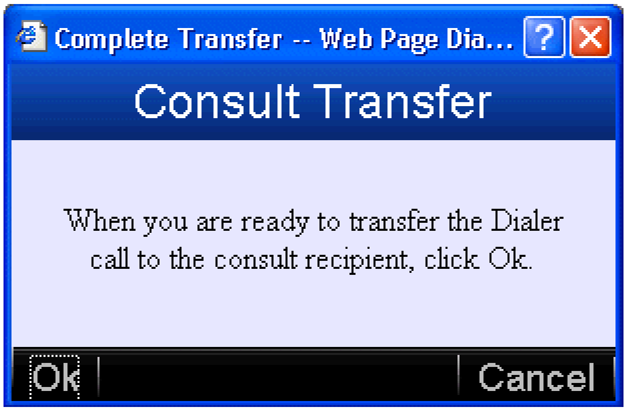
Note: If you do not select the *Disposition this contact record after transfer* check box, the wrap up code will be automatically be specified as Transferred.

6. Click OK. If you press *Cancel*, the dialog is closed without transferring the call.

To make a consult transfer

Transferring a call after speaking with the intended recipient is called a *consult transfer*.

1. Click the *Transfer* button.
2. When the *Transfer dialog* appears, select an *Extension Type*. This displays a list of recipients to which you can transfer the call.
3. Select from the *Extension Name* list. This is the person you will talk to, before transferring the call.
4. Check the *Perform a consult transfer* box.
5. If you are transferring the call to an agent who is not participating in the campaign, check "Disposition this contact record after transfer". The form will remain on the screen until you disposition it by pressing a button on the right side of the display.
6. Click *OK*. The *Consult Transfer dialog* appears. If the recipient isn't available, click *Cancel* to cancel the transfer. Otherwise, talk to the recipient as needed. When you have finished speaking and are ready to consult transfer the call, click *OK*.



Complete Transfer -- Web Page Dia...

Consult Transfer

When you are ready to transfer the Dialer call to the consult recipient, click Ok.

Disconnect button

This button disconnects the current call, and places the agent in Follow Up status until the agent dispositions the call by pressing a call disposition button.

Record button

The record button works as a toggle switch. Recording begins when the button is pressed, and ends when the button is pressed a second time or when the call is disconnected.

Hold button

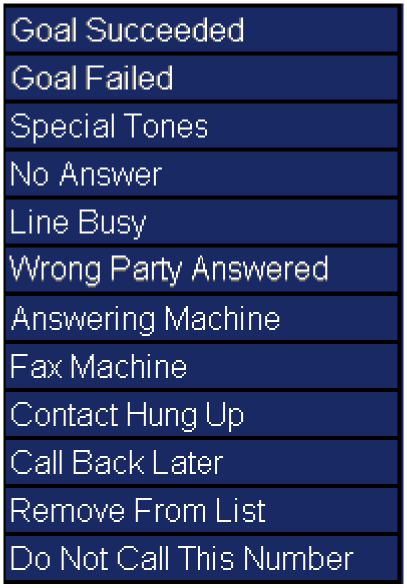
This button places the call on hold. Press Hold a second time to take the call off hold.

Mute button

Mute prevents the contact from hearing what the agent is currently saying. Press Mute a second time to allow the contact to hear again.

Call Disposition buttons

Call Disposition buttons



Call disposition buttons make it easy to close out (disposition) the current call. For example, if a call reaches an answering machine, the agent presses *Answering Machine* to hang up and reenter an available status.

Call disposition buttons are inactive unless the agent is connected to a party. The buttons highlight automatically as the cursor passes over them. This makes it easier to see which button is selected.

These buttons record whether or not the call reached the contact, and whether or not the campaign objective was met.

Only one disposition button may be pressed per call. Most calls are disconnected immediately once a call disposition is selected.

If you select *Call Back Later*, the call is disconnected when the callback is successfully scheduled. If you transfer a call, you are disconnected from the contact when *OK* is pressed after selecting a destination user, station, workgroup, station group, or Attendant profile.

What to do if the desired contact is not reached

Several call disposition buttons make it easy for agents to transition to the next call if a number is out of service, if the call is not answered, is answered by the wrong person, or is answered by a machine. Behind the scenes, Interaction Scripter assigns *reason codes* that are used later to report on the effectiveness of the campaign. Call Disposition buttons are disabled while the Agent is on break, or waiting for a call or screen pop.

Special Tones

Special Tones are sometimes called SIT tones. These tones indicate that a number is out of service, that the call could not be placed by the central office, or another type of error.

No Answer

Press this button when no one answers the call.

Wrong Party Answered

Press this button when the party who answers is not the desired contact. This option is useful when children answer, for example.

Answering Machine

Press this button when calls reach an answering machine.

Fax Machine

Press this button if you hear Fax tones, which sound like modem tones. Unlike SIT tones, Fax tones are not accompanied by a prerecorded message.

What to do when the desired contact is reached

When calls do reach the desired contact, agents usually capture information from the party by filling out the form. The last step is to disposition the call. Dispositioning ends the interaction. Suppose, for example that the agent places an order or conducts a survey. The agent would complete the form, and would press *Goal Succeeded* to disposition the call.

Goal Succeeded

This records that the call successfully met the campaign objective (e.g. a product or service was sold.)

Goal Failed

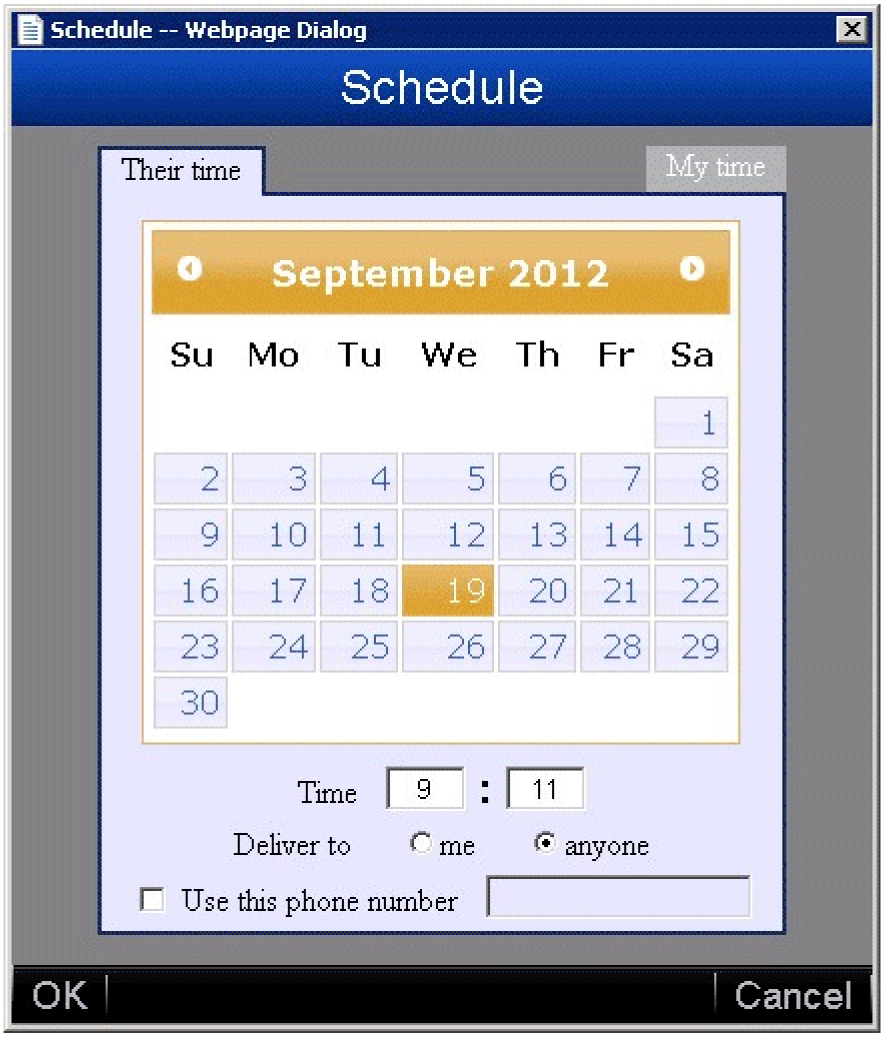
This records that a contact was reached, but the campaign call was unsuccessful. The targeted party was not interested in the product, service, or goal of the campaign.

Contact Hung Up

This records that the call was abandoned, meaning that the call was terminated by the contact or by the predictive dialer before a conversation with the agent could take place.

Call Back Later

If the contact asks to be called back at a specific time and date, press *Call Back Later* to schedule the callback. This opens the *Schedule Callback* dialog, so that you can set the date and time when the system will call the contact back.



September 2012

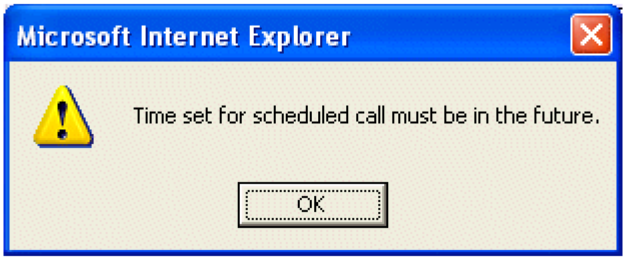
Su	Mo	Tu	We	Th	Fr	Sa
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

Time 9 : 11

Deliver to ○ me ● anyone

□ Use this phone number

Next, click on the calendar to select a day of the month. Enter a time using 24-hour notation, and then press *OK*. If you press *Cancel*, the dialog will close without scheduling a callback. If you accidentally select a past date or time, the *Schedule Callback* dialog stays open, and an alert appears to tell you to select a date and time in the future.



Time set for scheduled call must be in the future.

Finally, set the option that determines whether the scheduled callback will be routed to you, or to any available agent. If you wish to own the callback, select "me". To assign the call to anyone who is available, select "anyone". The call is immediately dispositioned when valid values are provided.

Remove from List

This option prevents the contact from being called again.

Do Not Call This Number

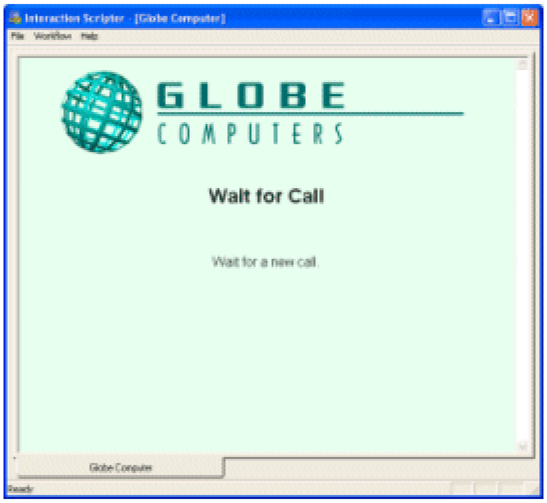
Flags this telephone number so that it is never called again.

Work with Custom Scripts

Custom campaign scripts are web-based applications that provide any needed combination of forms, text, graphics, and navigational logic. Every campaign has its own script. Custom scripts display a narrative for agents to read, and provides pushbuttons that transition to the next call stage.

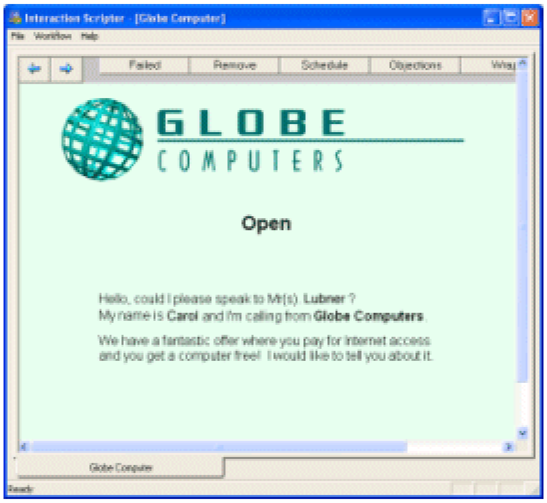
Web Page Developers use a separately licensed API called Advanced Interaction Scripter to create custom scripts. It provides the script language extensions needed to develop HTML pages that mix agent dialog with forms-based data entry and complex navigational aids. Custom campaign scripts are developed by programmers who are experienced with HTML and ECMA-262 JavaScript. However, no special expertise is needed to run custom scripts.

1. Custom scripts guide you through each stage of a campaign call by telling you what to say at each step, by providing navigation options, and by prompting you to enter data. When you log into a campaign, the default page for its script is displayed. The appearance of the page is determined by the script. The screens displayed in your call center won't visually resemble the sample screens shown below. However, the opening page usually displays a company's logo, the name of the current campaign, and other general information.



Wait for Call

Wait for a new call.



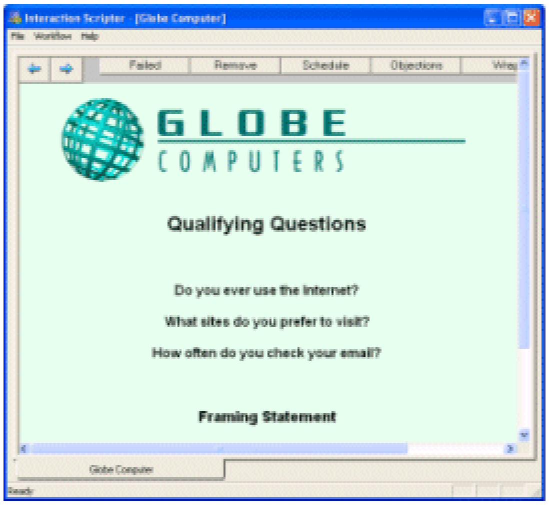
Open

Hello, could I please speak to Mr(s) **Lubner** ?
My name is **Carol** and I'm calling from **Globe Computers**.

We have a fantastic offer where you pay for Internet access and you get a computer free! I would like to tell you about it.

When an outbound call is sent to you, the script displays pages that prompt you through interactions with the targeted party. Push buttons on each screen allow you to control navigation. Information about the targeted party may be read from databases, and displayed on-screen, to help you better interact with the customer. This makes it easy to converse, since the on-screen instructions tell you what to say.

2. You may be prompted to ask qualifying questions, or to describe the features of a product or service. The script will provide navigational options (such as the arrow buttons in the example below) that take you from one screen to the next.



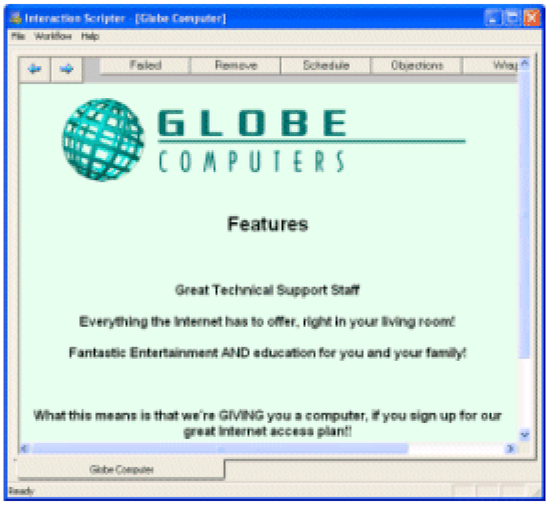
Qualifying Questions

Do you ever use the Internet?

What sites do you prefer to visit?

How often do you check your email?

Framing Statement



Features

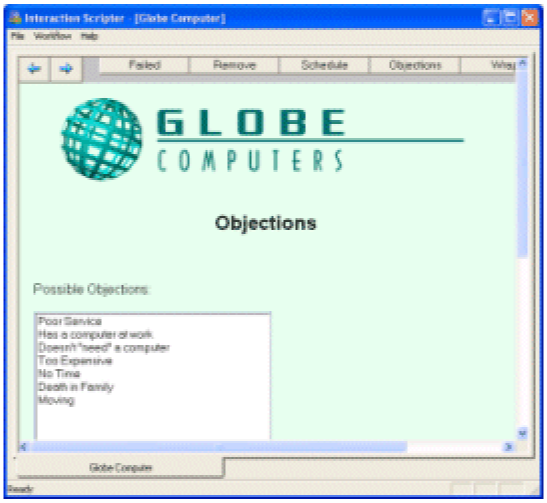
Great Technical Support Staff

Everything the Internet has to offer, right in your living room!

Fantastic Entertainment AND education for you and your family!

What this means is that we're GIVING you a computer, if you sign up for our great Internet access plan!!

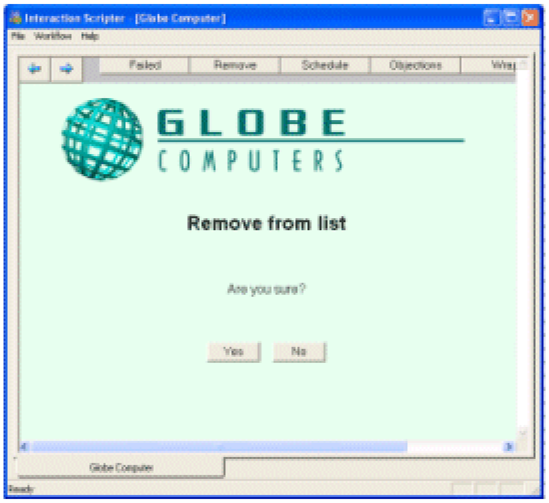
3. Since not all contacts are receptive or interested, the script will provide various branching paths that record the customer's objections, remove the customer from the ContactList, or schedule a call back. Afterwards, the call may be terminated.



Objections

Possible Objections:

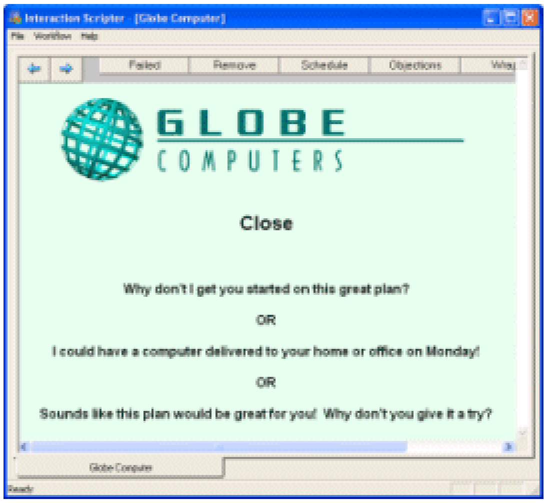
Poor Service
Has a computer at work
Doesn't "need" a computer
Too Expensive
No Time
Death in Family
Moving



Remove from list

Are you sure?

4. If the campaign call is successful, the script helps you close the deal, usually by prompting to collect information from the contact. Forms are often filled in at this stage. Wrap-up can take place after the connection has ended, or while you are speaking with the contact. It depends upon the script.



Close

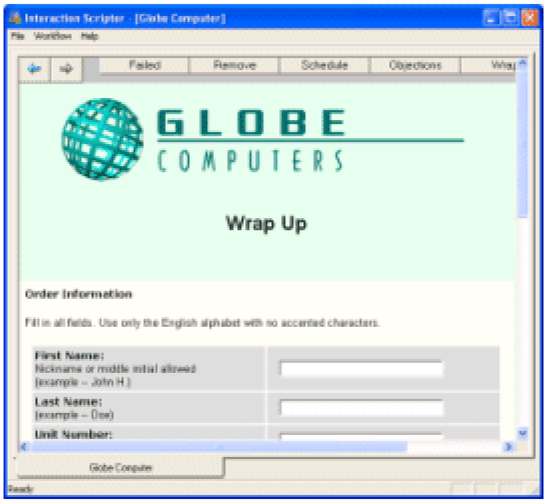
Why don't I get you started on this great plan?

OR

I could have a computer delivered to your home or office on Monday!

OR

Sounds like this plan would be great for you! Why don't you give it a try?



Wrap Up

Order Information

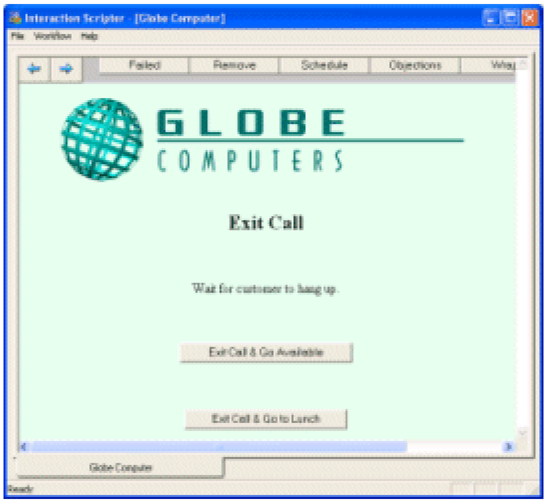
Fill in all fields. Use only the English alphabet with no accented characters.

First Name:
Nickname or middle initial allowed
(example – John H.)

Last Name:
(example – Doe)

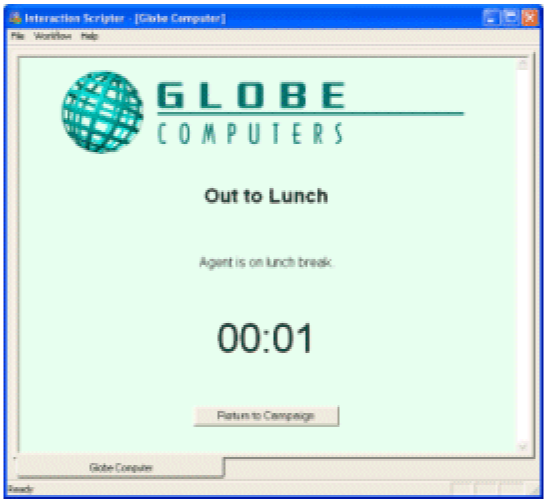
Unit Number:

5. At the end of each call, you may be given the opportunity to change your availability status. For example, a pushbutton may set your status to "At Lunch". This places your workstation in a do-not-disturb state. But most of the time you will simply make yourself available to take another call.



Exit Call

Wait for customer to hang up.



Out to Lunch

Agent is on lunch break.

00:01

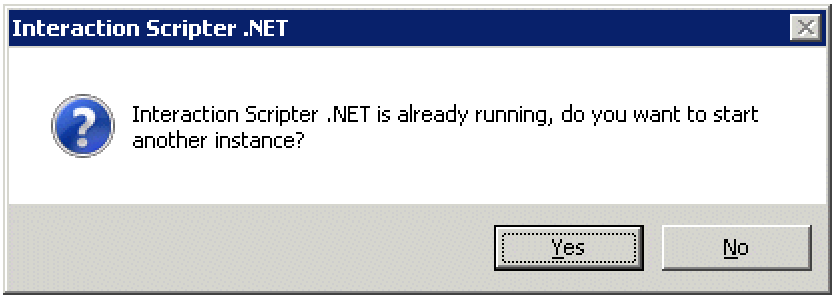
6. Scripts make it easy to return to an available status. To rejoin the campaign from a do-not-disturb status, press a button, or use whatever interface element the script designer placed on the page. Radio buttons, combo boxes, edit fields, and other standard interface elements are fully supported by Interaction Scripter. Since scripts are based upon HTML, virtually any type of information can be presented to help you interact with your contacts.

As you can see, custom scripts structure interactions with customers, and make it easy to transition between stages of a call. The information you need is on the screen when you need it, and there is no need to memorize obscure codes or phrases.

General Information

Start Interaction Scripter .NET Client

1. Press **Start**.
2. Select **Programs** > **Genesys** > **Interaction Scripter .NET**. If Interaction Scripter is already running, you will be asked to confirm whether to start another instance:

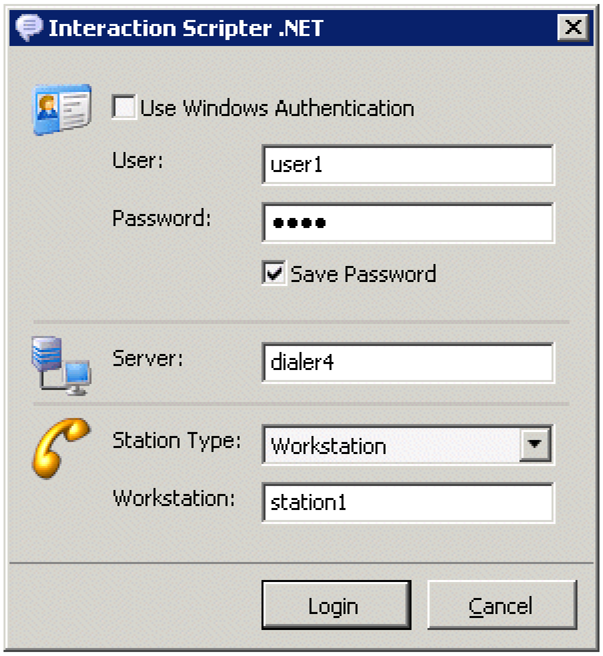


Interaction Scripter .NET

Interaction Scripter .NET is already running, do you want to start another instance?

Starting multiple instances of Interaction Scripter .NET Client is not recommended. If you see this dialog, click **No** to use the currently running instance of the program. You may need to make Scripter's window active by clicking on the task bar.

3. Interaction Scripter may present a login dialog at startup, so that it can establish a connection with your CIC server. Login with your CIC user name and password. If you check *Save Password*, the client will remember your password the next time you log in.



Interaction Scripter .NET

☐ Use Windows Authentication

User: user1

Password: ●●●●

☑ Save Password

Server: dialer4

Station Type: Workstation

Workstation: station1

4. Type the name of your Outbound Dialer server in the *Server* field. You can optionally enter its IP address.
5. Select a Station Type. The choices are:

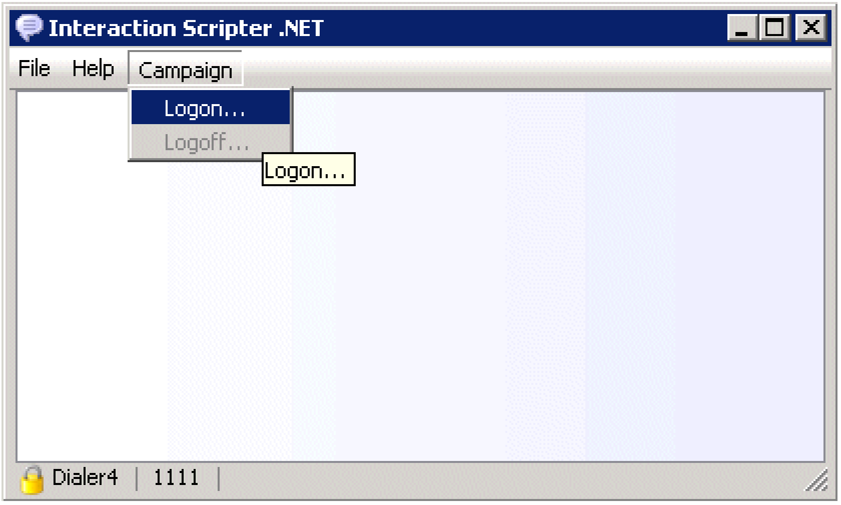
<this computer>—sets the station name to the name of your computer.

Workstation—select this if you have a station phone directly connected to your CIC server. When this option is selected, you must also type your station name in the *Station* field.

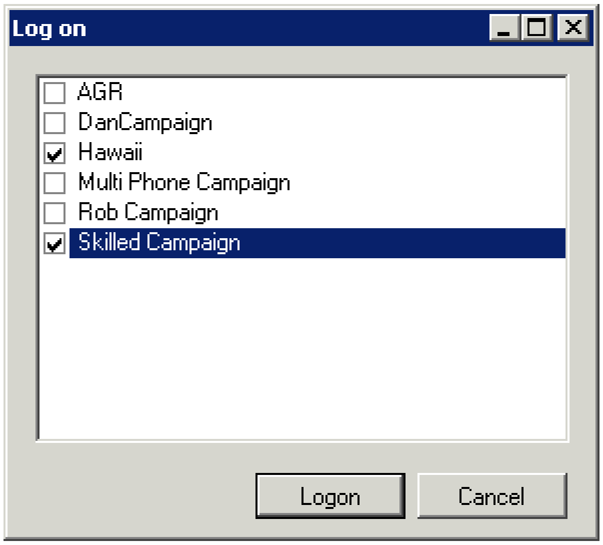
Remote Workstation—select this if you are connecting to a remote station. You may specify a remote phone number if allowed by your Administrator.

Remote Number—select this option if you are working as a Remote Agent. When this option is selected, enter your telephone number in the Remote Number field instead of a station name. If you enter a slash in front of a telephone extension, the server will dial the main number, and will wait for a connection before dialing the extension number. For example: 317 899-9876/113

6. If a language pack has been installed, Interaction Scripter .NET Client's Login dialog displays an *Options* tab that allows the user to select a language other than English. To select a language:
   a. Select the **Options** tab.
   b. Select a language from the *Language* list box.
7. Click **Login**. Wait while the client establishes a session with the server.
8. The client will display a blank page, because you are connected to the CIC server, but have not logged into any campaigns yet. Pull down the **Campaign** menu and select **Logon**.



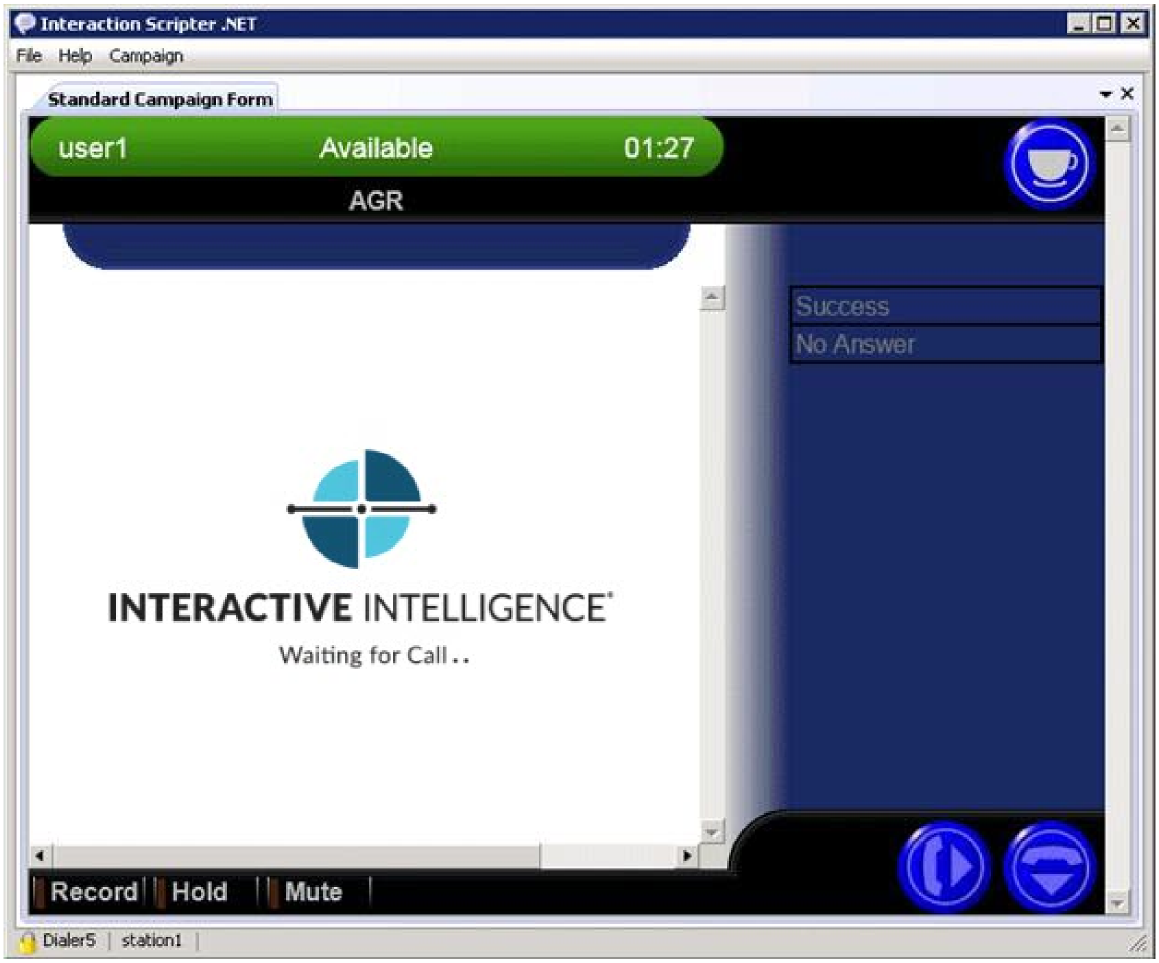
9. If the **Login Campaign** security right is assigned to you, the **Log on** dialog appears, so that you can place a checkmark next to campaigns that you wish to accept calls from.



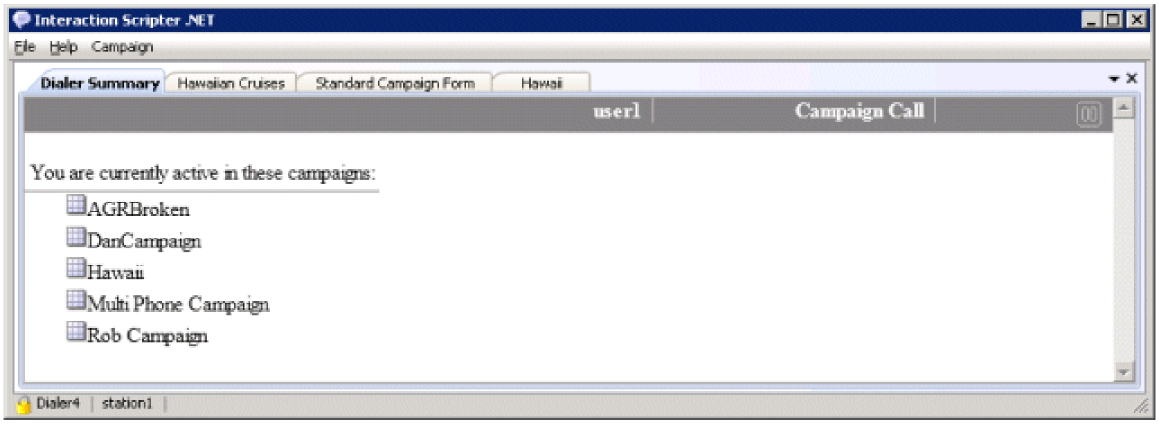
□ AGR
□ DanCampaign
☑ Hawaii
□ Multi Phone Campaign
□ Rob Campaign
☑ Skilled Campaign

The **Log on** dialog does not appear unless the **Login Campaign** right is assigned to you. If you do not have that right, you are automatically logged into all campaigns, and Interaction Scripter .NET Client opens the default page for each campaign's script on individual tab pages.

The tab for each campaign will display the script for a campaign call, or a page that indicates you are waiting for a call:



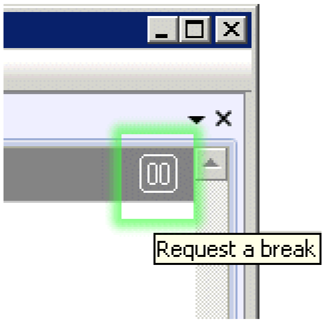
If the *Login Campaign* right is not assigned to you, an additional **Dialer Summary** tab will appear. It lists the campaigns you are active in.



You are currently active in these campaigns:

- AGRBroken
- DanCampaign
- Hawaii
- Multi Phone Campaign
- Rob Campaign

You can request a break by pressing a button on the Dialer Summary page:



10. You are now ready to take campaign calls.

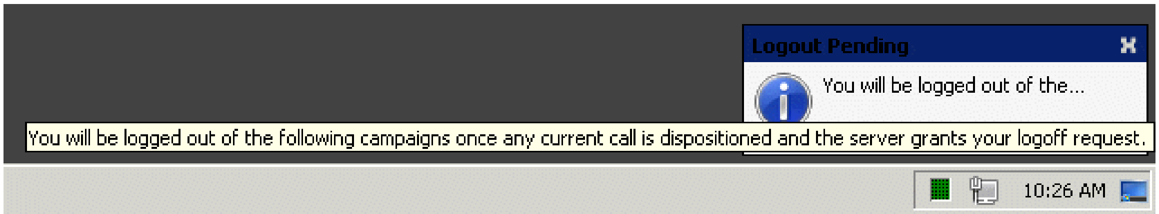
Log out of campaigns

You cannot logoff if a scheduled callback is pending. Interaction Scripter will display a message indicating that there is an own agent callback pending. After you disposition the scheduled call and return to an available status, the system will log you off automatically.

To log out of campaigns:

1. Pull down the *Campaign* menu and select *Logoff*.
2. If you are in a call, you will be logged out as soon as you disposition (finish) the call. Otherwise, you are logged out immediately, unless a scheduled callback is pending.

   Notification messages pop in the client window if Interaction Scripter Client is waiting for you to do something.



Logout Pending

You will be logged out of the...

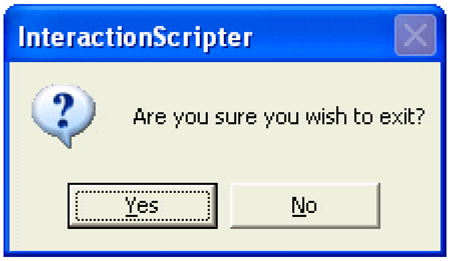
You will be logged out of the following campaigns once any current call is dispositioned and the server grants your logoff request.

To display the entire message in a tool tip, hold the mouse over the popup window.

Close Interaction Scripter

To shut down Interaction Scripter, follow these steps:

1. Pull down the *Campaign* menu and select *Logoff*.
2. Disposition the current call, if any.
3. Pull down the *File* menu and select *Exit*.
4. Press *Yes* to confirm that you wish to exit.


InteractionScripter

Are you sure you wish to exit?

Open the About dialog to display version and contact information

The *About* dialog provides information that may be useful in a technical support situation. It provides general information about your computing environment, and helps you access Genesys resources on the World Wide Web.

To open the *About* dialog, pull down the *Help* menu and select *About* (or *About application name*). This dialog displays:

- Interaction Center version number
- Name and version of the program that opened this About dialog.
- User name
- Computer name
- Operating System and version
- Program copyrights
- Organization or person that the application is licensed to.

Ok button

Closes the About dialog.

CIC Info button

Click *CIC Info* for information about your CIC server. It opens a dialog which displays:

- CIC Username
- Notifier Key
- Notifier Server (current)
- Notifier Server (registry)

- Notifier Connection Open (whether yes or no)
- Using Backup Server (whether yes or no)
- Debug—Yes if the application is running in debug mode; otherwise No.
- String Type—indicates whether the application uses ANSI or Unicode strings.
- MAPI Installed—Yes if MAPI is installed; otherwise No.

IP Configuration button

Click the *IP Configuration* button to display Internet Protocol (IP) information. This opens the IP Configuration dialog, which displays information about the network adaptor and IP configuration:

Ethernet adapter / Local Area Connection	**Windows IP Configuration**
Connection-specific DNS Suffix	
Description	
Physical Address	
DHCP Enabled	
Auto-configuration Enabled	Host Name
IP Address	Primary DNS Suffix
Subnet Mask	Node Type
Default Gateway	IP Routing Enabled
DHCP Server	WINS Proxy Enabled
DNS Servers	DNS Suffix Search List
Primary WINS Server	
Secondary WINS Server	
Lease Obtained date	
Lease Expires date	

When you are finished, dismiss the *IP Configuration* dialog by pressing the *Close* button.

System Info button

Click this button to access numerous details about your computer system. It opens the *System Information* dialog, which displays a wealth of information, organized by category:

- Hardware Resources
- Components
- Software Environment
- Internet Settings
- Office 200x Applications

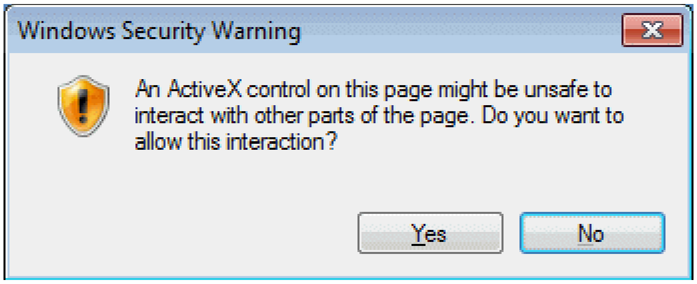
When you are finished, dismiss the dialog by pressing the *Close* button.

Contact Genesys button

Click this button to open Genesys online resources in your web browser.

What to do if you get a Windows Security Warning

In rare cases, agents may see a Windows Security warning:

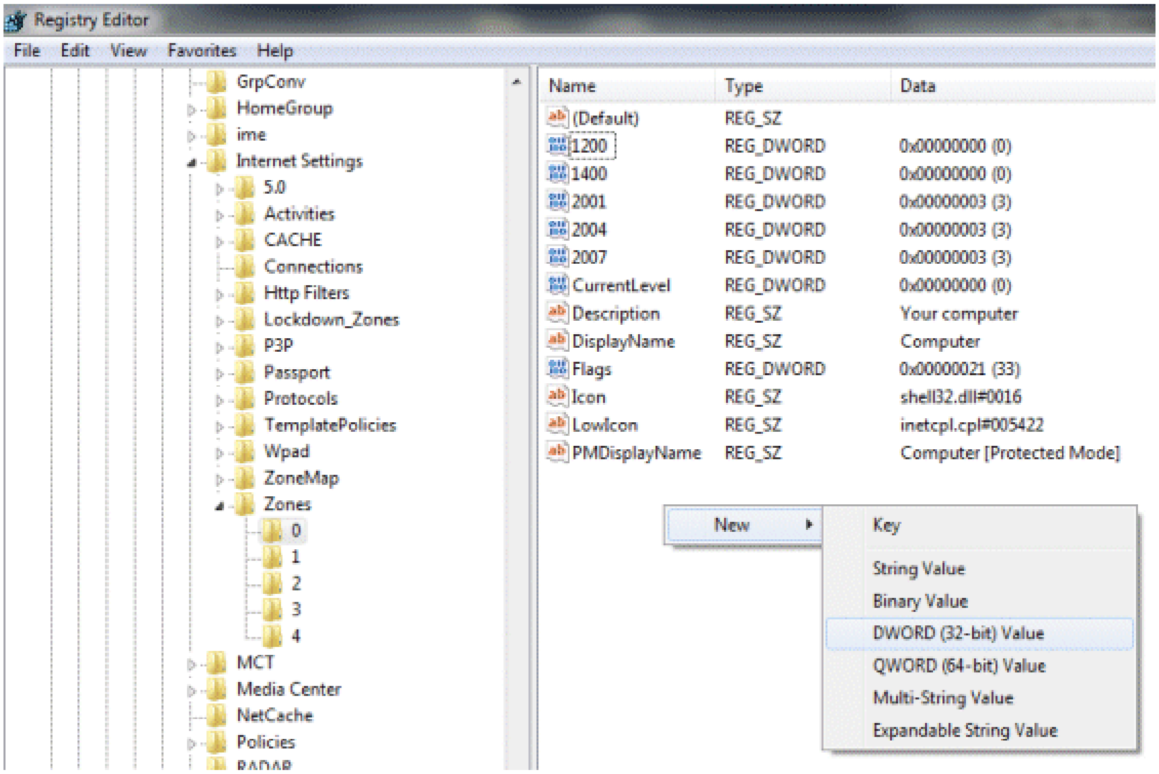


An ActiveX control on this page might be unsafe to interact with other parts of the page. Do you want to allow this interaction?

If this message appears when you attempt to create a callback in Interaction Scripter .NET Client by clicking a "schedule" disposition, follow the instructions below:

1. Click **Start**.
2. Type **Regedit** in the *Search Programs and Files* box.
3. If prompted to allow changes to this computer, click **Yes**.
4. Expand **HKEY_CURRENT_USER**.
5. Then expand Software > Microsoft > Windows > CurrentVersion > Internet Settings > Zones.
6. Select the "**0**" folder.

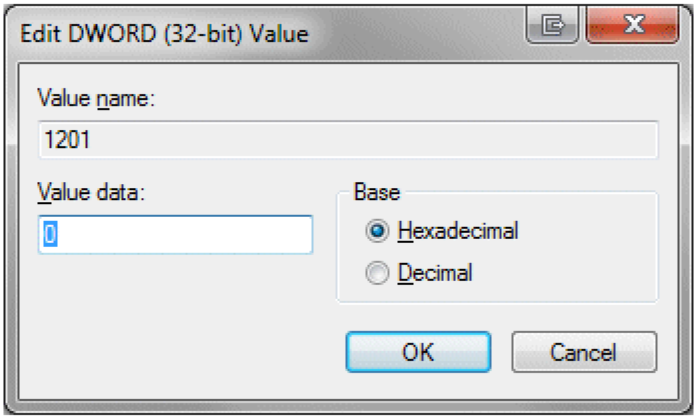
7. Click the right mouse button to open the shortcut menu. Select **New** > **DWORD (32-bit) Value**.



Name	Type	Data
(Default)	REG_SZ	
1200	REG_DWORD	0x00000000 (0)
1400	REG_DWORD	0x00000000 (0)
2001	REG_DWORD	0x00000003 (3)
2004	REG_DWORD	0x00000003 (3)
2007	REG_DWORD	0x00000003 (3)
CurrentLevel	REG_DWORD	0x00000000 (0)
Description	REG_SZ	Your computer
DisplayName	REG_SZ	Computer
Flags	REG_DWORD	0x00000021 (33)
Icon	REG_SZ	shell32.dll#0016
LowIcon	REG_SZ	inetcpl.cpl#005422
PMDisplayName	REG_SZ	Computer [Protected Mode]

8. Type **1201**. Then press **Enter**.

9. Double-click **1201** to open its properties. Make sure that its value is zero.



Value name: 1201

Value data: 0

Base

● Hexadecimal

○ Decimal

10. Click **OK** to close the dialog.

11. Choose **File** > **Exit** to end the registry editor.

ActiveX control in a custom script

If the message appears while you are running a custom script, it is likely that the script contains an ActiveX control. Follow the steps below to solve this problem in Windows 7 with IE 9. These settings allow sites in the list of trusted sites to run potentially harmful or malicious ActiveX controls. Add only sites that you are completely confident of to this list.

Tip: Instead of steps 1 through 3, click **Start,** and then enter **inetcpl.cpl**.

1. Click **Start** -> **Control Panel**.
2. Click **Network and Internet**.
3. Click **Internet Options**.
4. Click the **Trusted Sites** zone icon.
5. Click **Custom Level**.
6. Move the slider to **Low**.
7. Click **Sites**.
8. In the Trusted Sites dialog box, copy the URL of the Customer Interaction Center into the **Add this website to the Zone** box. This URL is similar to https://webclient.yourcompany.com/WebClient/WebClientPage.aspx.
9. Click **Add**.
10. Click **Close**.
11. In the **Internet Properties** dialog box, click **OK**.

What to do when the Skip button is disabled on a Preview Call

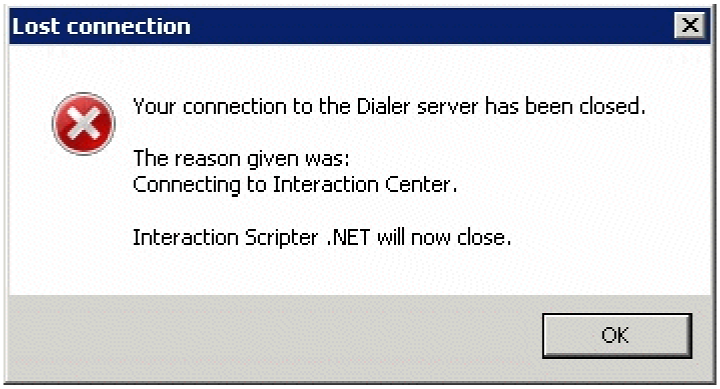
For a base script, the Skip button is grayed out while agents are on a connected Preview Call. While on a custom script Preview Call, the Skip Call button is disabled to prevent things getting into a bad state. Since the agent cannot Skip the call, she must manually disconnect the call, or disconnect using the My Interactions page.

If the agent does not disconnect the call, here's what happens while on a custom script Preview Call with Campaign Call status with "Preview Call" Dialer status:

- Agent clicks the Skip Call button. Status changes to Follow Up and Dialer status changes to Waiting For Call.
- Agent clicks the Available button. Status changes to Available, with Dialer status Waiting For Call.
- Clicking Disconnect or Disposition does not disconnect the call.

Automatic Switchover to alternate server

This procedure applies to Interaction Dialer environments only. Your Outbound Dialer Server may be configured to pass control to an alternate ODS server in the event of a catastrophic hardware failure. When this *switchover* takes place, Interaction Scripter .NET Client immediately closes. It displays a message to inform that its connection to the ODS server was lost:



Lost connection

Your connection to the Dialer server has been closed.

The reason given was:
Connecting to Interaction Center.

Interaction Scripter .NET will now close.

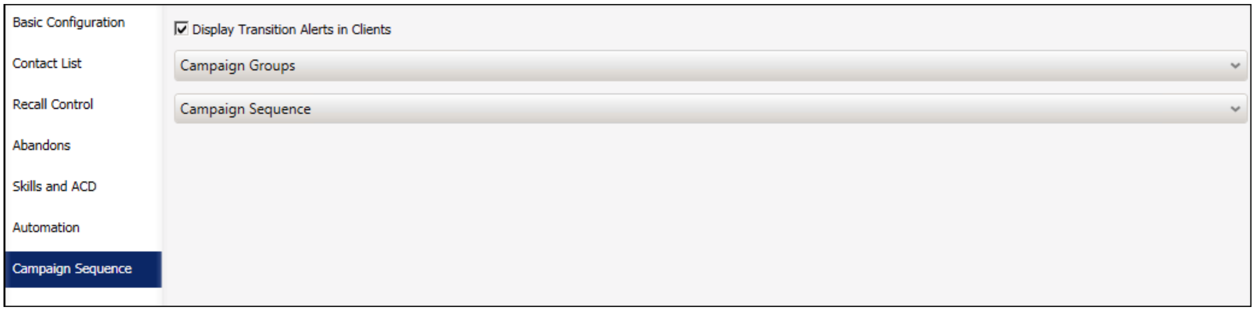
Interaction Scripter automatically restarts and logs the agent into campaigns that the agent was logged into when the server failed.

Contact Genesys

It is easy to access online resources from Genesys applications. From the **Help** menu, select **Genesys on the Web**, or **Help > About**. From there, you can navigate to Genesys sites using the default web browser installed on your computer.

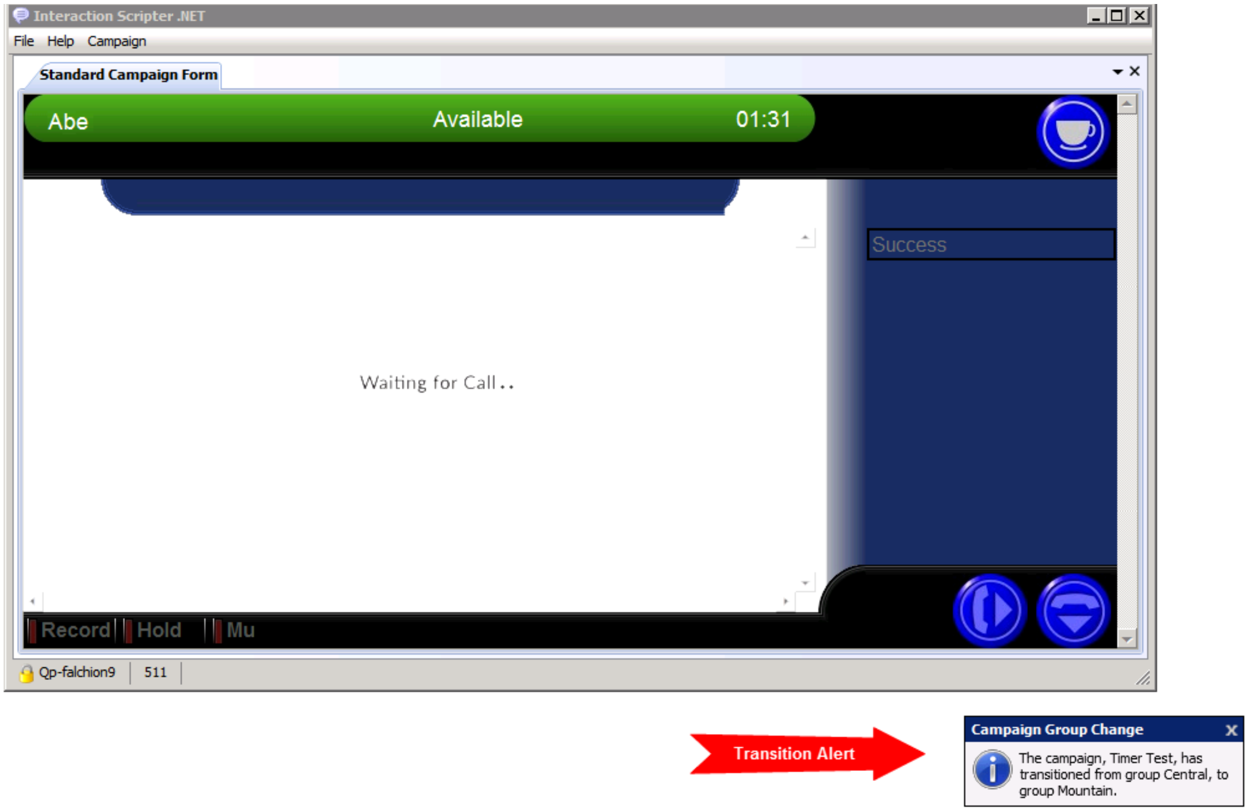
Advanced Campaign Management

If you have installed the Advanced Campaign Management Dialer Feature License, you'll see the Campaign Sequence tab in the Campaign Properties Panel. At the top of the tab, you'll see the Display Transition Alerts in Clients check box. Select the Transition Alerts in Clients check box if you want to Dialer to display a toast notification in Scripter to alert your agents when a campaign transition occurs. If you don't enable this feature, agents will not be aware of campaign transitions.



☑ Display Transition Alerts in Clients

Selecting the Display Transition Alerts in Clients check box on the Campaign Sequence tab will configure Dialer to display toast notifications in Interaction Scripter whenever a campaign transition occurs. Notifications appear in the bottom right corner of the screen.



Campaign Group Change

The campaign, Timer Test, has transitioned from group Central, to group Mountain.

You can learn more about this feature in the *Advanced Campaign Management for Interaction Dialer - Overview Guide*, which is available in the Interaction Dialer section of the PureConnect Documentation Library.

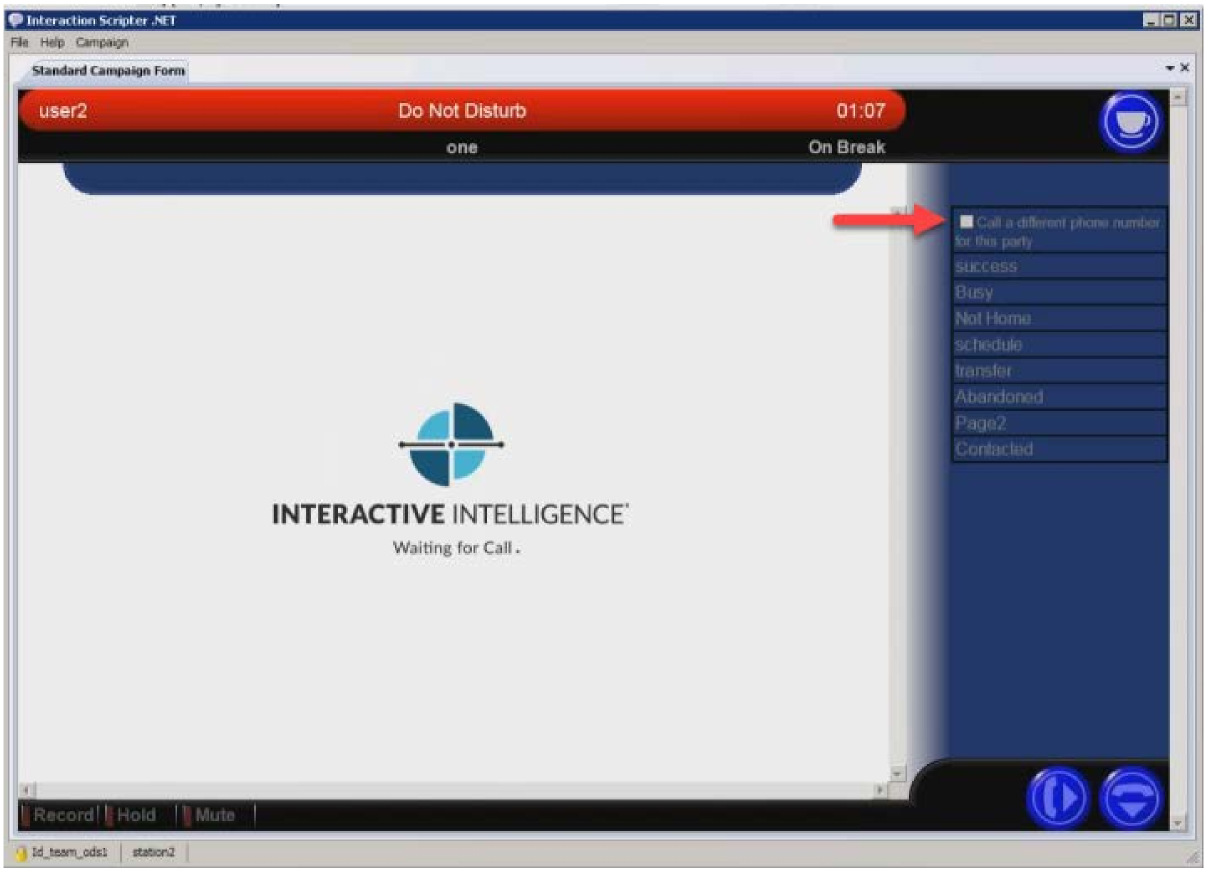
Follow Up Call feature

The Follow Up Call feature is designed to allow Dialer customers to be able to place multiple, consecutive calls within a preview campaign to a single contact based on a disposition selected for one of the numbers associated with the contact.

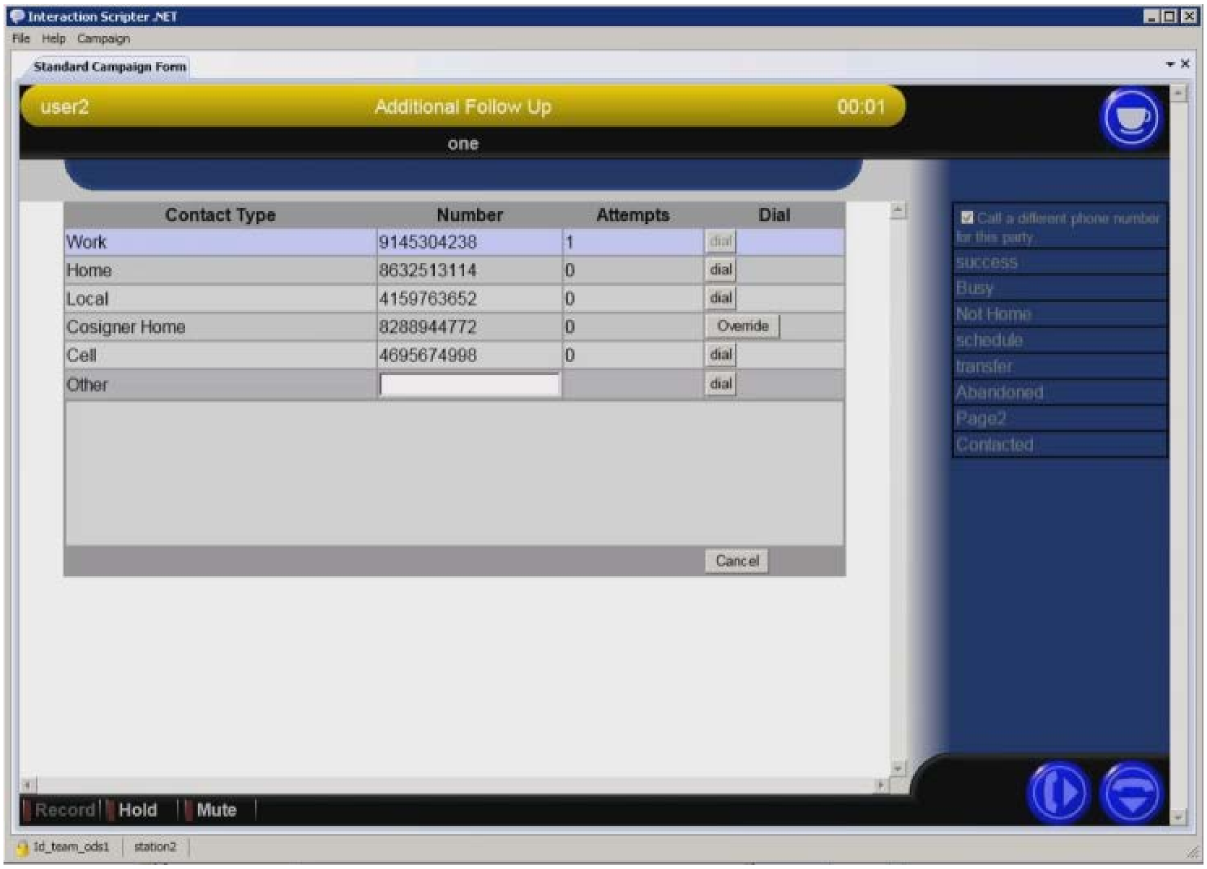
For example, suppose that an agent dials a number, reaches the contact, but is asked to call back on another number that is either already on the contact record or provided to the agent by the contact. Using the Follow Up Call feature, the agent will be able make the additional calls from within Dialer where the calls will be logged and will fall under Dialer's compliance controls.

If you have enabled the Modify Base Script Compliance Overrides security right for an administrator and the Allow Agent Compliance Override security right for an agent, you can take advantage Follow Up Call feature.

When a Base script that uses the Follow Up Call feature appears in Scripter, you'll see the "Call a different phone number for this party" check box appear at the top of the Call Disposition panel.



When you select the "Call a different phone number for this party" check box and disposition the call, you'll see the Additional Follow Up window, which shows all of the numbers on the contact record. You'll also find a field into which you can manually enter a phone number to make an ad hoc call. While in this screen, an agent's status will be set to Additional Follow Up.



Contact Type	Number	Attempts	Dial
Work	9145304238	1	dial
Home	8632513114	0	dial
Local	4159763652	0	dial
Cosigner Home	8288944772	0	Override
Cell	4695674998	0	dial
Other			dial

Numbers that can be dialed immediately, will have a dial button. Numbers that are blocked, due to DNC block or timezone block, will have an Override button. As long as the agent has the Allow Agent Compliance Override security right, they can click the Override button and make the call anyway.

You can learn more about the Follow Up Call feature in Interaction Dialer Manager Help, which is available in the PureConnect Documentation Library.

Copyright and Trademark Information

Interaction Dialer and *Interaction Scripter* are registered trademarks of Genesys Telecommunications Laboratories, Inc. The foregoing products are ©2000-2017 Genesys Telecommunications Laboratories, Inc. All rights reserved.

Messaging Interaction Center and *MIC* are trademarks of Genesys Telecommunications Laboratories, Inc. The foregoing products are ©2001-2017 Genesys Telecommunications Laboratories, Inc. All rights reserved.

Interaction Director is a registered trademark of Genesys Telecommunications Laboratories, Inc. *e-FAQ Knowledge Manager* and *Interaction Marquee* are trademarks of Genesys Telecommunications Laboratories, Inc. The foregoing products are ©2002-2017 Genesys Telecommunications Laboratories, Inc. All rights reserved.

Interaction Conference is a trademark of Genesys Telecommunications Laboratories, Inc. The foregoing products are ©2004-2017 Genesys Telecommunications Laboratories, Inc. All rights reserved.

Interaction SIP Proxy and *Interaction EasyScripter* are trademarks of Genesys Telecommunications Laboratories, Inc. The foregoing products are ©2005-2017 Genesys Telecommunications Laboratories, Inc. All rights reserved.

Interaction Gateway is a registered trademark of Genesys Telecommunications Laboratories, Inc. *Interaction Media Server* is a trademark of Genesys Telecommunications Laboratories, Inc. The foregoing products are ©2006-2017 Genesys Telecommunications Laboratories, Inc. All rights reserved.

Interaction Desktop is a trademark of Genesys Telecommunications Laboratories, Inc. The foregoing products are ©2007-2017 Genesys Telecommunications Laboratories, Inc. All rights reserved.

Interaction Process Automation, Deliberately Innovative, Interaction Feedback, and *Interaction SIP Station* are registered trademarks of Genesys Telecommunications Laboratories, Inc. The foregoing products are ©2009-2017 Genesys Telecommunications Laboratories, Inc. All rights reserved.

Interaction Analyzer is a registered trademark of Genesys Telecommunications Laboratories, Inc. *Interaction Web Portal* and *IPA* are trademarks of Genesys Telecommunications Laboratories, Inc. The foregoing products are ©2010-2017 Genesys Telecommunications Laboratories, Inc. All rights reserved.

Spotability is a trademark of Genesys Telecommunications Laboratories, Inc. ©2011-2017. All rights reserved.

Interaction Edge, CaaS Quick Spin, *Interactive Intelligence Marketplace, Interaction SIP Bridge,* and *Interaction Mobilizer* are registered trademarks of Genesys Telecommunications Laboratories, Inc. *Interactive Intelligence Communications as a Service℠* and *Interactive Intelligence CaaS℠* are trademarks or service marks of Genesys Telecommunications Laboratories, Inc. The foregoing products are ©2012-2017 Genesys Telecommunications Laboratories, Inc. All rights reserved.

Interaction Speech Recognition and *Interaction Quality Manager* are registered trademarks of Genesys Telecommunications Laboratories, Inc. *Bay Bridge Decisions* and *Interaction Script Builder* are trademarks of Genesys Telecommunications Laboratories, Inc. The foregoing products are ©2013-2017 Genesys Telecommunications Laboratories, Inc. All rights reserved.

Interaction Collector is a registered trademark of Genesys Telecommunications Laboratories, Inc. *Interaction Decisions* is a trademark of Genesys Telecommunications Laboratories, Inc. The foregoing products are ©2013-2017 Genesys Telecommunications Laboratories, Inc. All rights reserved.

DISCLAIMER

GENESYS TELECOMMUNICATIONS LABORATORIES (GENESYS) HAS NO RESPONSIBILITY UNDER WARRANTY, INDEMNIFICATION OR OTHERWISE, FOR MODIFICATION OR CUSTOMIZATION OF ANY GENESYS SOFTWARE BY GENESYS, CUSTOMER OR ANY THIRD PARTY EVEN IF SUCH CUSTOMIZATION AND/OR MODIFICATION IS DONE USING GENESYS TOOLS, TRAINING OR METHODS DOCUMENTED BY GENESYS.

Genesys Telecommunications Laboratories, Inc.
2001 Junipero Serra Boulevard
Daly City, CA 94014
Telephone/Fax (844) 274-5992
www.genesys.com

Compliance

Please note that it is the sole responsibility of the user of this software to comply with all federal, state, and local laws applicable to the software, the use thereof, and the conduct of the user’s business. In no event will Genesys Telecommunications Laboratories, Inc. (“Genesys”) be responsible for providing, implementing, configuring, or coding software in a manner that complies with any laws or regulatory requirements that apply to the user’s business or industry, including, without limitation, U.S. Federal Trade Commission (FTC) regulations, Federal Communications Commission (FCC) regulations, the Telephone Consumer Protection Act (TCPA) of 1991, and the Health Insurance Portability and Accountability Act (HIPAA) (collectively “Customer Specific Laws”). The user agrees that it will comply with all such Customer Specific Laws and, regardless of anything to the contrary, in no event will Genesys, its affiliates, or related entities be held liable for any claim or action arising from, or related to, the user’s failure to comply with any Customer Specific Laws. The above conditions apply regardless of anything to the contrary and your use of Interaction Dialer constitutes your acceptance of the above provisions.

Revisions

This topic summarizes changes in Interaction Scripter .NET Client.

Interaction Scripter 2018 R3

1. Starting with PureConnect 2018 R3, Dialer agents can use Interaction Connect or Interaction Scripter .NET to process outbound calls that use custom scripts. As before, both clients support base scripts.
2. Interaction Connect now allows agents executing a base script to change their user status when they are not involved in a Dialer interaction. Agents can change their status before logging into a campaign, or while on break.
3. Noted in the Custom Scripts topic that Interaction Connect supports both base scripts and custom scripts. Due to differences in the way that Scripter .NET and Interaction Connect execute JavaScript, custom scripts must be programmed to target one client or the other. See I*nteraction Scripter Developer's Guide* for details.

Interaction Scripter 2018 R2

- A new server parameter resolves an issue caused by email and chat interactions entering Followup status. When an agent's status moves to Followup from a non-fully utilized chat or email interaction, this causes base scripts to put the agent into Do Not Disturb status. An agent who is on a chat or email interaction will remain in the Available status and, therefore, is considered to be Idle by Dialer.

  When the agent moves from these interactions into a Followup status, Dialer will put the agent on break, since they moved from Idle to a non-ACD status. Because Dialer is putting the agent on break, base scripts in InteractionScripter.Net move the agent to the Do Not Disturb status. This occurs while the agent is still in Followup.

  In summary, if an agent logged into a base script in InteractionScripter.Net while moving to Followup from an email or chat interaction, this will cause the agent's status to immediately enter Do Not Disturb status. Enabling the "Dialer Prevent Autobreak On Workgroup Followup" server parameter prevents this from occurring. See also, Optional Server Parameters in Interaction Dialer Manager Help.
- Added What to do when the Skip button is disabled on a Preview call.

Interaction Scripter 2018 R1

No changes.

Interaction Scripter 2017 R4

- The information in this document pertains to Interaction Scripter .NET Client only. It does not pertain to a new client that runs inside Interaction Connect. For information about Scripter with Interaction Connect, see Interaction Connect Help.
- Explained that *base scripts* and *standard forms* are interchangeable terms. Both refer to the on-screen forms that your Call Center Administrator created to help you process a call.
- Updated formatting to improve readability.
- Rebranded topics to apply Genesys terminiology and styling.
- Updated copyright, trademark, and compliance topics.

Interaction Scripter 2017 R1

- Updated access control rights information in To Transfer a Blind Call of the Work with Standard forms section.
- Added general information about Follow Up Call feature.

Interaction Scripter 2016 R4

- Minor documentation corrections.

Interaction Scripter 2016 R3

- Slight reorganization of Help system content.

Interaction Scripter 2016 R2

- Added the Advanced Campaign Management topic to the General Information section.
  - You can learn more about this new feature in the *Advanced Campaign Management for Interaction Dialer - Overview Guide*, which is available in the Interaction Dialer section of the PureConnect Documentation Library.

Interaction Scripter 2016 R1

- Updated documentation and screen shots to reflect new logo and color scheme.

Interaction Scripter 2015 R4

- Slight reorganization of Help system content.

Interaction Scripter 2015 R3

- The Autologin command line parameter, which allows you to configure Scripter to bypass the logon dialog box and automatically log on if a switchover occurs, now provides support for logging on to multiple campaigns.

Interaction Scripter 2015 R2

- Added information about a new command line parameter called AllowMultipleInstances.
- Updated the Nostartreceiving topic with information pertaining to the use of the Dialer StartReceivingCalls Per Campaign server parameter.

Interaction Scripter 2015 R1

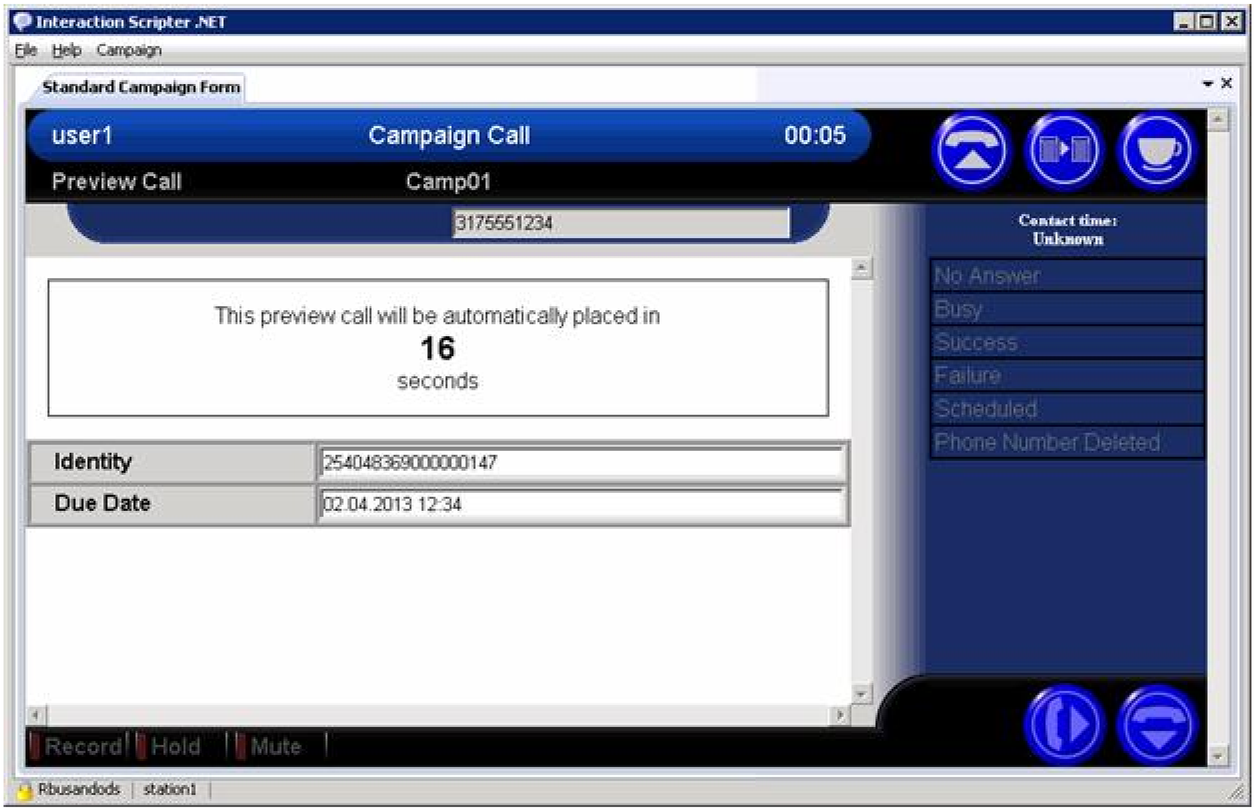
- Updated documentation to reflect changes required in the transition from version 4.0 SU# to Interaction Dialer 2015 R1, such as updates to product version numbers, system requirements, installation procedures, references to Interactive Intelligence Product Information site URLs, and copyright and trademark information.

Interaction Scripter 4.0 Service Update 3

1. Added a new optional command line parameter, /forcescripturl, to ensure that Scripter loads the correct script when a campaign transition occurs.

Interaction Scripter 4.0 Service Update 2

Interaction Scripter now displays a countdown timer that indicates how much time agents have to review the preview pop before the contact is automatically dialed. Agents cannot stop a countdown, but they can skip preview calls as before, if the script provides a Preview Skip option. The preview countdown feature must be configured for the campaign. By default, agents have an unlimited amount of time to review information about each preview call.



user1 Campaign Call 00:05

Preview Call Camp01

3175551234

Contact time:
Unknown

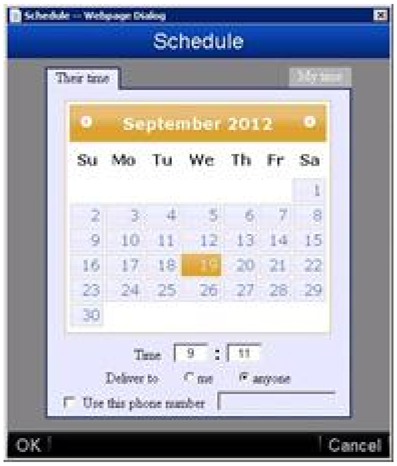
This preview call will be automatically placed in
16
seconds

Identity	254048369000000147
Due Date	02.04.2013 12:34

Interaction Scripter 4.0 Service Update 1

1. If a language pack has been installed, Interaction Scripter .NET Client's Login dialog displays an Options tab that allows the user to select a language other than English. For details, see Start Interaction Scripter .NET Client.

2. Interaction Scripter Client no longer uses an ActiveX control to display date\time selection when an agent dispositions a call to schedule. A JavaScript control is used instead. This eliminates security issues in Internet Explorer associated with the ActiveX control.



September 2012

Su	Mo	Tu	We	Th	Fr	Sa
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

Time 9 : 11

Deliver to ○ me ● anyone

□ Use this phone number

The new date and time picker.

Interaction Scripter 4.0

1. Starting with this release, **Interaction Scripter Client Win32 Edition** is retired. Interaction Scripter .NET Edition Client is now used exclusively to run base scripts and custom scripts developed with the Interaction Scripter API.

   The Win32 edition relied on code libraries developed by Interactive Intelligence for CIC and Dialer integration, such as the Interaction Client Win32 COM API (ClientCOM) and the Predictive Dialer COM API (PDCOM). Both of those APIs are now obsolete. Consequently this client application is no longer supported.

   It is replaced by **Interaction Scripter Client .NET Edition**, a rewrite of Interaction Scripter in C# using the *Interaction Center Extension Library* (IceLib for short). The .NET Edition does not use ClientCOM or PDCOM in any way. It offers more functionality than the Win32 Scripter Client. For example, it supports multi-page base scripts, and uses Session Manager. Interaction Scripter .NET is compatible with all base scripts, and it supports most custom scripts. However, it is not fully backwards compatible with all custom scripts developed using the Interaction Scripter API.

2. On Windows Server 2008 only, if Interaction Scripter .NET is minimized and a new call comes in, the agent does not receive any notification in the tray nor does the window pop to the user. Scripter normally maximizes and enters the foreground when a call is received. But on Windows Server 2008, the Scripter Client will not maximize. This is unlikely to be a significant issue, since agents typically do not run a server operating system on their desktop PCs (DIALER-6650).

3. The ShowDialog command line parameter was removed. Scripter Client always displays the login dialog, unless a -silent command line option is used to suppress the dialog (DIALER-6184).