Genesys®

## Universal Routing 8.1

# Strategy Samples

## About Genesys

Genesys solutions feature leading software that manages customer interactions over phone, Web, and mobile devices. The Genesys software suite handles customer conversations across multiple channels and resources—self-service, assisted-service, and proactive outreach—fulfilling customer requests and optimizing customer care goals while efficiently using resources. Genesys software directs more than 100 million customer interactions every day for 4000 companies and government agencies in 80 countries. These companies and agencies leverage their entire organization, from the contact center to the back office, while dynamically engaging their customers. Go to www.genesyslab.com for more information.

Each product has its own documentation for online viewing at the Genesys Technical Support website or on the Documentation Library DVD, which is available from Genesys upon request. For more information, contact your sales representative.

## Notice

Although reasonable effort is made to ensure that the information in this document is complete and accurate at the time of release, Genesys Telecommunications Laboratories, Inc., cannot assume responsibility for any existing errors. Changes and/or corrections to the information contained in this document may be incorporated in future versions.

## Your Responsibility for Your System's Security

You are responsible for the security of your system. Product administration to prevent unauthorized use is your responsibility. Your system administrator should read all documents provided with this product to fully understand the features available that reduce your risk of incurring charges for unlicensed use of Genesys products.

## Trademarks

## Technical Support from VARs

If you have purchased support from a value-added reseller (VAR), please contact the VAR for technical support.

## Technical Support from Genesys

If you have purchased support directly from Genesys, please contact Genesys Technical Support at the regional numbers provided on page 10. For complete contact information and procedures, refer to the ***Genesys Technical Support Guide***.

## Ordering and Licensing Information

Complete information on ordering and licensing Genesys products can be found in the ***Genesys Licensing Guide.***

## Released by

Genesys Telecommunications Laboratories, Inc. www.genesyslab.com

**Document Version:** 81r_strat_09-2011_v8.1.101.00

Genesys

# Table of Contents

**Supplements**

**Index**

# Preface

Welcome to the *Universal Routing 8.1 Strategy Samples.* The purpose of this document is twofold:

1. To simplify configuration for first-time users of the strategy development tool, Interaction Routing Designer (IRD).
2. To present examples of different types of routing strategies.

To achieve this goal, this document supplies examples of simple routing strategies that can be used as general guidelines during the design stage.

This document is valid only for the 8.1 release(s) of this product.

**Note:** For versions of this document created for other releases of this product, visit the Genesys Technical Support website, or request the Documentation Library DVD, which you can order by e-mail from Genesys Order Management at orderman@genesyslab.com.

This preface contains the following sections:

For information about related resources and about the conventions that are used in this document, see the supplementary material starting on page 215.

## Universal Routing and the CIM Platform

Universal Routing is a part of the Genesys Customer Interaction Management (CIM) Platform, a group of Genesys products that together provide core interaction management functionality.

On its own, Universal Routing provides voice-routing capabilities. When combined with Genesys eServices (called Multimedia in 8.0.0 and earlier), you can also route various types of non-voice media. Universal Routing and

Genesys eServices work together to enable you seamlessly to route both voice and non-voice interactions.

# CIM Platform

The CIM Platform consists of the following:

- Management Framework
- Reporting (CC Analyzer, CCPulse+)
- Interaction Management, which in turn consists of the following:
    - Universal Routing
    - Interaction Workflow
    - Knowledge Management
    - Content Analysis
    - Universal Contact History

On top of the CIM Platform are various media channels. Some, such as Genesys Network Voice, handle traditional telephony. Others, such as Genesys E-mail, handle other media.

## eServices

The CIM Platform can handle various media channels. Genesys eServices (formerly eServices) encompasses those Genesys components that work together to manage interactions whose media is something other than traditional telephonic voice (for example, e-mail or chat).

eServices includes some parts of the Genesys Customer Interaction Management (CIM) Platform, plus certain of the media channels that run on top of the Platform:

- From the CIM Platform, all of Interaction Management except for Universal Routing:
    - Interaction Workflow—Centralized handling of interactions irrespective of media type
    - Knowledge Management—Creation and maintenance of standard responses and screening rules
    - Content Analysis—Optional enhancement to Knowledge Management, applying natural language processing technology to categorize interactions
    - Universal Contact History—Storage of data on contacts and on interactions (linked as threads)
- From the media channels, at least one of the following:
    - Genesys E-mail
    - Genesys Chat (formerly Genesys Web Media)
    - Genesys SMS (Short Message Service)

- Genesys MMS (eServices Messaging Service)
- Genesys Web Callback
- Genesys 3rd Party Media—Ability to add customized support for other media (for example, fax)
- Optionally, Web Collaboration—Ability for agents and customers to co-browse (simultaneously navigate) shared web pages. This is an option that you can add to either Genesys Chat or Inbound Voice.

See Figure 1.

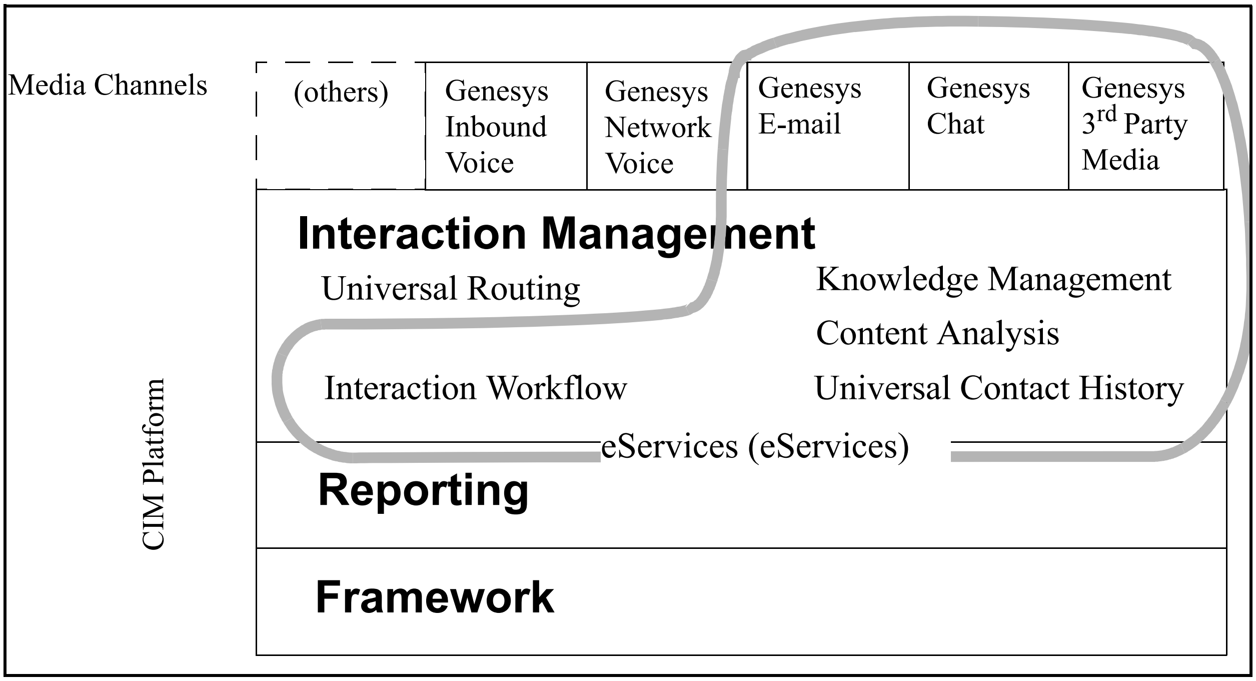


**Figure 1: eServices in Relation to the CIM Platform and Media Channels**

**Note:** Any functioning solution (platform plus channels) that includes any part of the Interaction Management sector requires Universal Routing.

# Intended Audience

This document is primarily intended for first time strategy designers who are responsible for planning strategies and creating them in IRD. This guide assumes that you have a basic understanding of:

- Computer-telephony integration (CTI) concepts, processes, terminology, and applications.
- Network design and operation.
- Your own network configurations.

- Routing concepts and the different types of routing as described in the *Universal Routing 8.1 Deployment Guide*.
- How to use the strategy and business process development tool, Interaction Routing Designer.

You should also be familiar with Genesys Framework architecture and functions, as well as Genesys eServices (if it is installed).

Ideally you will have taken Genesys University routing courses, such as Routing Installation and Configuration, Building Basic Routing Strategies, and Advanced Routing Strategies Workshop.

# Making Comments on This Document

If you especially like or dislike anything about this document, feel free to e-mail your comments to Techpubs.webadmin@genesyslab.com.

You can comment on what you regard as specific errors or omissions, and on the accuracy, organization, subject matter, or completeness of this document. Please limit your comments to the scope of this document only and to the way in which the information is presented. Contact your Genesys Account Representative or Genesys Technical Support if you have suggestions about the product itself.

When you send us comments, you grant Genesys a nonexclusive right to use or distribute your comments in any way it believes appropriate, without incurring any obligation to you.

# Contacting Genesys Technical Support

If you have purchased support directly from Genesys, contact Genesys Technical Support at the following regional numbers:

| Region | Telephone | E-Mail |
|---|---|---|
| North America and<br>Latin America | +888-369-5555 (toll-free)<br>+506-674-6767 | support@genesyslab.com |
| Europe, Middle East, and Africa | +44-(0)-1276-45-7002 | support@genesyslab.co.uk |
| Asia Pacific | +61-7-3368-6868 | support@genesyslab.com.au |
| Before contacting technical support, refer to the *Genesys Technical Support Guide* for complete contact information and procedures. | | |

| Region | Telephone | E-Mail |
|---|---|---|
| Malaysia | 1-800-814-472 (toll-free in Malaysia)<br>+61-7-3368-6868 (toll) | support@genesyslab.com.au |
| India | 000-800-100-7136 (toll-free)<br>+61-7-3368-6868 (International) | support@genesyslab.com.au |
| Japan | +81-3-6361-8950 | support@genesyslab.co.jp |
| Before contacting technical support, refer to the *Genesys Technical Support Guide* for complete contact information and procedures. | | |

# Document Change History

This is the first release of the *Universal Routing 8.1 Strategy Samples*. In the future, this section will list topics that are new or that have changed significantly since the first release of this document.

Genesys

Chapter

# 1 Samples and Strategies Overview

This chapter provides a brief description of the two types of samples: routing strategy samples provided with Universal Routing, and business process samples provided with Genesys eServices (formerly eServices).

**Note:** Detailed instructions for installing these samples are located in the *Universal Routing 8.1 Deployment Guide* (for strategy samples) and the *eServices (eServices) 8.0 Deployment Guide* (for the Interaction Workflow Samples).

This chapter also presents a high-level description of routing strategies in preparation for understanding the strategy samples. The information in this chapter is divided into the following topics:

For more detailed information on routing strategies and IRD objects, see the *Universal Routing 8.1 Reference Manual*, the *Universal Routing 8.1 Business Process User's Guide*, and the *Universal Routing 8.1 Interaction Routing Designer Help*.

# About the Samples

This section provides a brief description about the two types of samples: routing strategy samples and business process samples.

## Routing Strategy Samples

This document includes a number of voice routing strategy samples that illustrate frequently-used functions. The samples, presented in Chapter 2, “Voice Routing Strategy Samples,” on page 37, are the following:

- Strategy 1: Load Balancing Among Queues, page 37
- Strategy 2: Percentage Allocation Among Targets, page 40
- Strategy 3: Routing to Agent Groups, page 43
- Strategy 4: Skills-Based Routing, page 46
- Strategy 5: Database Lookup, page 51
- Strategy 6: Using IVR Data, page 58
- Strategy 7: Using Subroutines, page 64

### Additional Sample Strategies

There are three additional sources of sample strategies to assist you in developing your own strategies.

#### IRD Samples

The Interaction Routing Designer (IRD) installation process places two `*.zcf` files, `Samples.zcf` and `RLU.zcf,` in the IRD installation directory. Importing the `*.zcf` files into IRD provides you with sample voice strategies, subroutines, and list objects that appear in, and can be opened and edited from, IRD.

**Note:** For detailed installation and import instructions, see the “Samples” chapter in the *Universal Routing 8.1 Deployment Guide*.

#### URS Sample

When you install URS, you are provided with a sample, `OutboundMultiCampaign.ooo`, which is pre-written strategy byte code designed to route outbound calls/interactions to Campaign Groups. For additional information about this sample and how to use it, see the “Samples” chapter in the *Universal Routing 8.1 Deployment Guide*.

### Sample Strategies in the IRD Help

In addition, a number of strategy samples are provided and discussed in the *Universal Routing 8.1 Interaction Designer Help*:

- Sample A: DNIS, Load Balancing, Percentage Allocation, Default Route
- Sample B: Day of Week, Time, Routing Rules, IVR
- Sample C: Date, Day of Week, Time, JumpToStrategy
- Sample D: CED, Variables, Attached Data, Skill-based Routing, Generic Segmentation
- Sample E: Attributes, Business Rules, Database Lookup

## Business Process Samples

Genesys eServices includes the Interaction Workflow Samples component. After installing this component, you can access the sample business processes in the IRD Interaction Design window.

Table 1 on page 15 lists the business processes included in the Interaction Workflow Samples. These business processes are described in Chapter 3, “Multimedia Samples,” on page 73.

You can learn more about business processes and about these samples in the *Universal Routing 8.1 Business Process User’s Guide* and in the *Universal Routing 8.1 Interaction Routing Designer Help*.

**Table 1: Business Processes Provided in the Interaction Workflow Samples**

| Business process name | Description |
|---|---|
| ABC Simple BP | A basic business process useful to get the general concept of business processes and check the samples installation. |
| ABC Simple Chat BP | A simple business process that demonstrates basic chat interaction processing. |
| ABC Simple MMS | A simple business process that demonstrates basic inbound MMS (Multimedia Messaging Service) processing. |
| ABC Simple SMS Paging | A simple business process that demonstrates basic SMS (Short Messaging Service) processing. |
| WebCallback | A business process that provides web callback functionality. |
| Default BP | A complex business process that incorporates many types of interaction handling. |

**Table 1: Business Processes Provided in the Interaction Workflow Samples (Continued)**

| Business process name | Description |
|---|---|
| Ten samples beginning with *Step 0. Common Components* and ending with *Step 4*. | A breakdown of Default BP into functionally-themed components. |
| How to: apply escalation procedure | |
| How to: attach classification categories and use the Attach Categories object. | |
| How to: attach classification categories and use the Multi-Screen object. | |
| How to: get credit card number from the e-mail. | |
| How to: handle fax interactions. | |
| How to: identify contact and create interaction in UCS. | |
| How to: place the interaction in the workbin. | |
| How to: screen with multiple rules and use the screening switch. | |

# Routing Defined

In the context of telecommunications, routing is the process of sending an interaction to a target, for example, sending an incoming telephone call to an agent.

In practice, many steps must be taken between the arrival of an interaction and the selection and use of a routing target.

Not all interactions should go to the same target; choices must be made in order to determine the best target for each interaction.

Each choice-point is an opportunity to make a decision based on the current situation with the goal of getting the interaction delivered to the right target.

# What Are Routing Strategies?

Routing strategies are an integral part of Genesys Universal Routing. They are created in an environment called Interaction Routing Designer (IRD) where you also test, modify, and load routing strategies. Universal Routing Server (URS or “Router”) is the server that executes routing strategy instructions.

A routing strategy instructs URS how to handle and where to direct interactions under different circumstances. Figure 2 shows a simple voice call routing strategy that routes based on caller-entered digits (CEDs).

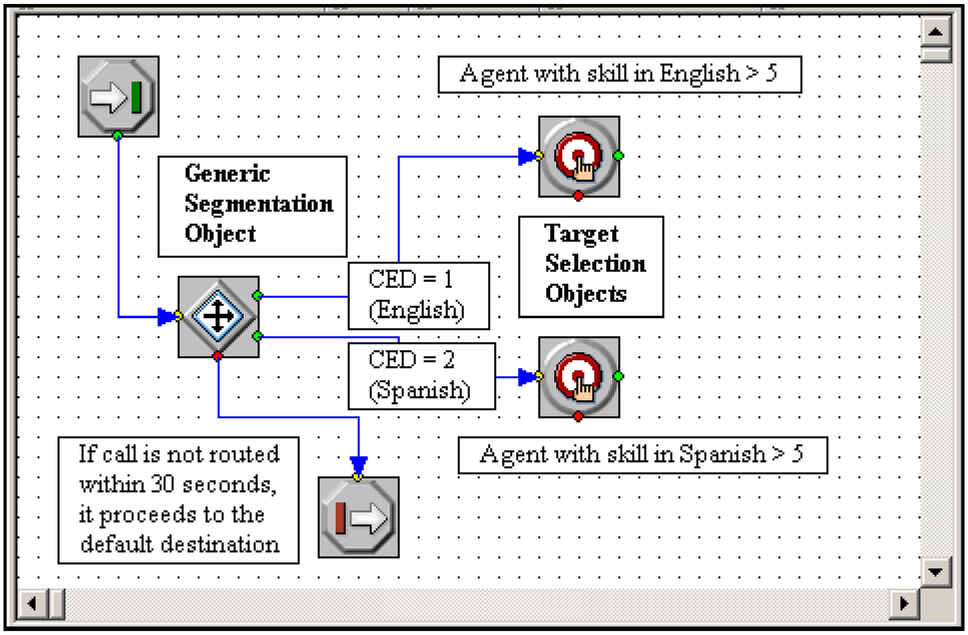

Agent with skill in English > 5
Generic Segmentation Object
Target Selection Objects
CED = 1 (English)
CED = 2 (Spanish)
If call is not routed within 30 seconds, it proceeds to the default destination
Agent with skill in Spanish > 5

Figure 2: Example Voice Routing Strategy

### About the Graphics

Some graphics showing business processes and strategies in this document have been rearranged to make all the included objects fit in the available space. When actually creating business processes and strategies, Genesys recommends that you allow adequate space between the objects for clarity. The actual workspaces provided by IRD are much larger than the areas shown in these graphics.

## Choice-Points

A strategy is a structured set of choice-points, each of which analyzes some aspect of the current interaction. This can include facts related to the interaction itself, the customer initiating the interaction, the state of the contact center, the particular point in time, etc.

At any given choice-point, only one of several possible outcomes can be true. URS determines which outcome is true and sends the interaction along a specified route accordingly. The specified route may be another object in the routing strategy or it may be the final target.

## Objects and Ports

Typically, a choice-point is represented graphically in a strategy by an *object* with one yellow entry port, one red error port, and one or more green exit ports. For example, the second object in Figure 2 on page 17, is a Generic Segmentation object. It causes incoming interactions to take different paths in the strategy. This particular instance of the Generic Segmentation object has a yellow input port, a red error port, and two green exit ports (see Figure 3).

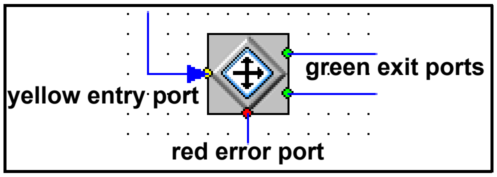


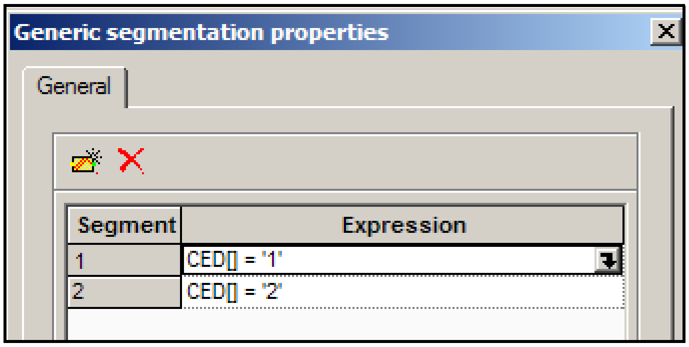
**Figure 3: Generic Segmentation Object Ports**

The object, which routes based on the true/false value of an expression, uses the ports as follows: If the expression is true, the interaction goes out the green exit port to the next object; if the expression is false, the interaction goes out the red error port to the next object.

**Note:** You can customize where the ports appear on an object. For more information, see *Universal Routing 8.1 Interaction Routing Designer Help*.

## Object Properties

There are two exit ports in Figure 3 because the strategy designer anticipated two possible caller-entered digits (CEDs) attached to an interaction: `1` or `2`. You can see these digits in the properties dialog box that opens when you double-click the Generic Segmentation object (see Figure 4).



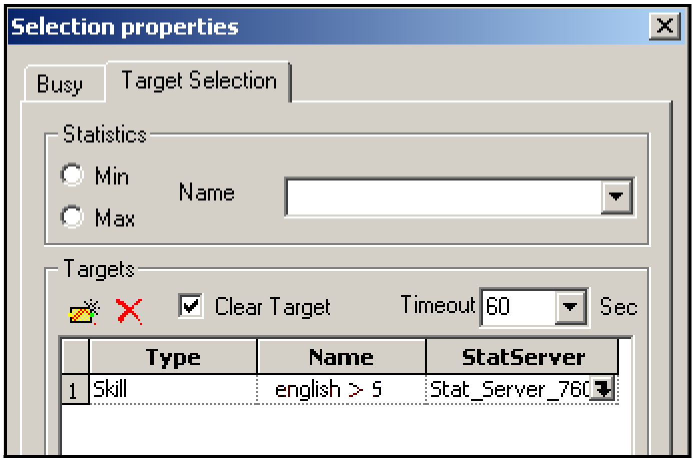
| Segment | Expression |
|---|---|
| 1 | CED[] = '1' |
| 2 | CED[] = '2' |

**Figure 4: Segmenting on Caller-Entered Digits**

**Note:** Instances of the Generic Segmentation object in other strategies may have more than two exit ports. For an example, see Figure 117 on page 104.

Each expression in Figure 4 on page 18 was created in the Expression Builder (see Figure 28 on page 35). The Expression Builder opens when you click in a row under the `Expression` heading (shown in Figure 4) and then click the resulting down arrow.

Each output port in Figure 2 on page 17 goes to a different target Selection object. Figure 5 shows the properties dialog box that opens when you double-click the Selection object to which the upper green port leads.



○ Min

○ Max

Name:

☑ Clear Target

Timeout 60 Sec

|  | Type | Name | StatServer |
|---|---|---|---|
| 1 | Skill | english > 5 | Stat_Server_760 |

**Figure 5: Routing Based on Skill Level**

The Target Selection tab indicates that the call will be routed to an agent having the `Skill` called `english` with a skill level greater than `5`.

In addition to `Skill` targets, you can also route calls to an `ACD Queue, Agent, Agent Group, Destination Label, Place, Place Group, Queue Group, Routing Point,` and `Variable`.

# Types of Objects

Strategies are built and tested using two different types of objects and connections in the IRD graphical interface:

- Strategy-building objects, such as the Segmentation and Routing objects just discussed.
- Re-usable objects.

So you can better understand the samples presented in this book, the next section contain a basic explanation of re-usable and strategy building objects.

# Re-Usable Objects

With IRD, you can define the following re-usable objects and data, which any strategy can then use:

- Strategies
- Subroutines
- Routing Rules
- Business Rules
- Attributes
- Interaction Data
- Statistics
- Schedules
- List Objects
- Macros

Figure 6 shows the buttons for re-usable objects, which are located in the Routing Design shortcut bar on the left side of the IRD main window (not all buttons shown).

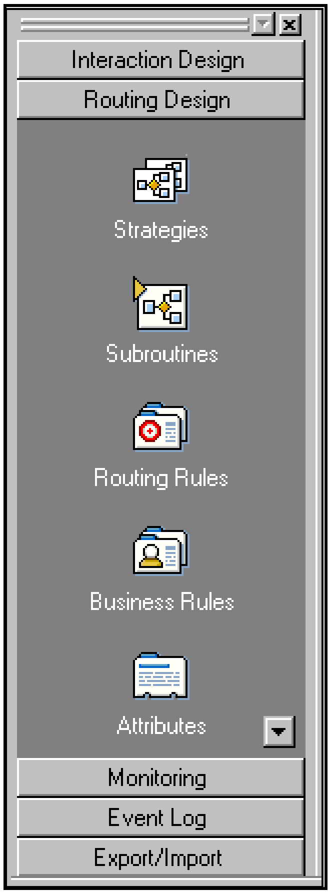


**Figure 6: IRD Interface, Re-Usable Objects**

## Strategies

A strategy is a set of decisions and instructions that instruct URS how to handle and where to direct interactions under different circumstances.

Within a strategy, you may use subroutines, routing rules, business rules, attributes, interaction data, statistics, lists, and macros. After creating a strategy you *load* it. URS then uses the loaded strategy to route interactions on the specified switch and routing points.

- To load a routing strategy, go to IRD Monitoring view, click the `Loading` (or `Group Loading`) button, select the switch and routing point, right-click, and then select `Load Strategy` from the shortcut menu.

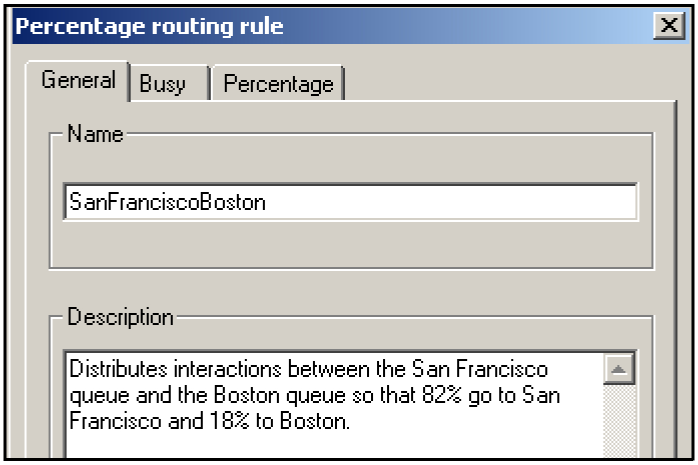
**Note:** For information about how to load the strategies that are included in a business process, which is a collection of strategies that together direct the workflow for a eServices interaction, see the *Universal Routing 8.1 Business Process User's Guide*.

## Subroutines

*Subroutines* are strategies called from within a strategy or another subroutine. Like other strategies, subroutines can contain any IRD objects or functions. Like other re-usable objects, subroutines can be accessed by any strategy. See "Strategy 7: Using Subroutines" on page 64 for an example.

## Routing Rules

*Routing rules* specify the method of target selection: force, service level, load balancing, percentage, statistics, switch-to-strategy, or workforce. Many Routing objects (see page 30) use routing rules in their properties. You can configure routing rules once and then reuse them in multiple Routing objects and strategies. Figure 7 shows an example routing rule that routes based on percentage allocation.



Name

SanFranciscoBoston

Description

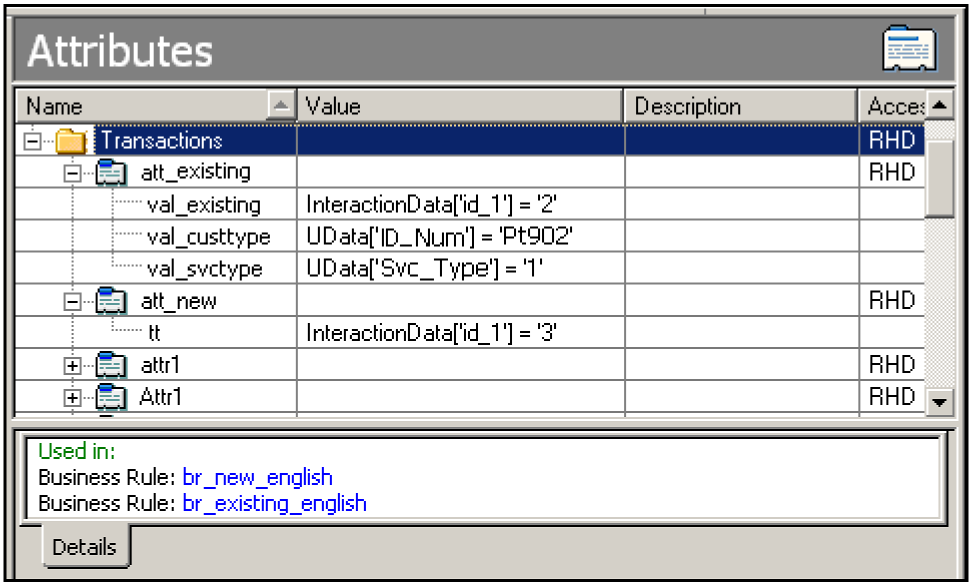
Distributes interactions between the San Francisco queue and the Boston queue so that 82% go to San Francisco and 18% to Boston.

**Figure 7: Example Routing Rule**

See “Strategy 1: Load Balancing Among Queues” on page 37 and “Strategy 2: Percentage Allocation Among Targets” on page 40 for sample strategies that use routing rules.

## Attributes

Used mostly by voice strategies, *attributes* are pieces of interaction or customer data, along with all their possible values. You use attributes in order to create *business rules* (discussed next). Figure 8 shows example attributes.



| Name | Value | Description | Acces |
|---|---|---|---|
| Transactions | | | RHD |
| att_existing | | | RHD |
| val_existing | InteractionData['id_1'] = '2' | | |
| val_custtype | UData['ID_Num'] = 'Pt902' | | |
| val_svctype | UData['Svc_Type'] = '1' | | |
| att_new | | | RHD |
| tt | InteractionData['id_1'] = '3' | | |
| attr1 | | | RHD |
| Attr1 | | | RHD |

Used in:
Business Rule: br_new_english
Business Rule: br_existing_english

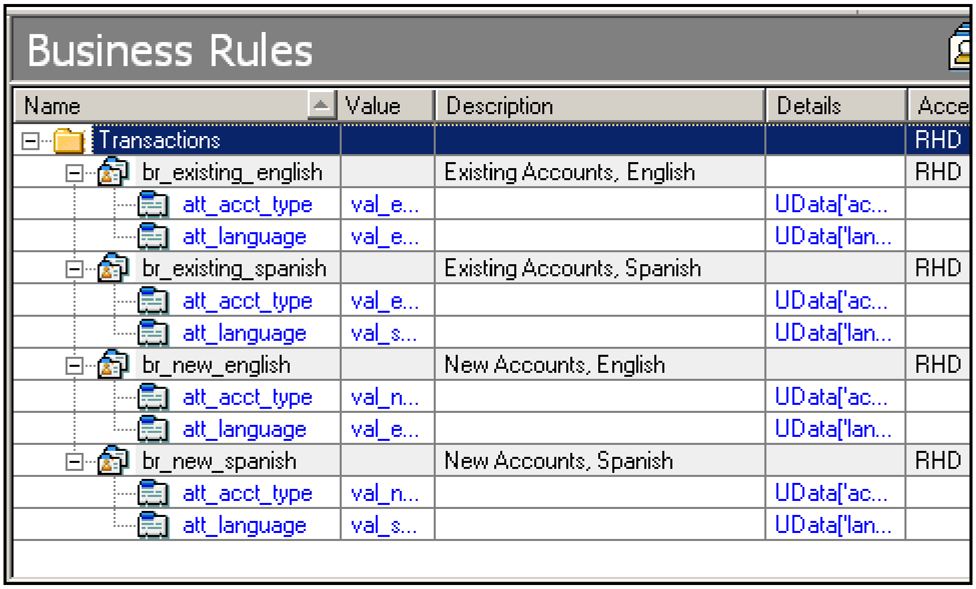
**Figure 8: Example Attributes**

“Strategy 4: Skills-Based Routing” on page 46 uses attributes.

**Note:** Do not confuse attributes that you define in IRD with `Business Attributes` that you define in Configuration Manager (see “E-mail Accounts Business Attributes” on page 111).

## Business Rules

*Business rules* are created from attributes (see Figure 9).



Business Rules

| Name | Value | Description | Details | Acce |
|---|---|---|---|---|
| Transactions | | | | RHD |
| br_existing_english | | Existing Accounts, English | | RHD |
| att_acct_type | val_e... | | UData['ac... | |
| att_language | val_e... | | UData['lan... | |
| br_existing_spanish | | Existing Accounts, Spanish | | RHD |
| att_acct_type | val_e... | | UData['ac... | |
| att_language | val_s... | | UData['lan... | |
| br_new_english | | New Accounts, English | | RHD |
| att_acct_type | val_n... | | UData['ac... | |
| att_language | val_e... | | UData['lan... | |
| br_new_spanish | | New Accounts, Spanish | | RHD |
| att_acct_type | val_n... | | UData['ac... | |
| att_language | val_s... | | UData['lan... | |

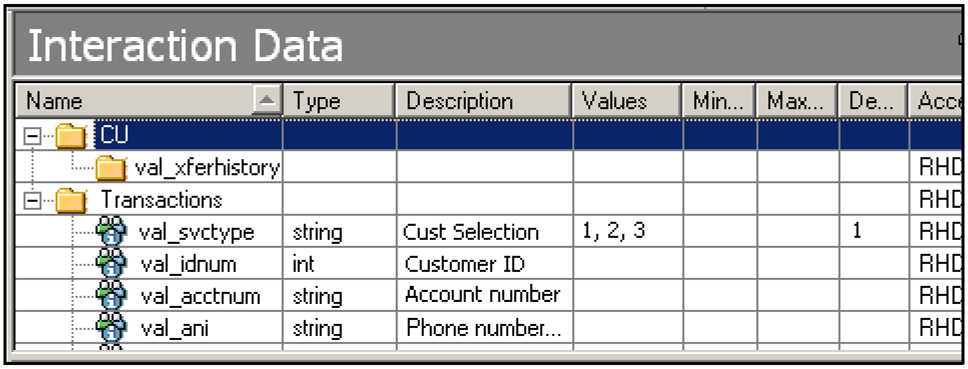
**Figure 9: Example Business Rules Created From Attributes**

Business rules and attributes enable you to create logical expressions for segmenting interactions to take different paths in a strategy. The advantage of using business rules and attributes is that you can reuse the same business decisions multiple times in the same strategy and in many different strategies.

When you use business rules and attributes, you use a Business Segmentation object (see Figure 44 on page 48). For a sample strategy that uses business rules, see "Strategy 4: Skills-Based Routing" on page 46.

## Interaction Data

*Interaction data* defines attached data keys used in a strategy (see Figure 10).



Interaction Data

| Name | Type | Description | Values | Min... | Max... | De... | Acce |
|---|---|---|---|---|---|---|---|
| CU | | | | | | | |
| val_xferhistory | | | | | | | RHD |
| Transactions | | | | | | | RHD |
| val_svctype | string | Cust Selection | 1, 2, 3 | | | 1 | RHD |
| val_idnum | int | Customer ID | | | | | RHD |
| val_acctnum | string | Account number | | | | | RHD |
| val_ani | string | Phone number... | | | | | RHD |

**Figure 10: Example Interaction Attributes**

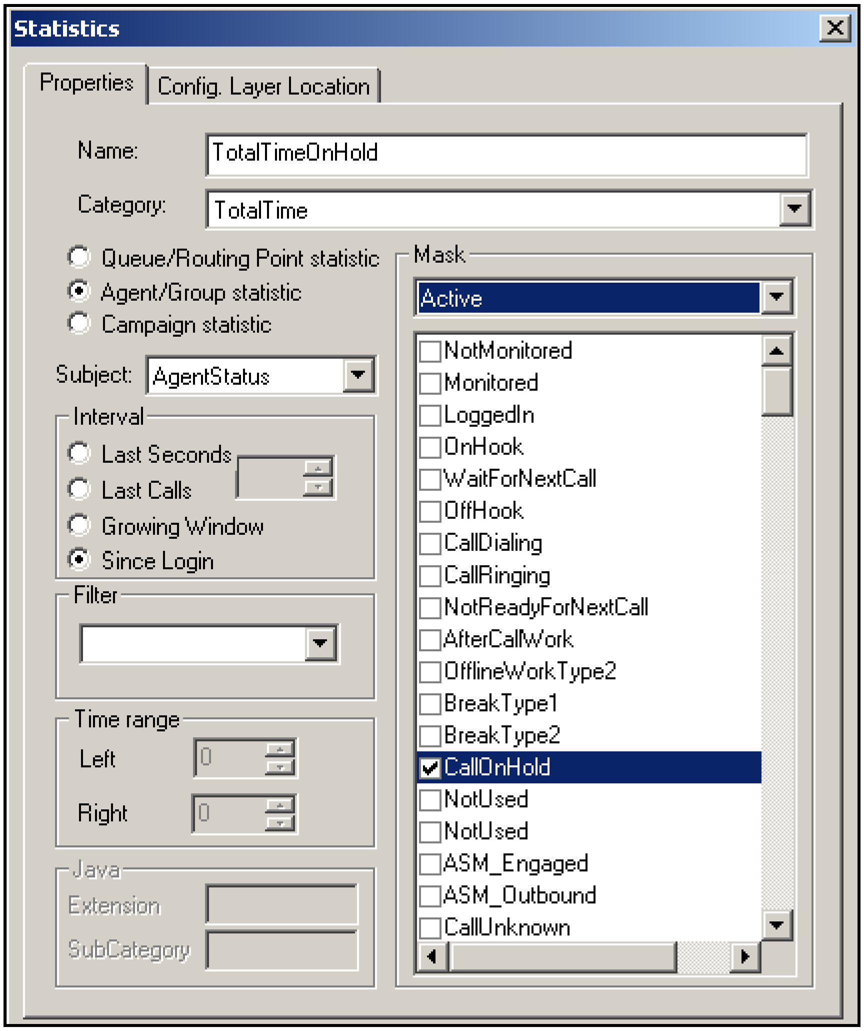
You define interaction data in order to enable strategies to gather attached data from event messages, such as that which is used in screen pops. You also can use interaction data in the Assign, Multi-Assign (see Figure 104 on page 94), If (see Figure 105 on page 95), and Function objects (see Figure 48 on page 50).

# Statistics

You can route interactions based on the value of a statistic. You can also use a statistic to determine the optimal target if more than one target is available.

To configure a new statistic:

1. Click the Statistics button on the IRD main window shortcut bar (shown in Figure 6 on page 20).
2. Select File > New. The Statistics Properties dialog box opens (shown in Figure 11).



Name: TotalTimeOnHold

Category: TotalTime

○ Queue/Routing Point statistic
● Agent/Group statistic
○ Campaign statistic

Subject: AgentStatus

Interval
○ Last Seconds
○ Last Calls
○ Growing Window
● Since Login

Filter

Time range
Left 0
Right 0

Java
Extension
SubCategory

Mask
Active

□ NotMonitored
□ Monitored
□ LoggedIn
□ OnHook
□ WaitForNextCall
□ OffHook
□ CallDialing
□ CallRinging
□ NotReadyForNextCall
□ AfterCallWork
□ OfflineWorkType2
□ BreakType1
□ BreakType2
☑ CallOnHold
□ NotUsed
□ NotUsed
□ ASM_Engaged
□ ASM_Outbound
□ CallUnknown

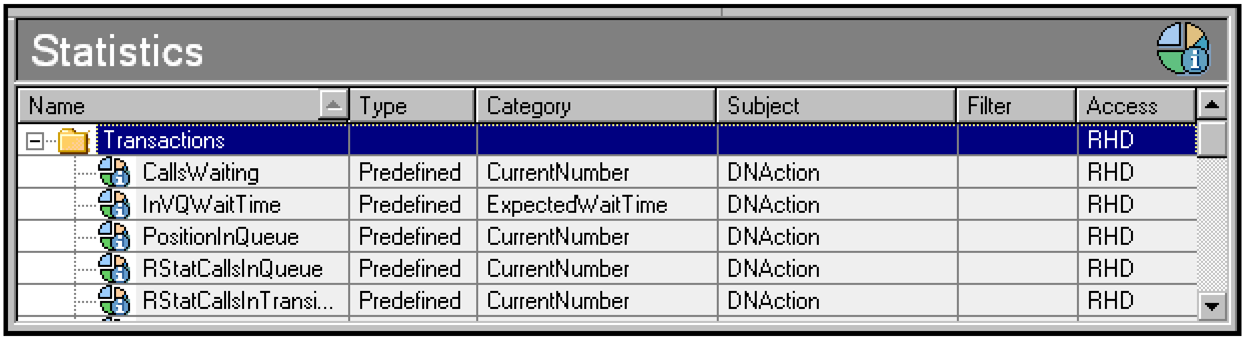
**Figure 11: Statistics Properties Dialog Box**

Click the appropriate button (see Figure 11) to define one of the following types of statistics:

- Queue/Routing Point
- Agent/Group
- Campaign

Once you have defined a statistics object, it becomes available for use in the properties dialog box that is associated with the Selection, Service Level, and Statistics objects.

User-defined statistics appear along with IRD predefined statistics in the Statistics list on the IRD main window when you click the Statistics icon on the IRD main window shortcut bar (see Figure 12).



Statistics

| Name | Type | Category | Subject | Filter | Access |
|---|---|---|---|---|---|
| Transactions | | | | | RHD |
| CallsWaiting | Predefined | CurrentNumber | DNAction | | RHD |
| InVQWaitTime | Predefined | ExpectedWaitTime | DNAction | | RHD |
| PositionInQueue | Predefined | CurrentNumber | DNAction | | RHD |
| RStatCallsInQueue | Predefined | CurrentNumber | DNAction | | RHD |
| RStatCallsInTransi... | Predefined | CurrentNumber | DNAction | | RHD |

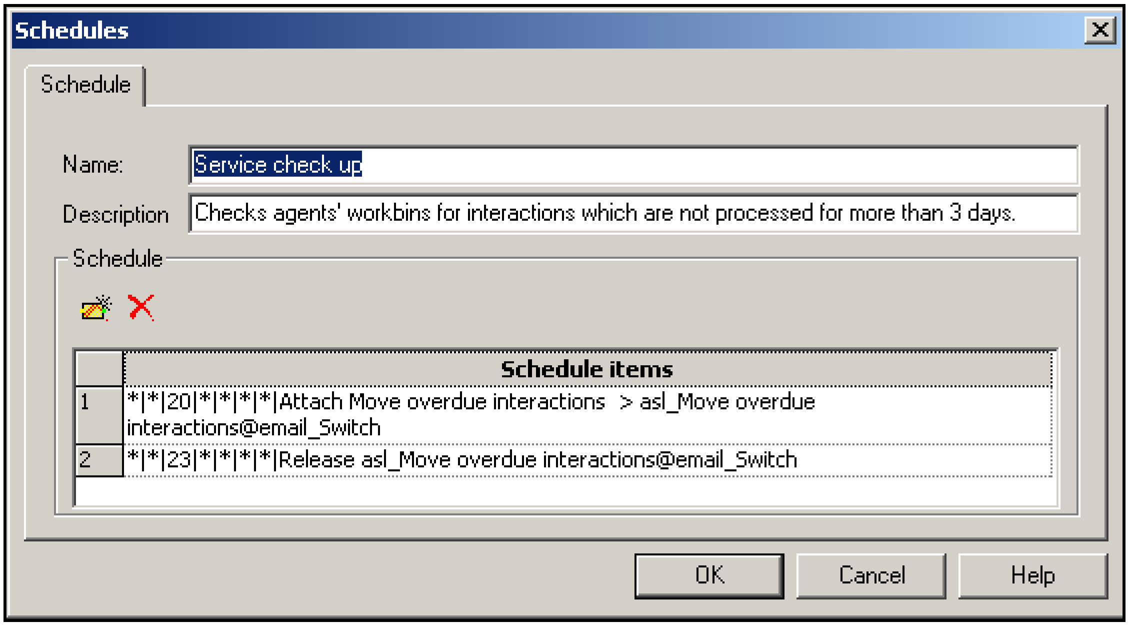
**Figure 12: IRD Predefined Statistics**

**Note:** If you are in Check Integrity mode and the statistic row appears green, this indicates that the statistic is used by a loaded routing strategy.

## Schedules

A *schedule* instructs URS when to load and release a routing strategy automatically. A schedule definition also includes the switch and DNs on which to load the strategy (see Figure 13).

You can specify times for loading and releasing schedules by using seconds, minutes, hours, dates, months, years, and days.



Name: Service check up

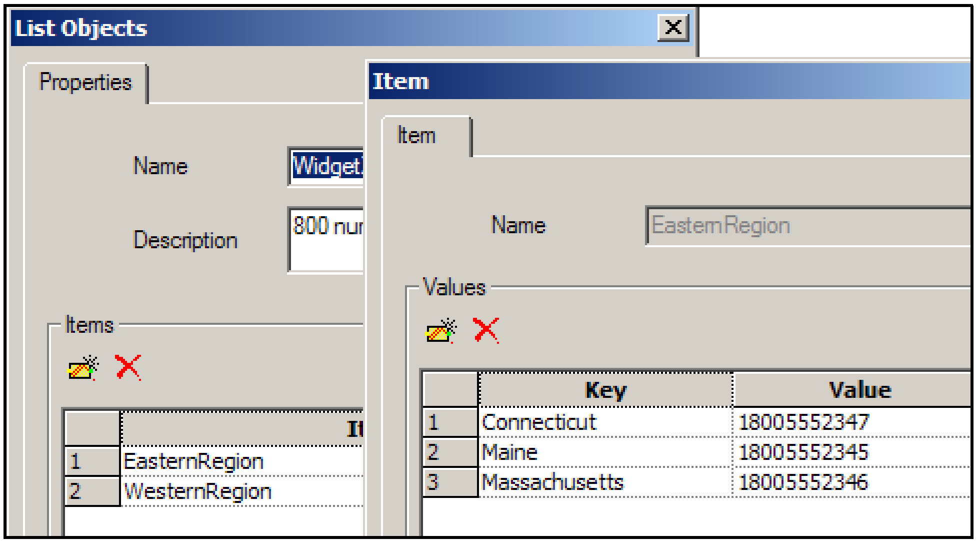
Description: Checks agents' workbins for interactions which are not processed for more than 3 days.

<table>
<tr><th></th><th>Schedule items</th></tr>
<tr><td>1</td><td>*|*|20|*|*|*|*|Attach Move overdue interactions &gt; asl_Move overdue interactions@email_Switch</td></tr>
<tr><td>2</td><td>*|*|23|*|*|*|*|Release asl_Move overdue interactions@email_Switch</td></tr>
</table>

Figure 13: Example Schedule

## Lists

A *list* re-usable object contains strings of any nature (for example, DNIS or ANI strings), which can be used in strategies and are included in integrity checking (see Figure 14).



Name: Widget

Description: 800 nur

| | It |
|---|---|
| 1 | EasternRegion |
| 2 | WesternRegion |

Name: EasternRegion

| | Key | Value |
|---|---|---|
| 1 | Connecticut | 18005552347 |
| 2 | Maine | 18005552345 |
| 3 | Massachusetts | 18005552346 |

Figure 14: Example List Object

For example, you might use a list of toll-free numbers in a strategy. Instead of referencing each individual toll-free number in the strategy, you can logically group numbers together and name the group. Then, when you need to add or edit numbers, you do not need to edit the strategy; you just add to or edit the list object.

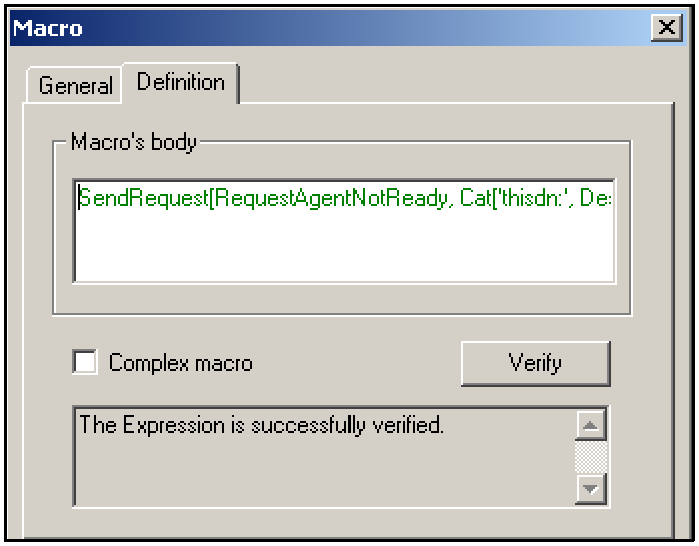
## Macros

A *macro* object enables you to combine several objects and expressions into one re-usable block. It works rather like a user-defined function.

You can define both simple and complex macros. The difference between the two is as follows:

- A simple macro is a sequence of actions, one after the other, usually (but not necessarily) performed by IRD objects.
- A complex macro uses a condition, which can be defined in the Expression Builder, to determine whether to perform an action. If the condition is true, an action is performed, such as executing a function call.

Figure 15 shows the `Macro's body` area for the predefined macro `MakeAgentNotReady`.



SendRequest[RequestAgentNotReady, Cat['thisdn:', De:

□ Complex macro

The Expression is successfully verified.

**Figure 15: Predefined Macro Example**

Once a macro is defined, you place the Macro object in a strategy and select the macro you created from the list of available macros. You do not have to repeatedly enter the same set of objects and/or parameters.

IRD supplies the following predefined macros:

- `ComplexSample`
- `MakeAgentNotReady` (uses the SendRequest function)
- `RedirectCall` (uses the SendRequest function)
- `RedirectCallMakeNotReady` (uses the SendRequest function)
- `SimpleSample`
- `DelimitTargetList`

If your switch and T-Server support it, you can use three of the predefined macros and the SendRequest function (along with Genesys T-Library functions) to handle ring-no-answer situations at the strategy level.

**Note:** For more information, see the SendRequest function in the *Universal Routing 8.1 Reference Manual*.

# Strategy-Building Objects

In addition to the re-usable objects just described, strategies are also built using strategy-building objects.

## Object Categories

IRD provides different categories of strategy-building objects. In toolbar order, the categories are:

- **Voice Treatment**—To apply busy treatments and mandatory actions, such as a recorded announcement.
- **Data and Services**—For searching the database for information, requesting an external service, or accessing a Web service.
- **Segmentation**—For causing incoming interactions to take different paths (typically the second type of object in a strategy).
- **Routing**—For routing target selection (sometimes the last type of object placed in a strategy).
- **Miscellaneous**—To perform miscellaneous operations.
- **eServices (Multimedia)**—To control how eServices interactions should be processed. eServices consists of various types of non-voice interactions, such as e-mail, chat, and Open Media.
- **Outbound**—To process `outbound_preview` Open Media interactions using only routing strategies, without any need for agent handling.
- **Workflow and Resource Management**—To control certain settings for strategies, Places, and Agents.
- **SMS—**For creating and sending SMS messages.

As you can see in the example in Figure 2 on page 17, a strategy is made up of objects from the various categories of strategy-building objects along with the connections between those objects.

All strategies have an Entry object. After the Entry object, the subsequent strategy-building objects control the actions that URS performs on each interaction. The connections that you make between objects set the sequence in which each object is executed. When handling an interaction, URS selects only one connection for an interaction to follow after it is handled by an object.

Icons for each object category appear on the objects toolbar when you create or edit a strategy or subroutine in the IRD Routing Design window (see Figure 16).

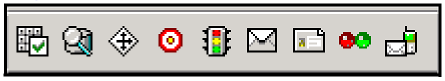


**Figure 16: Icons on the Objects Toolbar**

When you click an icon on the toolbar representing an object category, buttons for all objects belonging to that category come into view. For example, if you click the third icon, buttons for all Segmentation objects drop into view (see Figure 17).

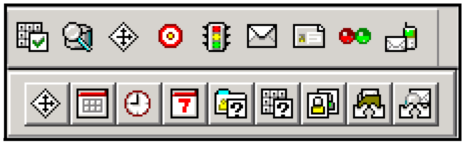


**Figure 17: Dropdown Buttons for Segmentation Objects**

If you were creating a new strategy and wanted to segment incoming interactions to take different paths in the strategy, you would:

1. Click the applicable button on the Segmentation drop-down toolbar.
2. Click in the Routing Design window workspace to place the object in the strategy.
3. Double-click the Segmentation object to open its properties dialog box.
4. Configure properties in the dialog box.
5. Connect the Segmentation object to another object; for example, a Routing object or a Miscellaneous object.

# Buttons Associated with Toolbar Icons

Clicking an icon on the objects toolbar drops down a subtoolbar containing icons for objects you can place in your strategy. The next section describes the buttons associated with each object category.

## Buttons for Voice Treatment Objects

Figure 18 shows the buttons that come into view when you click the Voice Treatments icon.

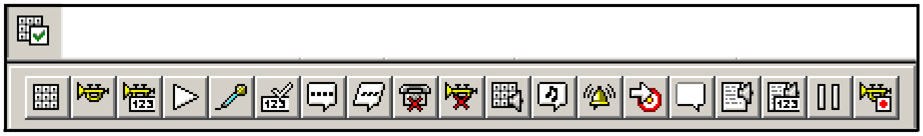


**Figure 18: Buttons for Voice Treatment Objects**

Voice Treatment objects specify an action to be performed on the current call, such as playing music for the caller.

The Voice Treatment objects from left to right are: Collect digits, Play announcement, Play announcement and collect digits, Play application, Record user announcement, Verify digits, Busy, Fast busy, Cancel call, Delete user announcement, IVR, Music, Ringback, Set default destination, Silence, Text to speech, Text to speech and collect digits, Pause, RAN (Play recorded announcement).

## Buttons for Data and Services Objects

Clicking the Data and Services icon opens a subtoolbar containing three buttons (see Figure 19).

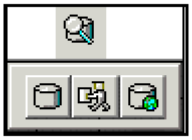


**Figure 19: Data and Services Objects**

## Buttons for Segmentation Objects

Clicking the third button on the objects toolbar in reveals additional buttons for various types of Segmentation objects. A Segmentation object is usually the second object in a strategy, following immediately after the Entry object (see Figure 16 on page 29). Figure 20 shows the buttons for Segmentation objects that come into view when you click the Segmentation icon.

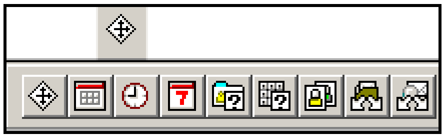


**Figure 20: Buttons for Segmentation Objects**

You might segment interactions based on the identity of the customer as defined by their revenue potential. Or you may choose to segment interactions based on time, date, week day of customer contact, customer phone number (ANI), number dialed (DNIS), business rule, or classification code.

The buttons from left to right are: Generic (brings up Expression Builder), Date, Time, Day of the Week, ANI (originating phone), DNIS (number dialed), Business (series of logical expressions called business rules), Classify, and Screen.

## Buttons for Routing Objects

Clicking the fourth button on the objects toolbar reveals additional buttons for various types of Routing objects (see Figure 21).

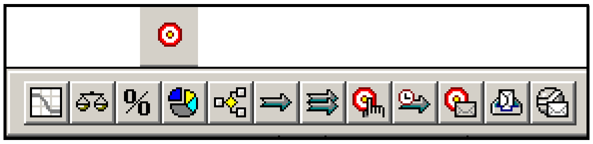


**Figure 21: Buttons for Routing Objects**

Routing objects specify a routing action to be performed with the current interaction, such as connecting the caller to a specific agent group.

The buttons from left to right are: Service level, Load balancing, Percentage, Statistics, Switch-to-Strategy, Default, Force, Selection, Workforce, Route Interaction, Workbin, and Queue Interaction.

## Buttons for Miscellaneous Objects

Clicking the fifth button on the objects toolbar reveals additional buttons for various types of Miscellaneous objects (see Figure 22).

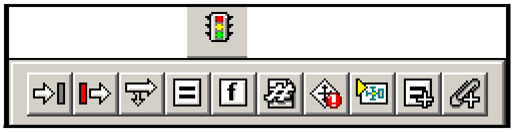


**Figure 22: Buttons for Miscellaneous Objects**

Miscellaneous objects are used for flow control or performing operations, such as executing a function or directing interactions based on an IF statement.

The buttons from left to right are: Entry, Exit, If, Assign, Function, Macro, Error Segmentation, Call Subroutine, MultiAssign, MultiAttach.

## Buttons for eServices Objects

Clicking the sixth button on the objects toolbar reveals additional buttons for eServices (formerly called Multimedia) objects (see Figure 23).

**Note:** Unless the object name specifies that it is designed to be used for e-mail only, eServices objects can be used in strategies for all types of non-voice interactions.

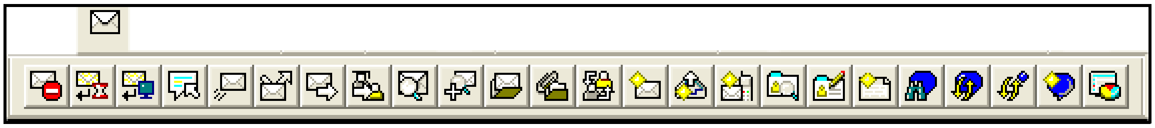


**Figure 23: Buttons for eServices Objects**

The buttons from left to right are: Stop Interaction, Acknowledgement, Autoresponse, Chat Transcript, Send E-Mail, Redirect E-Mail, Forward E-Mail, Reply E-Mail From External Resource, Screen, MultiScreen, Classify, Attach Categories, Create Interaction, CreateEmailOut, CreateNotification, CreateSMS, Identify Contact, Update Contact, Render Message Content, Find Interaction, Update Interaction, Update UCS Record, Submit New Interaction, and Distribute Custom Event.

## Buttons for Outbound Objects

Clicking the Outbound icon on the toolbar opens the Outbound objects subtoolbar (shown in Figure 24). The Outbound objects enable you to process `outbound_preview` Open Media interactions using only routing strategies, without any need for agent handling.

**Note:** You cannot use the Outbound objects to route interactions that are not of the `outbound_preview` media type.

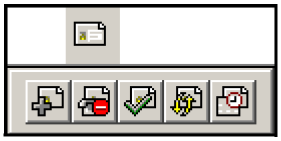


**Figure 24: Buttons for Outbound Objects**

The buttons from left to right are: Add Record, Do Not Call, Processed, Update, and Reschedule.

## Buttons for Workflow and Resource Management Objects

Clicking the eighth button (Workflow and Resource Management) on the objects toolbar reveals additional strategy-building objects that enable you to control certain settings for strategies, Places, and Agents (see Figure 25).

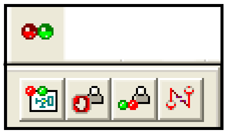


**Figure 25: Buttons for Workflow and Resource Management Objects**

The buttons from left to right are: Set Multimedia Strategy State, Set Agent DND State, Set Agent Media State, and Force Logout.

### Buttons for SMS Objects

Clicking the ninth button (SMS Objects) on the objects toolbar reveals buttons used for creating and sending SMS messages (see Figure 26).

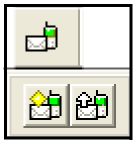


Figure 26:  Buttons for SMS Objects

The buttons from left to right are: Create SMS Out and Send SMS Out.

### Comment Object

A Comment object (shown in Figure 2 on page 17) is used to annotate a strategy in the Routing Design window. Text entered appears on the workspace. You can insert a Comment object, which appears as a text box, anywhere in the strategy.

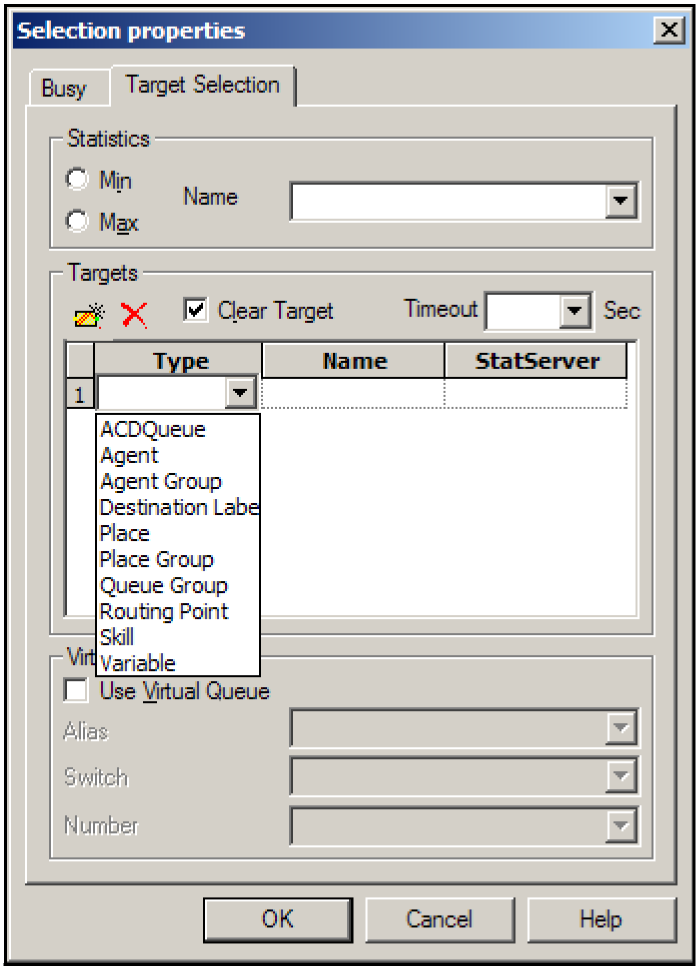
## Object Properties Dialog Boxes

Nearly every IRD strategy-building object has a set of properties or parameters that control the behavior of a portion of the strategy. The exceptions to this are, Entry, Exit, Default Route, Treatment Cancel Call, and a few Function objects.

After you place a strategy-building object is placed in the Routing Design window workspace, double-click it to open the properties dialog box. Here you can assign values to the object's parameters. The content of the properties dialog box varies depending on the object.

Parameters are criteria defining how the interaction is handled. The parameters can include how long an interaction should be held before being sent to a default extension, its priority in queue, and so forth.

Figure 27 shows an example properties dialog box for the Routing Selection object, which is used for routing voice interactions:

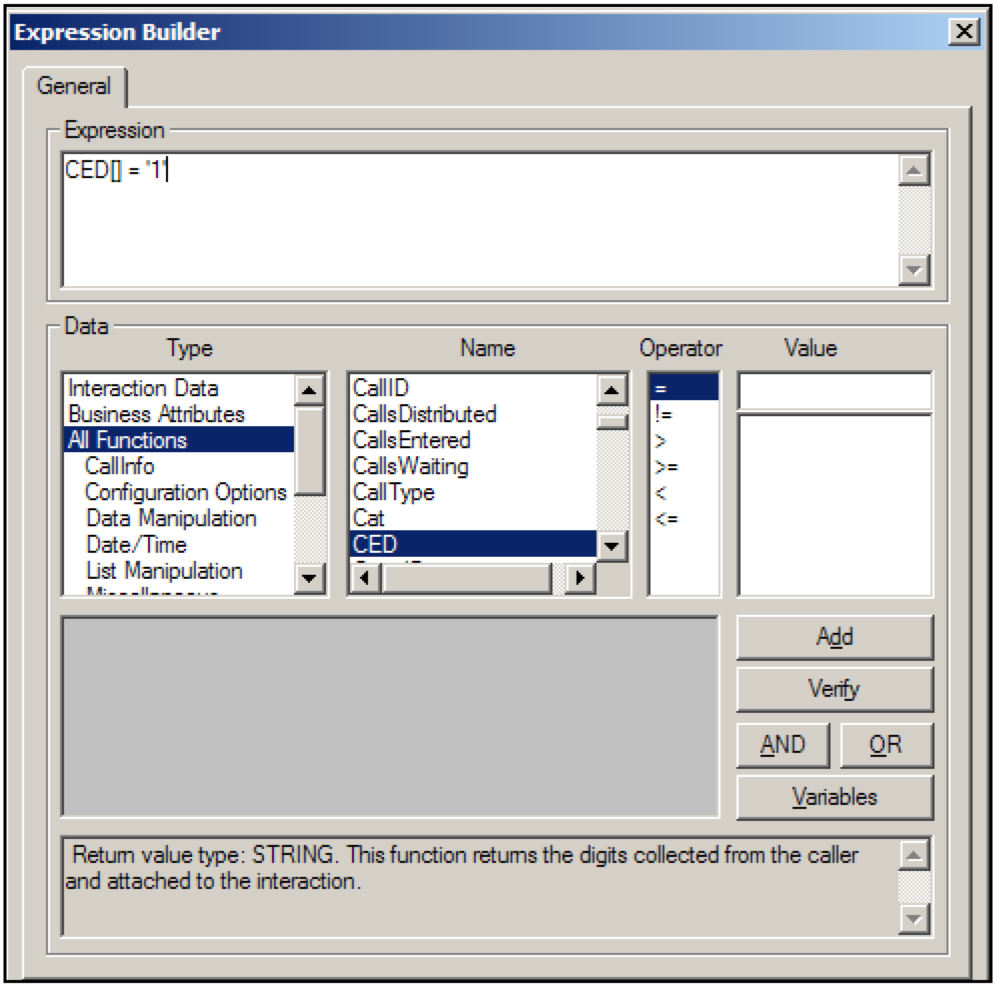


☑ Clear Target

**Figure 27: Selection Properties Dialog Box: Type Field Selected**

The parameters of each object in a properties dialog box are specific to the object. For example:

- The parameters of the Generic Segmentation object (see Figure 4 on page 18) are logical expressions built in the Expression Builder (see Figure 28).



CED[] = '1'

Return value type: STRING. This function returns the digits collected from the caller and attached to the interaction.

**Figure 28: Logical Expression Built in Expression Builder**

- The properties dialog box for the Function object includes the function name, parameters and values (see Figure 48 on page 50).
- The parameters of Voice Treatment objects are the type of treatment and some supplemental data. Voice Treatments affect what the caller hears, for example, music, silence, or a busy signal. Voice Treatments can also be used as parameters of Routing objects.

# What's Next...

Now that you have been introduced to routing strategies and the different types of re-usable and strategy-building objects, you can better understand the sample strategies presented in the chapters ahead.

For detailed information on the objects and dialog boxes in this chapter, see the *Universal Routing 8.1 Reference Manual* or *Universal Routing 8.1 Interaction Routing Designer Help*.

Chapter

# 2 Voice Routing Strategy Samples

This chapter contains the following sections:

For the sake of simplicity, the voice routing strategy samples in this chapter omit error processing.

**Note:** The strategies in this chapter are valid for Universal Routing 8.1.

### About the Graphics

Some graphics showing strategies in this chapter have been rearranged to make all the included objects fit in the available space. When actually creating business processes and strategies, Genesys recommends that you allow adequate space between the objects for clarity. The actual workspaces provided by IRD are much larger than the areas shown in these graphics.

## Strategy 1: Load Balancing Among Queues

Used on a weekday (Monday through Friday), this strategy distributes interactions between two ACD queues. Each call is distributed to the queue that is currently under the least load; the predefined statistic `StatLoadBalance` is used as a measure of the current load of each queue.

Since the strategy is used only on weekdays, it does not make routing decisions on Saturdays and Sundays. If a call is received on one of these days, it is routed to the default destination.

## StatLoadBalance

The predefined statistic `StatLoadBalance` (see “Statistics” on page 24) is designed for balancing loads among queues. The value it reports is generally not an estimate of the wait time for a call in the queue. Wait time is only one of the factors taken into account in the calculation of this statistic.

In order to use this statistic correctly, you must configure an agent group that corresponds to each of the queues for which the statistic is used. These agent groups must be named after the queue’s alias and must consist of exactly those agents to whom calls are distributed from the queue.

**Note:** When calls are routed to a queue or a routing point, the queue or routing point is always assumed to be available—that is, that the Stat Server specified as the target’s location has registered for it successfully, and that no statistical thresholds have been set for the queue or routing point. For such targets it is reasonable always to specify a wait time of `0` in the corresponding target object.

## Strategy in the Routing Design Window

Figure 29 shows the example load-balancing strategy in the IRD Routing Design window.

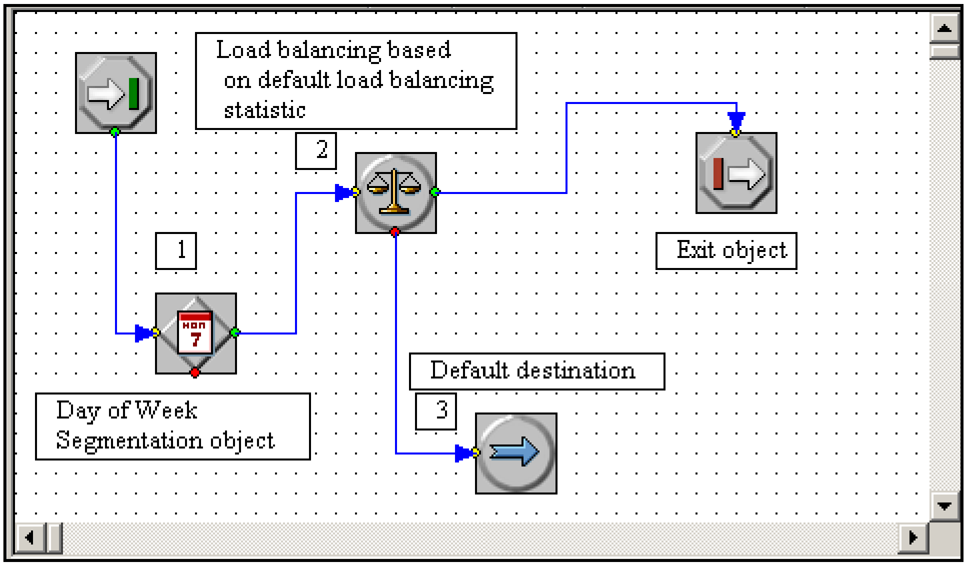
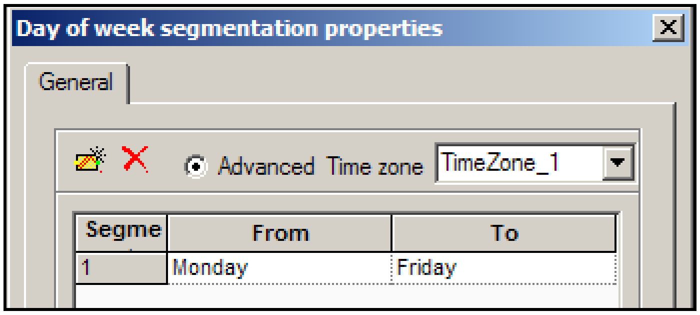

Load balancing based on default load balancing statistic
2
1
Exit object
Default destination
3
Day of Week Segmentation object

**Figure 29: Example Load-Balancing Strategy**

## Summary of Flow

**Note:** The IRD objects that are described this section are keyed to the numbers in Figure 29.

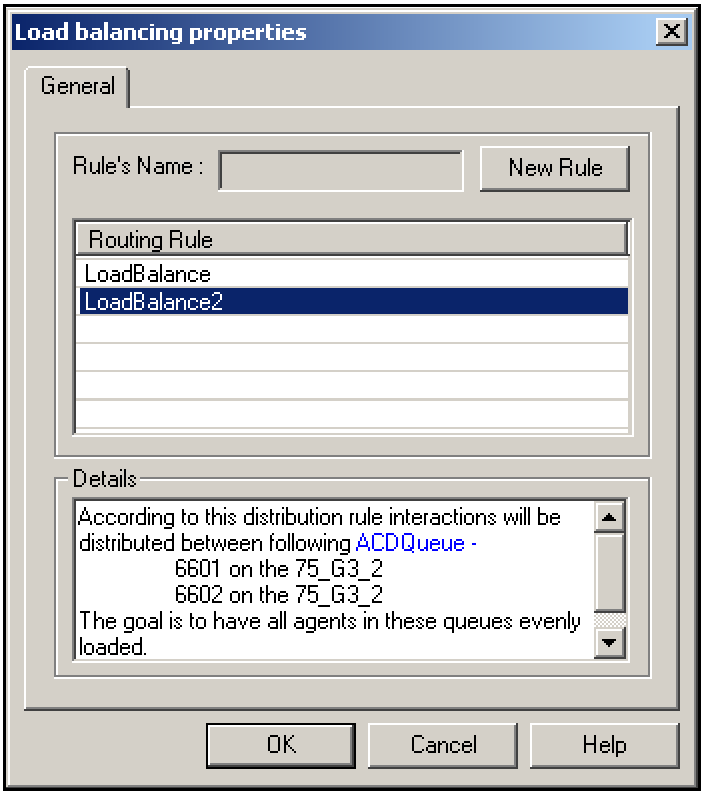
1. A Day of Week Segmentation object causes interactions to take different paths in the strategy depending on when they arrive (see Figure 30).



| Segme | From | To |
|---|---|---|
| 1 | Monday | Friday |

**Figure 30: Day of Week Segmentation Object**

2. If a call arrives during a weekday, it is routed using a Load Balancing Routing object. Figure 31 shows the Properties dialog box for the Load Balancing Routing object, which uses a pre-configured routing rule, `LoadBalance2`.



| Routing Rule |
|---|
| LoadBalance |
| LoadBalance2 |

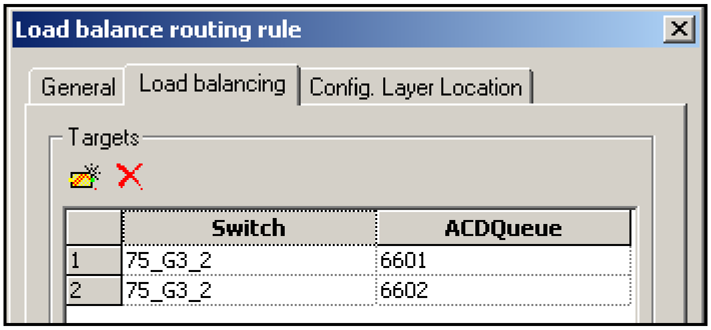
Details

According to this distribution rule interactions will be distributed between following ACDQueue -
6601 on the 75_G3_2
6602 on the 75_G3_2
The goal is to have all agents in these queues evenly loaded.

**Figure 31: Load-Balancing Routing Object**

### Load Balancing Routing Rule

Figure 32 shows the LoadBalance2 routing rule (see “Routing Rules” on page 21) used in Figure 31.



|  | Switch | ACDQueue |
|---|---|---|
| 1 | 75_G3_2 | 6601 |
| 2 | 75_G3_2 | 6602 |

Figure 32: Load Balance Routing Rule

The Switches and ACDQueues in Figure 32 are defined in Configuration Manager.

3. If the call cannot be routed to an ACD queue, the call is routed to the default destination.

**Note:** No dialog box opens for the Default object. You specify the default destination using the default_destination option in the URS Application object in Configuration Manager.

## Strategy 2: Percentage Allocation Among Targets

On weekdays (Monday through Friday), all interactions are distributed between two queues based on a percentage of total calls:

- 82 percent of calls are allocated to queue Boston.
- 18 percent of calls are allocated to queue SanFrancisco.

The proportion is given by the quotient of the weight assigned to a particular target and the sum of the weights of all targets on the list; since the sum of the weights in the example strategy is 100, the weights designate the actual percentage of the total number of calls.

For each available target, URS first computes the actual proportion of calls distributed to it from the total number of calls distributed from the target list. The call is then routed to the available target with the lowest ratio between this actual proportion and the ideal proportion determined from the specified weights.

## Target Type

The fact that this sample strategy uses targets of type queue is immaterial—you can use the same approach with targets of any type; the types of the different targets need not be the same.

## Default Destination

Since the condition is only satisfied on weekdays, the strategy makes no routing decisions on Saturdays and Sundays. If a call is received on one of these days, it is routed to the configured Default Destination.

## The Strategy in the Routing Design Window

Figure 33 shows the example percentage allocation strategy in the IRD Routing Design window.

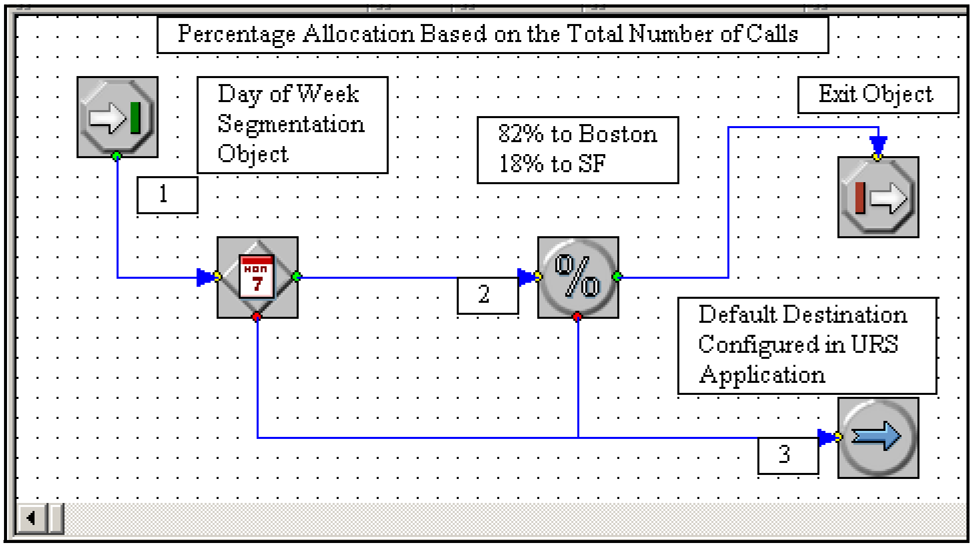
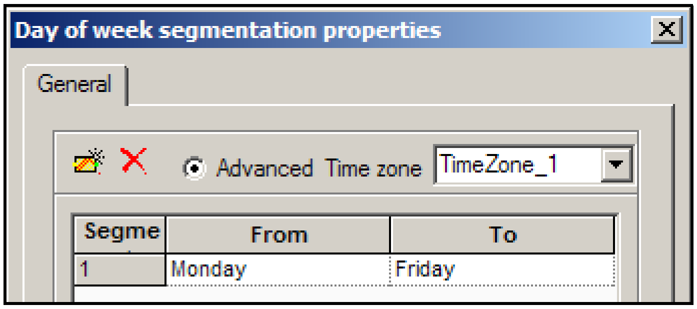

Percentage Allocation Based on the Total Number of Calls
Day of Week Segmentation Object
82% to Boston
18% to SF
Exit Object
1
2
3
Default Destination Configured in URS Application

**Figure 33: Example Percentage Allocation Strategy**

### Summary of Flow

**Note:** The IRD objects that are described this section are keyed to the numbers in Figure 33.

1. A Day of Week Segmentation object causes calls that arrive on a week day to take a different path from calls that arrive on the weekend (see Figure 34).

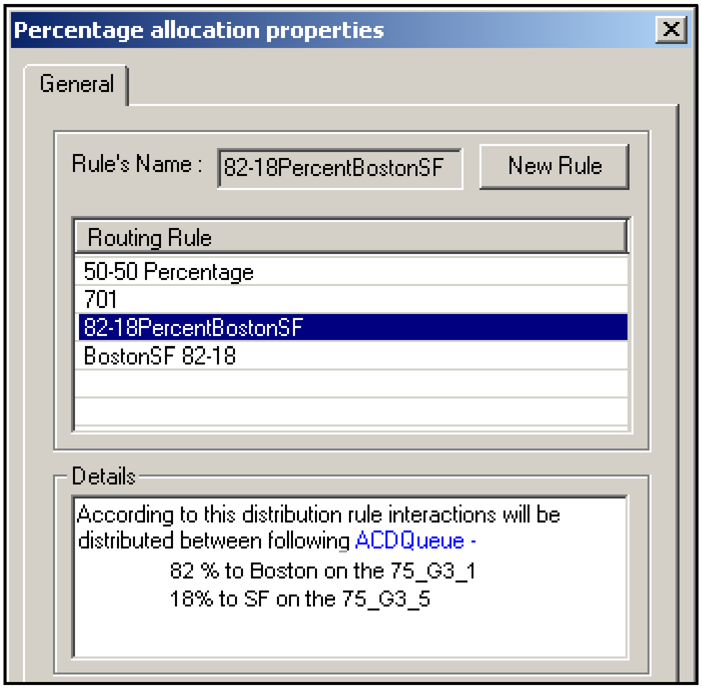


Time zone: TimeZone_1

| Segme | From | To |
|---|---|---|
| 1 | Monday | Friday |

Figure 34: Day of Week Segmentation Object

2. If a call arrives on a week day, a Percentage Routing object uses a pre-configured routing rule, `82-18PercentBostonSF` (see Figure 35).



Rule's Name : 82-18PercentBostonSF

| Routing Rule |
|---|
| 50-50 Percentage |
| 701 |
| 82-18PercentBostonSF |
| BostonSF 82-18 |

According to this distribution rule interactions will be distributed between following ACDQueue -

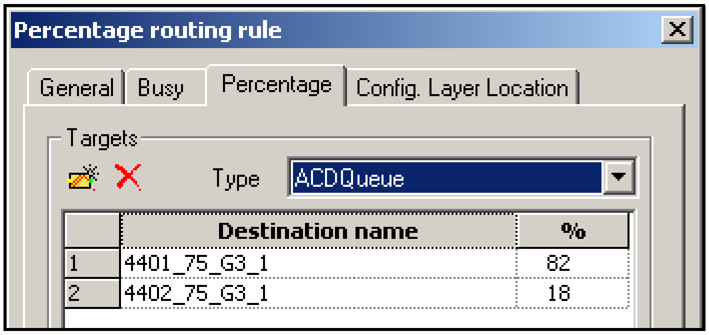
82 % to Boston on the 75_G3_1

18% to SF on the 75_G3_5

Figure 35: Percent Allocation Routing Object

## Percentage Allocation Routing Rule

Figure 36 shows the properties dialog box when the routing rule was initially defined.



| | Destination name | % |
|---|---|---|
| 1 | 4401_75_G3_1 | 82 |
| 2 | 4402_75_G3_1 | 18 |

**Figure 36: Percentage Allocation Routing Rule**

The switches and ACD queues are defined in Configuration Manager.

3. If the call cannot be routed to a specified target, URS routes it to the Default destination.

**Note:** The Default Destination object has no properties dialog box. The default destination is configured in the URS Application object.

# Strategy 3: Routing to Agent Groups

During business hours on weekdays, this strategy attempts to send calls to an agent belonging to one of the two specified Agent Groups, `ParisAgents` and `LondonAgents`.

- If an agent from either group is available to take the call within 15 seconds, the call is sent to that agent. The call goes to the next available agent, who can be in either of the specified Agent Groups.
- If, during the second fifteen-second interval, no agent from either group becomes available to take the call, then the call is sent to the queue with alias `BostonQueue1`.
- The same queue is the only recipient of calls during the limited business hours from 9:00 a.m. to 12 p.m. on Saturday.
- Outside of business hours, every incoming call is sent to one of the IVR ports configured as a group of places named `IVRGroup`.

## The Strategy in the Routing Design Window

Figure 37 shows the example agent group strategy in the IRD Routing Design window.

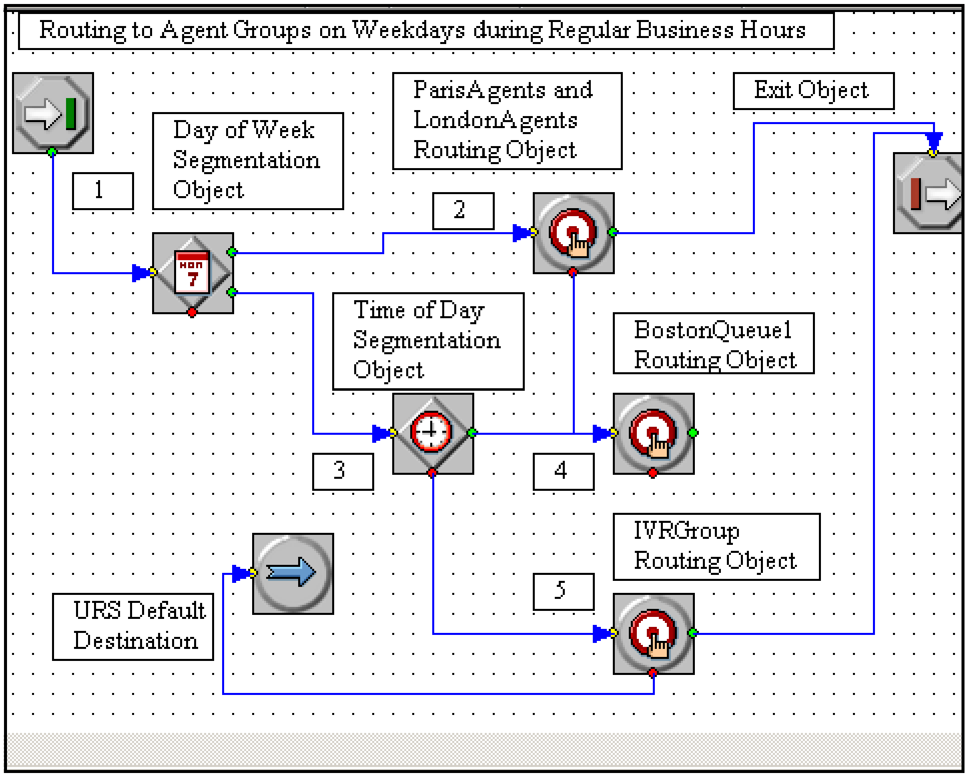
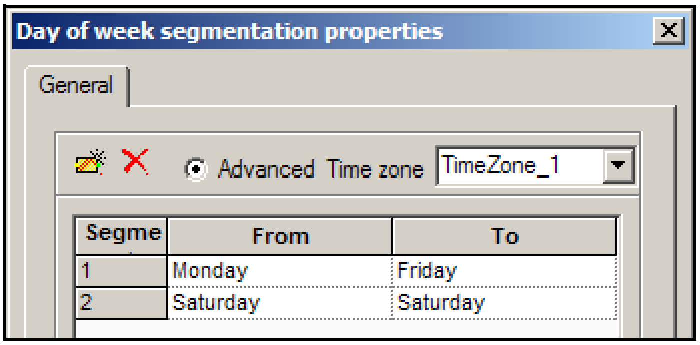

Routing to Agent Groups on Weekdays during Regular Business Hours
Day of Week Segmentation Object
ParisAgents and LondonAgents Routing Object
Exit Object
1
2
Time of Day Segmentation Object
BostonQueue1 Routing Object
3
4
IVRGroup Routing Object
5
URS Default Destination

Figure 37: Example Strategy for Routing to Agent Groups

## Summary of Flow

**Note:** The IRD objects that are described this section are keyed to the numbers in Figure 37.

1. A Day of Week Segmentation object causes calls to take different paths in the strategy based on when they arrive (see Figure 38).

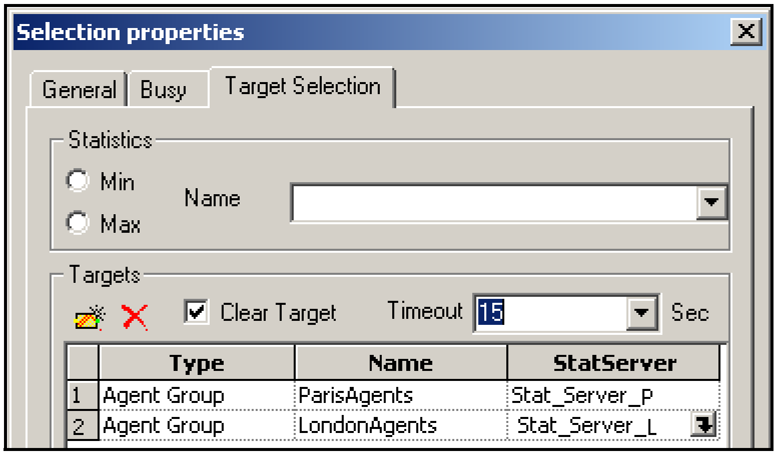


● Advanced Time zone TimeZone_1

| Segme | From | To |
|---|---|---|
| 1 | Monday | Friday |
| 2 | Saturday | Saturday |

Figure 38: Day of Week Segmentation Object

2. If a call arrives at any time on a week day, a Routing Selection object sends it to the first available agent in one of the two specified Agent Groups, `ParisAgents` and `LondonAgents` (see Figure 39).



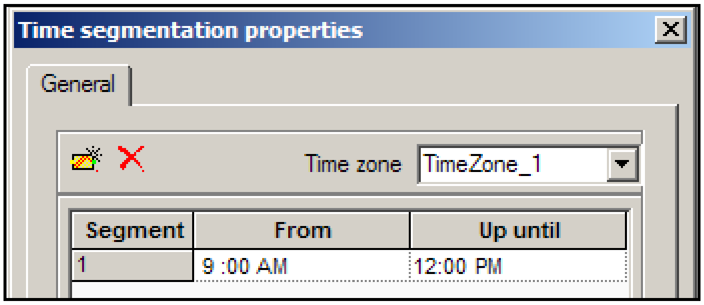
○ Min
○ Max
Name:

☑ Clear Target
Timeout: 15 Sec

|  | Type | Name | StatServer |
|---|---|---|---|
| 1 | Agent Group | ParisAgents | Stat_Server_P |
| 2 | Agent Group | LondonAgents | Stat_Server_L |

**Figure 39: Paris and London Agents Routing Selection Object**

3. If a call arrives on Saturday, a Time Segmentation object routes based on whether the call arrives between 9:00 a.m. and 12:00 p.m (see Figure 40)

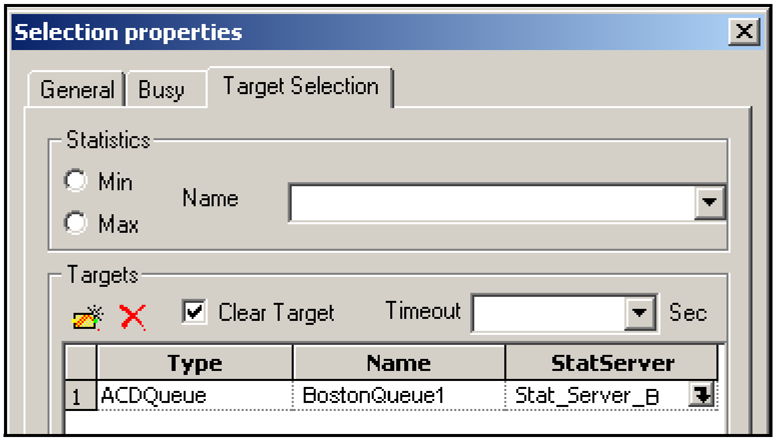


Time zone: TimeZone_1

| Segment | From | Up until |
|---|---|---|
| 1 | 9 :00 AM | 12:00 PM |

**Figure 40: Time Segmentation Object**

4. If the call arrives between 9:00 a.m. and 12:00 p.m., it goes to a Routing Selection object specifying an ACD queue `BostonQueue1` (see Figure 41).



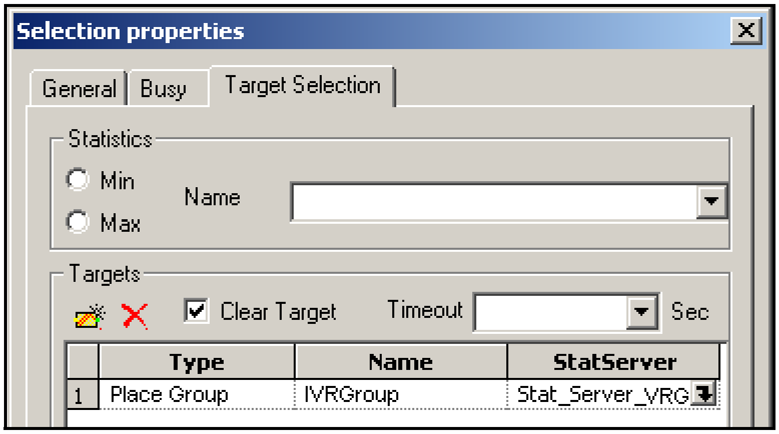
○ Min
○ Max
Name:

☑ Clear Target
Timeout: Sec

|  | Type | Name | StatServer |
|---|---|---|---|
| 1 | ACDQueue | BostonQueue1 | Stat_Server_B |

**Figure 41: BostonQueue1 Routing Selection Object**

5. If the call arrives before 9:00 a.m. or after 12:00 p.m. on Saturday, it goes to a Routing Selection object that specifies an IVR place group as target (see Figure 42 on page 46).



☑ Clear Target

Timeout Sec

|  | Type | Name | StatServer |
|---|---|---|---|
| 1 | Place Group | IVRGroup | Stat_Server_VRG |

**Figure 42: IVRGroup Routing Selection Object**

6. If the call cannot be sent to the ACQ queue or the IVR place group, it is routed to the Default destination.

**Note:** The Default Destination object has no properties dialog box. The default destination is configured in the URS Application object.

# Strategy 4: Skills-Based Routing

A cable provider would like to send calls from potential customers to representatives responsible for new accounts. The cable company sets up an IVR to prompt callers for:

- The language they prefer to speak:
  - 1 for English
  - 2 for Spanish
- Their account type:
  - 1 for new accounts
  - 2 for existing accounts

Based on the customer’s responses, the routing strategy routes the call to an agent that speaks the customer’s language and handles the customer’s account needs. The agent gets a screen pop indicating the customer’s preferred language.

The calls wait up to 10 seconds for an agent with a skill level of 5 or above. If none are available during that interval, the strategy checks for an immediately available agent with a skill level of 4 or below. If no such agent is found, URS sends the call to the Default destination.

## Interaction Data

Interactive Voice Response (IVR) interfaces can request T-Server to attach data to calls. This enables responses in the IVR to be passed to the Genesys software. URS can then receive this attached data from T-Server messaging, such as in an `EventRouteRequest` message.

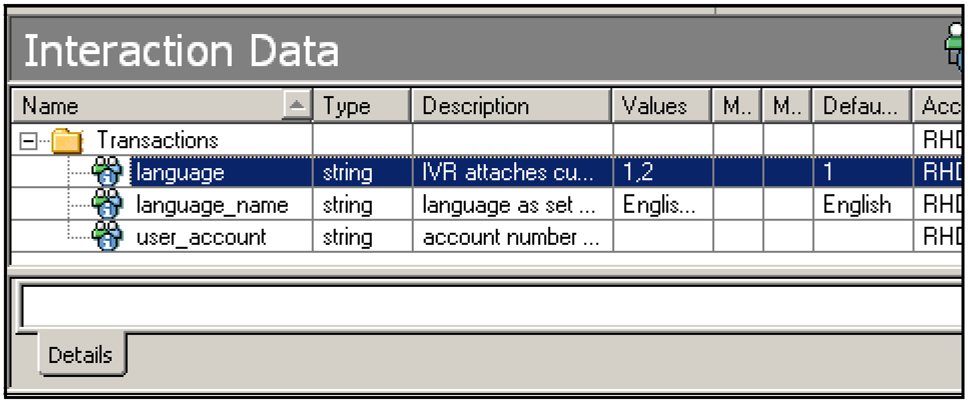
This example strategy uses interaction data (see page 23), which:

- Defines the attached data keys used in strategy.
- Enables strategies to gather attached data from event messaging.
- Enables strategies to attach data for screen pops.

**Note:** Interaction data names must match the names of keys used in the IVR interface, the agent desktop application for screen pops, and any other applications that will use the attached data.

The example strategy assumes that the IVR attaches the account type with the key `acct_type` and the customer's preferred language with the key `language`. The desktop programmer used the key `language` for the agent screen pop to indicate the customer's preferred language.

Figure 43 on page 47 shows the `language` interaction data configured for the example skills routing strategy.



| Name | Type | Description | Values | M.. | M.. | Defau... | Acc |
|---|---|---|---|---|---|---|---|
| Transactions | | | | | | | RHI |
| language | string | IVR attaches cu... | 1,2 | | | 1 | RHI |
| language_name | string | language as set ... | Englis... | | | English | RHI |
| user_account | string | account number ... | | | | | RHI |

**Figure 43: Interaction Data Configured for Skills Routing Strategy**

## The Strategy in the Routing Design Window

Figure 44 shows the example skills-based routing strategy as it appears in the IRD Routing Design window.

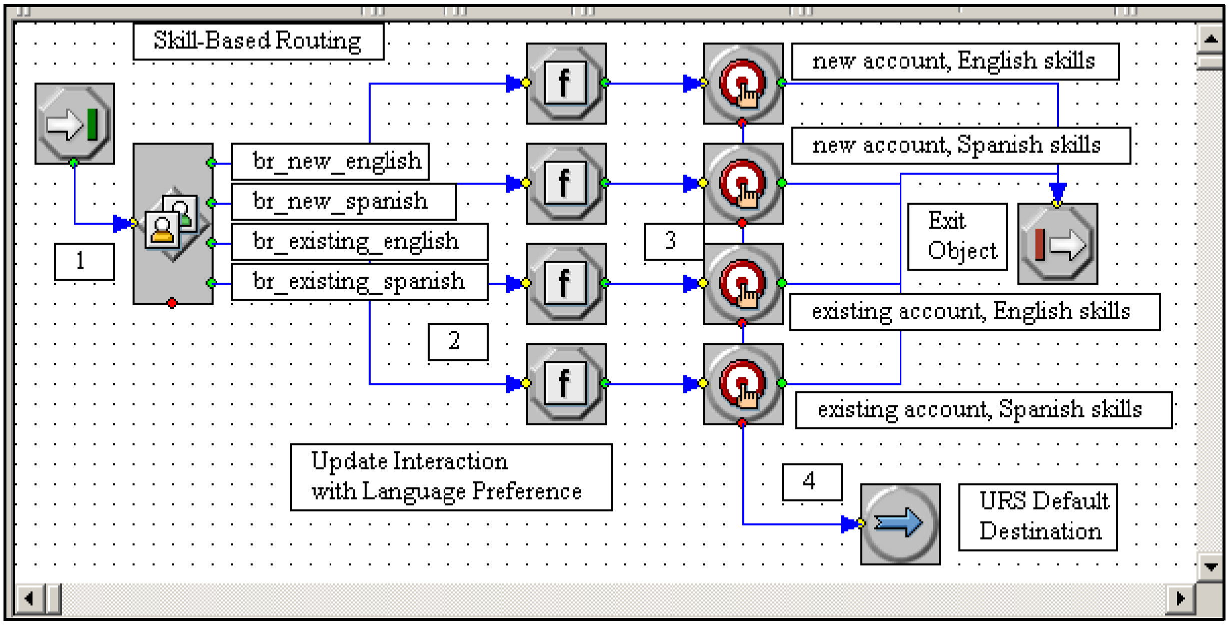
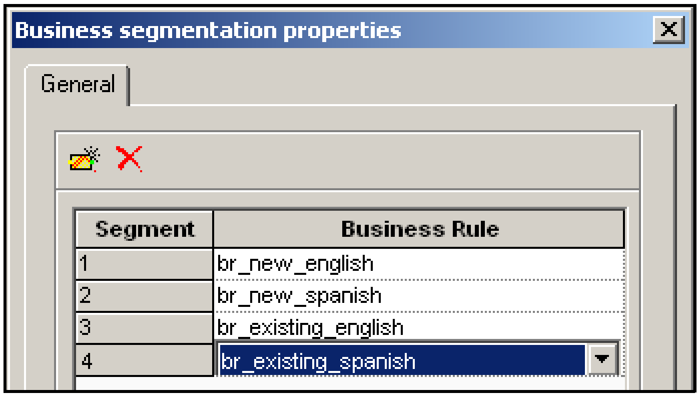

Skill-Based Routing
br_new_english
br_new_spanish
br_existing_english
br_existing_spanish
1
2
3
4
f
new account, English skills
new account, Spanish skills
Exit Object
existing account, English skills
existing account, Spanish skills
Update Interaction with Language Preference
URS Default Destination

Figure 44: Example Skills-Based Routing Strategy

## Summary of Flow

The IRD objects that are described this section are keyed to the numbers in Figure 44 on page 48.

1. A Business Segmentation object causes interactions to take different paths based on business rules (see Figure 45).


Business segmentation properties

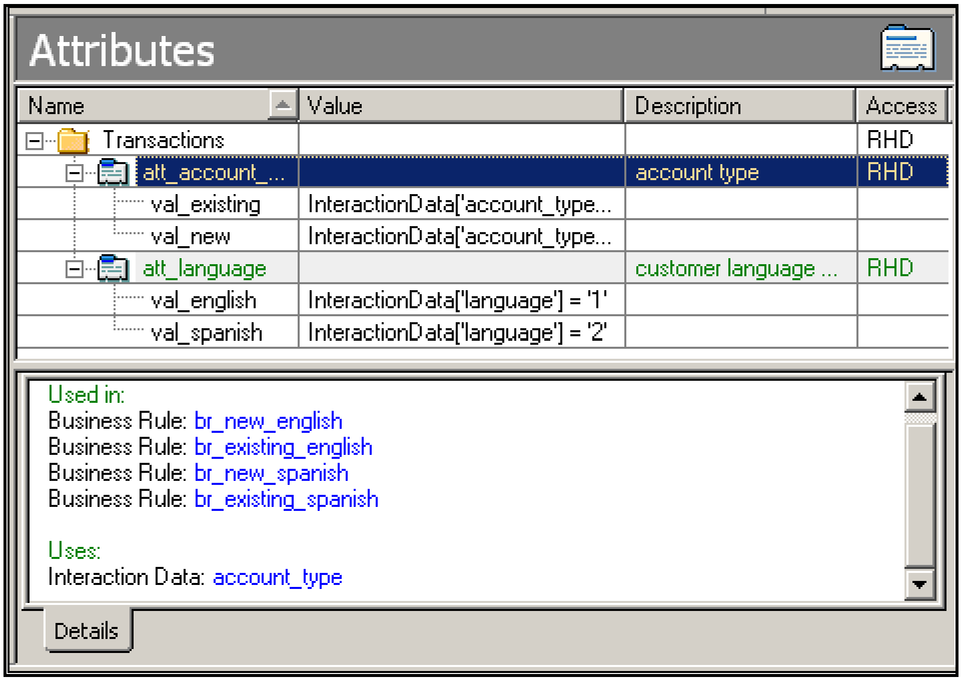
General

| Segment | Business Rule |
|---|---|
| 1 | br_new_english |
| 2 | br_new_spanish |
| 3 | br_existing_english |
| 4 | br_existing_spanish |

Figure 45: Business Segmentation Object for Skills Routing Strategy

### Attributes and Business Rules

The business rules in Figure 45 are created from attributes (see page 22). Attributes and business rules create logical expressions used for segmentation. Figure 46 shows the attributes configured for the business rules in Figure 47 on page 49.



Attributes

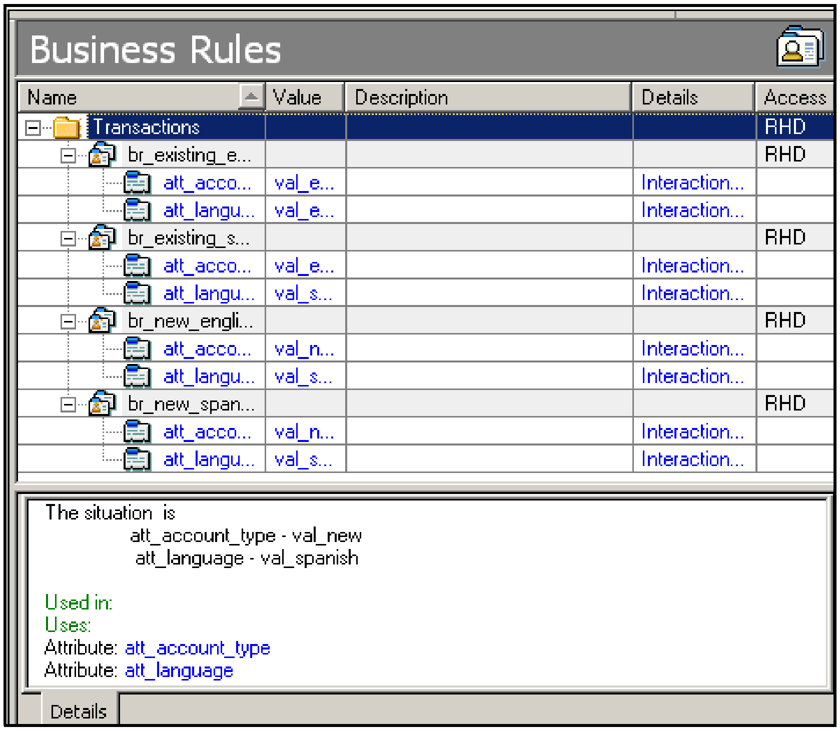
| Name | Value | Description | Access |
|---|---|---|---|
| Transactions | | | RHD |
| att_account_... | | account type | RHD |
| val_existing | InteractionData['account_type... | | |
| val_new | InteractionData['account_type... | | |
| att_language | | customer language ... | RHD |
| val_english | InteractionData['language'] = '1' | | |
| val_spanish | InteractionData['language'] = '2' | | |

Used in:
Business Rule: br_new_english
Business Rule: br_existing_english
Business Rule: br_new_spanish
Business Rule: br_existing_spanish

Uses:
Interaction Data: account_type

**Figure 46: Attribute Data Configured for Skills Routing Strategy**

Figure 47 shows the business rules used in Figure 45 on page 48 as they appear in the IRD main window:



Business Rules

| Name | Value | Description | Details | Access |
|---|---|---|---|---|
| Transactions | | | | RHD |
| br_existing_e... | | | | RHD |
| att_acco... | val_e... | | Interaction... | |
| att_langu... | val_e... | | Interaction... | |
| br_existing_s... | | | | RHD |
| att_acco... | val_e... | | Interaction... | |
| att_langu... | val_s... | | Interaction... | |
| br_new_engli... | | | | RHD |
| att_acco... | val_n... | | Interaction... | |
| att_langu... | val_s... | | Interaction... | |
| br_new_span... | | | | RHD |
| att_acco... | val_n... | | Interaction... | |
| att_langu... | val_s... | | Interaction... | |

The situation is
att_account_type - val_new
att_language - val_spanish

Used in:
Uses:
Attribute: att_account_type
Attribute: att_language

**Figure 47: Business Rules Configured for Skills Routing Strategy**

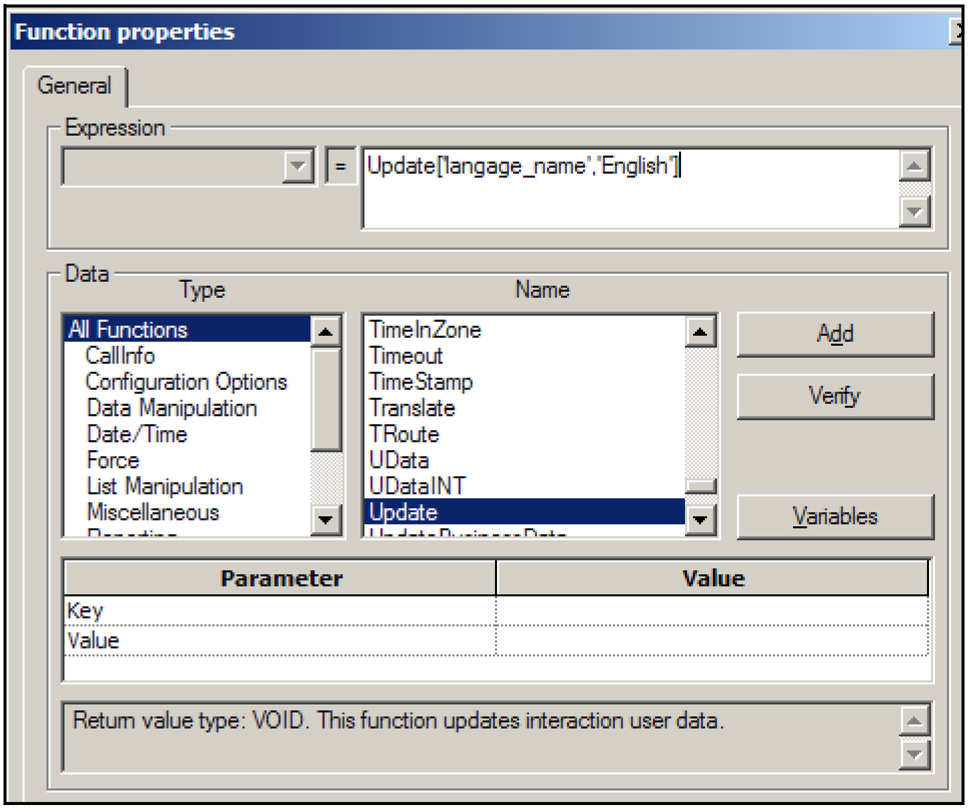
2. If the call matches the business rule shown in Figure 45 on page 48, it is routed to the corresponding Function object.

   For each branch of the Business Segmentation object:

   - English, new accounts (`br_new_english`)
   - Spanish, new accounts (`br_new_spanish`)
   - English, existing accounts (`br_existing_english`)
   - Spanish, existing accounts (`br_existing_spanish`)

   each Function object updates the interaction with the customer's preferred language, previously entered via the IVR.

   Figure 48 shows the Function object used for updating the English, new accounts segment (first segment in Figure 44 on page 48).



Expression: = Update['langage_name','English']

Type: All Functions (selected), CallInfo, Configuration Options, Data Manipulation, Date/Time, Force, List Manipulation, Miscellaneous

Name: TimeInZone, Timeout, TimeStamp, Translate, TRoute, UData, UDataINT, Update (selected)

| Parameter | Value |
|---|---|
| Key | |
| Value | |

Return value type: VOID. This function updates interaction user data.

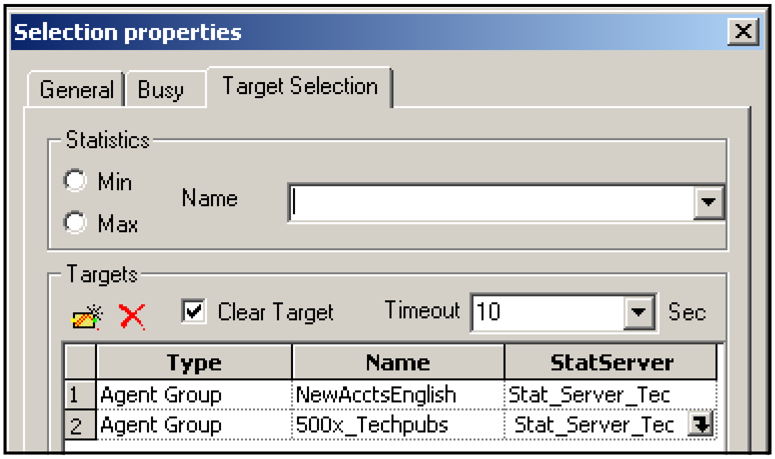
**Figure 48: Function Object for Updating Interaction Data**

3. For each branch of the Business Segmentation object:

   - English, new accounts (`br_new_english`)
   - Spanish, new accounts (`br_new_spanish`)
   - English, existing accounts (`br_existing_english`)
   - Spanish, existing accounts (`br_existing_spanish`)

   a corresponding Routing Selection object targets an agent group with the required language/account skill.

Figure 49 shows the properties dialog box for the Routing Selection object shown in the strategy in Figure 44 on page 48.



○ Min ○ Max Name

☑ Clear Target Timeout 10 Sec

|  | Type | Name | StatServer |
|---|---|---|---|
| 1 | Agent Group | NewAcctsEnglish | Stat_Server_Tec |
| 2 | Agent Group | 500x_Techpubs | Stat_Server_Tec |

**Figure 49: Routing Selection Objects for New Accounts**

4. If, within 10 seconds, an agent is not available from either agent group specified in the Routing Selection object in Figure 49, the interaction goes through the red port to the Default destination, as specified in the URS Application object.

# Strategy 5: Database Lookup

A company would like to route calls based on the calling customer's revenue potential (customer segment or *tier*).

When a customer calls, the switch prompts the customer to enter their account number, which is stored in the interaction as Caller Entered Digits (CEDs).

Tiers rank the customer by importance. Customers are assigned to one of the three possible tiers (`Platinum`, `Gold`, and `Silver`) depending on how much revenue they are expected to generate. Based on the tier, the call is routed to agents with a corresponding skill range:

- Platinum customers are routed to agents with a skill level of 9 or more.
- Gold customers are routed to agents with a skill level from 6 to 8.
- Silver customers are routed to agents with a skill level up to 5.

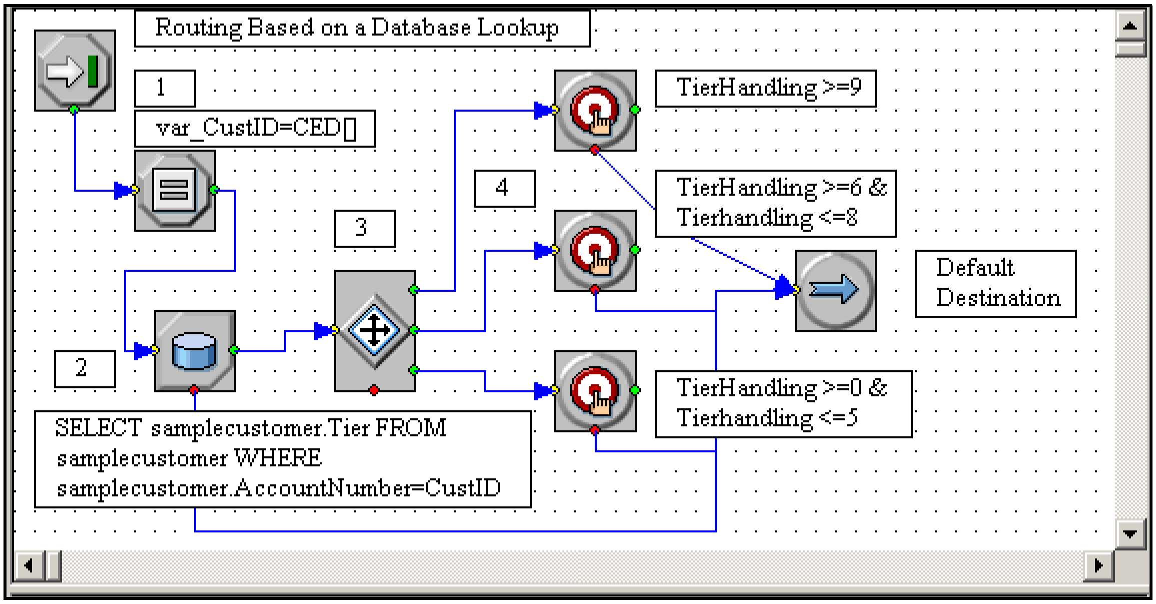
If the tier was not identified correctly or if none of the corresponding agents are immediately available, the call is routed to the default destination.

The database table, `samplecustomer`, contains at least two fields for each record: `AccountNumber` and `Tier`. For each account number, the table contains a unique record.

The contact center agents are organized in a skill hierarchy according to a skill named `Tierhandling`.

# The Strategy in the Routing Design Window

Figure 50 shows the `str_database_lookup` strategy in the IRD Routing Design window.



```
SELECT samplecustomer.Tier FROM
samplecustomer WHERE
samplecustomer.AccountNumber=CustID
```

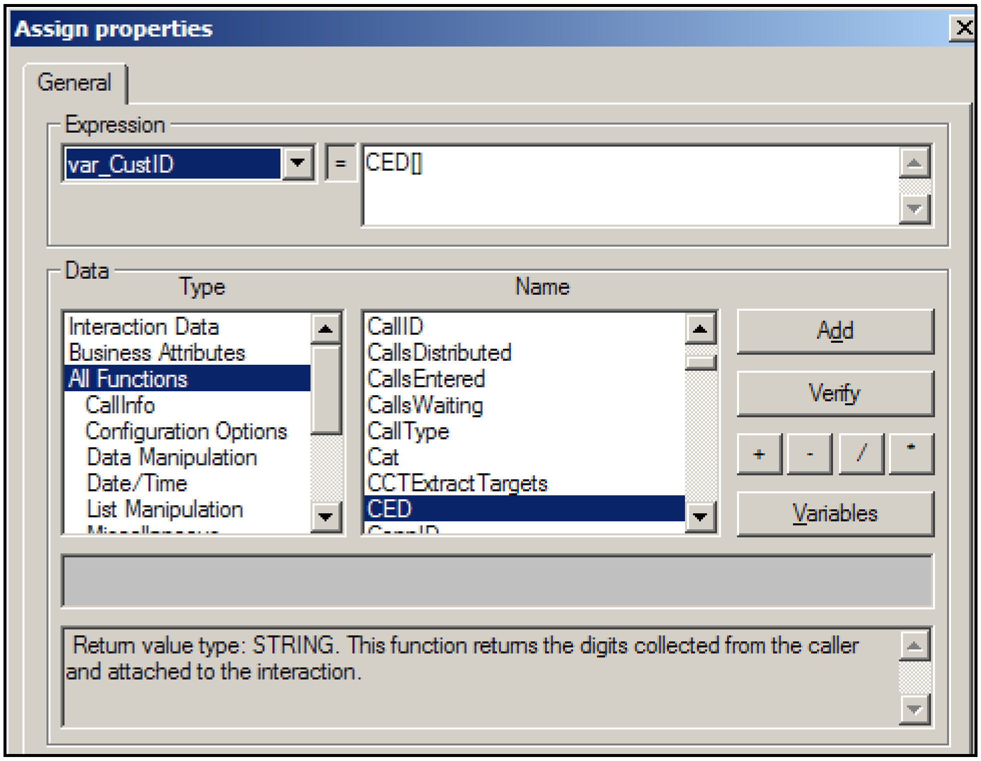
**Figure 50: Example Database Lookup Strategy**

## Summary of Flow

**Note:** The IRD objects that are described this section are keyed to the numbers in Figure 50.

1. An Assign object assigns the account number previously obtained using CEDs to the strategy variable `var_CustID`.

   The IRD function CED (Caller Entered Digits) retrieves the digits already collected from the caller via the switch (see Figure 51).



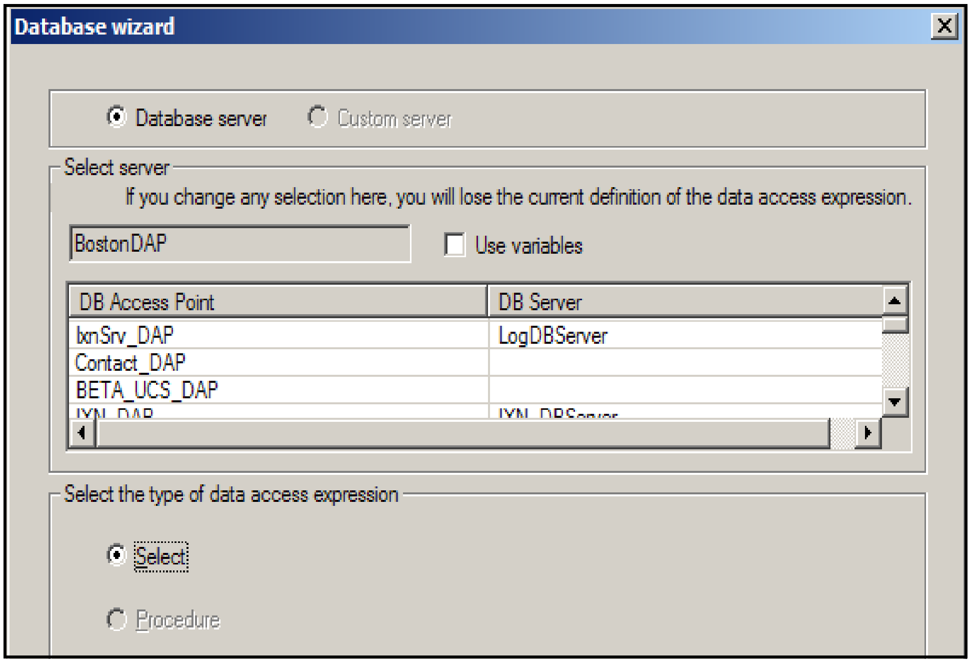
Expression: var_CustID = CED[]

| Type | Name |
|---|---|
| Interaction Data | CallID |
| Business Attributes | CallsDistributed |
| All Functions | CallsEntered |
| CallInfo | CallsWaiting |
| Configuration Options | CallType |
| Data Manipulation | Cat |
| Date/Time | CCTExtractTargets |
| List Manipulation | CED |

Return value type: STRING. This function returns the digits collected from the caller and attached to the interaction.

**Figure 51: Using the Assign Object to Assign CEDs to a Variable**

2. Using a Database Wizard object to look up the customer tier based on the entered account number, the strategy specifies a Database Server and Database Access Point (DAP) for the SELECT statement. The value is obtained through the database access point `Boston_DAP` (see Figure 52).



● Database server ○ Custom server

If you change any selection here, you will lose the current definition of the data access expression.

BostonDAP □ Use variables

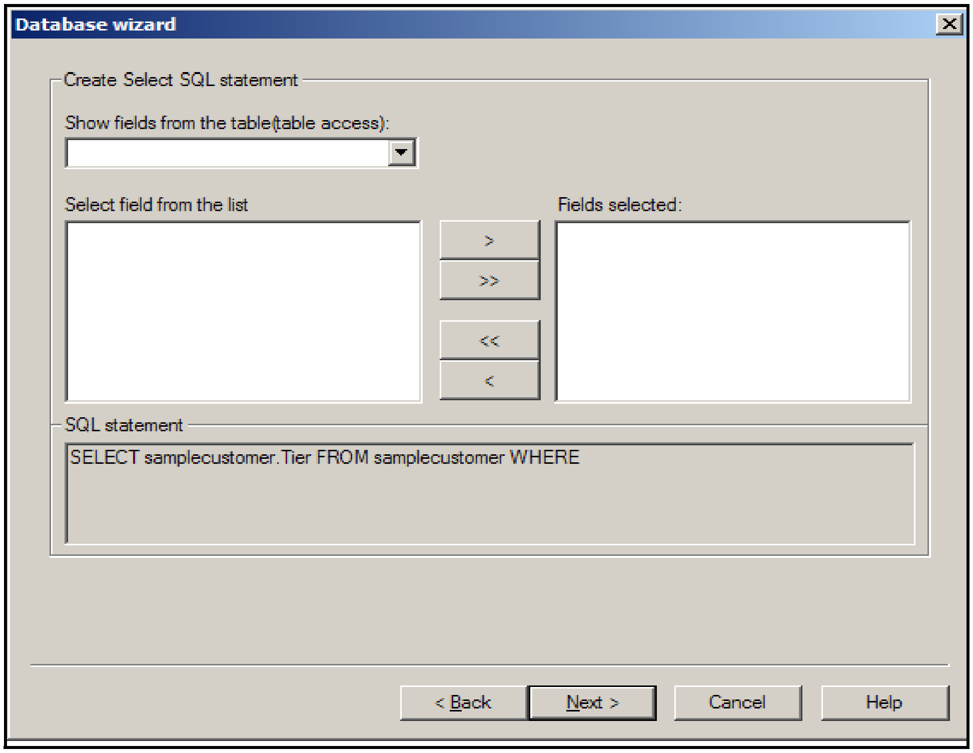
| DB Access Point | DB Server |
|---|---|
| IxnSrv_DAP | LogDBServer |
| Contact_DAP | |
| BETA_UCS_DAP | |

● Select

○ Procedure

**Figure 52: Database Wizard Starting Dialog Box**

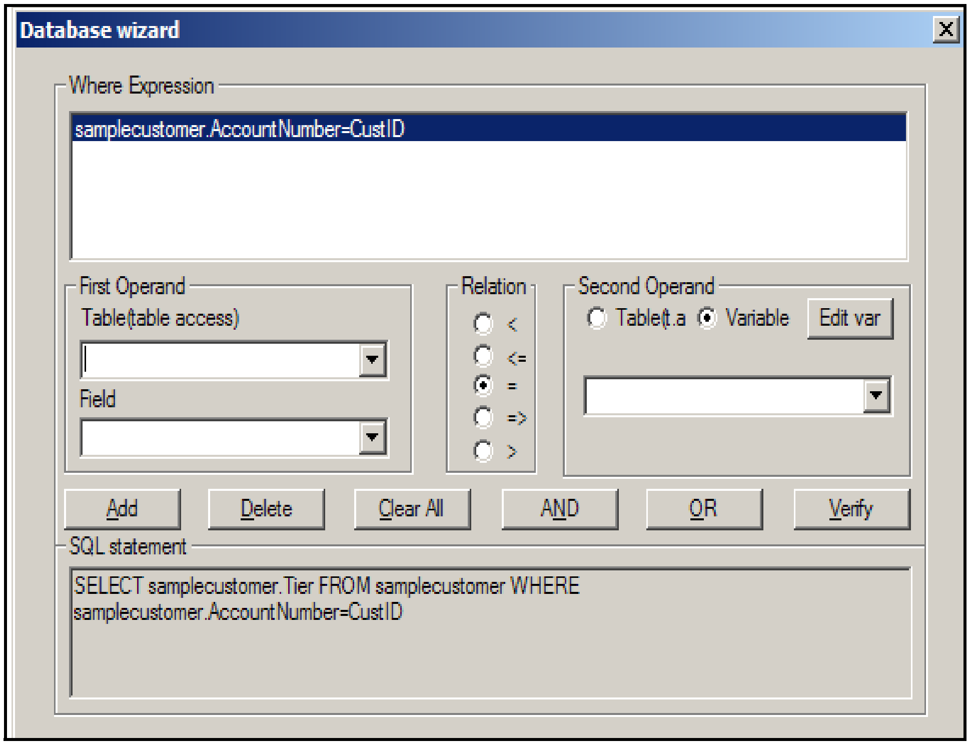
In the next Database Wizard object dialog box, the strategy retrieves the value of the field `Tier` from the table `samplecustomer` for the record where the field `AccountNumber` equals the number supplied (see Figure 53).


Database wizard
Create Select SQL statement
Show fields from the table(table access):
Select field from the list
Fields selected:
>
>>
<<
<
SQL statement
< Back
Next >
Cancel
Help

SELECT samplecustomer.Tier FROM samplecustomer WHERE

**Figure 53: Database Wizard, Create Select SQL Statement**

The next Database Wizard dialog box constructs the `WHERE` clause (see Figure 54).



Where Expression: samplecustomer.AccountNumber=CustID

SQL statement:

```
SELECT samplecustomer.Tier FROM samplecustomer WHERE
samplecustomer.AccountNumber=CustID
```

**Figure 54: Database Wizard, Where Expression**

The next Database Wizard dialog box enables you to specify a sort order. This option is not used in this example (see Figure 55).

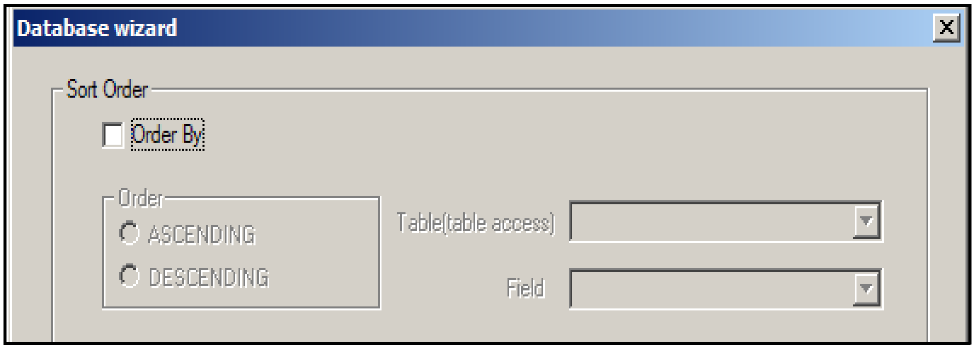
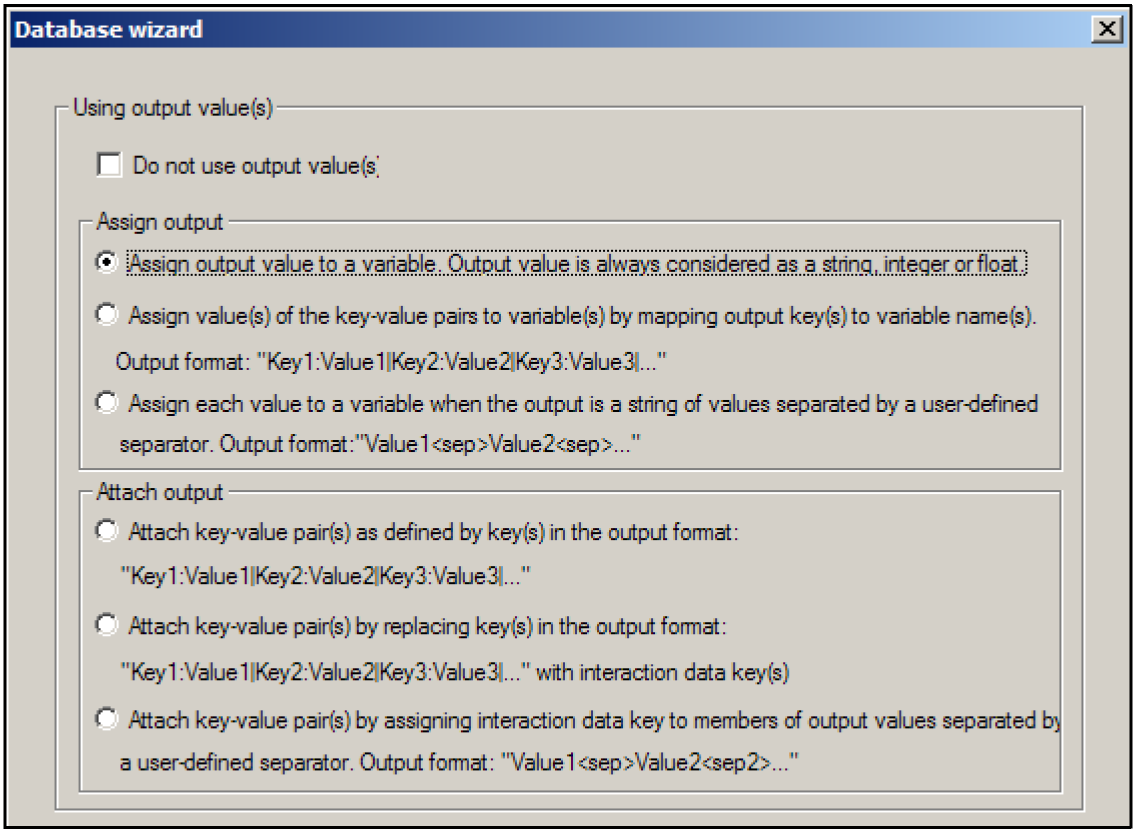


**Figure 55: Database Wizard, Sort Order**

The next Database Wizard dialog box gives the option of assigning the output to a variable or attaching it to the interaction (see Figure 56).



Using output value(s)

□ Do not use output value(s

Assign output

● Assign output value to a variable. Output value is always considered as a string, integer or float.

○ Assign value(s) of the key-value pairs to variable(s) by mapping output key(s) to variable name(s).

Output format: "Key1:Value1|Key2:Value2|Key3:Value3|..."

○ Assign each value to a variable when the output is a string of values separated by a user-defined separator. Output format:"Value1<sep>Value2<sep>..."

Attach output

○ Attach key-value pair(s) as defined by key(s) in the output format:

"Key1:Value1|Key2:Value2|Key3:Value3|..."

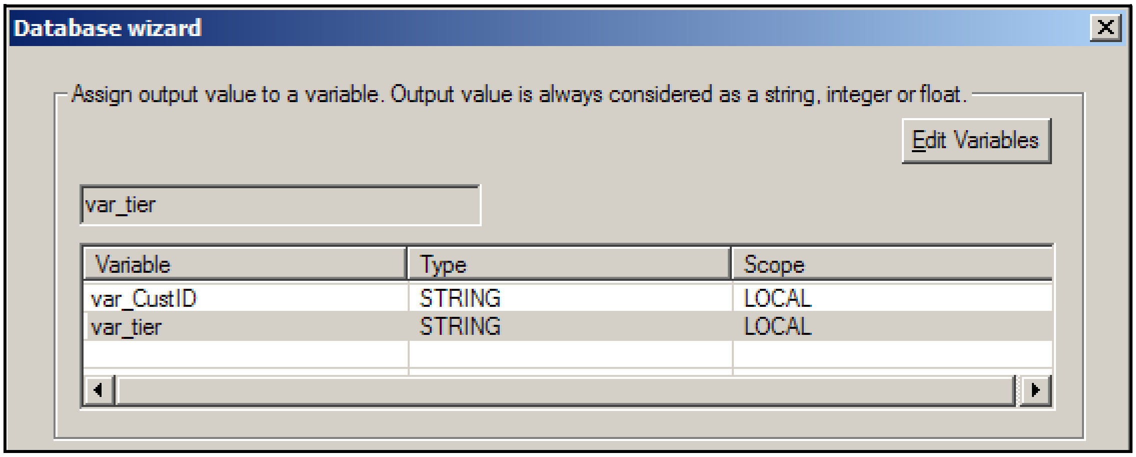
○ Attach key-value pair(s) by replacing key(s) in the output format:

"Key1:Value1|Key2:Value2|Key3:Value3|..." with interaction data key(s)

○ Attach key-value pair(s) by assigning interaction data key to members of output values separated by a user-defined separator. Output format: "Value1<sep>Value2<sep2>..."

**Figure 56: Database Wizard, Using Output Value(s)**

The final Database Wizard dialog box enables you to assign the output value to a variable (see Figure 57).



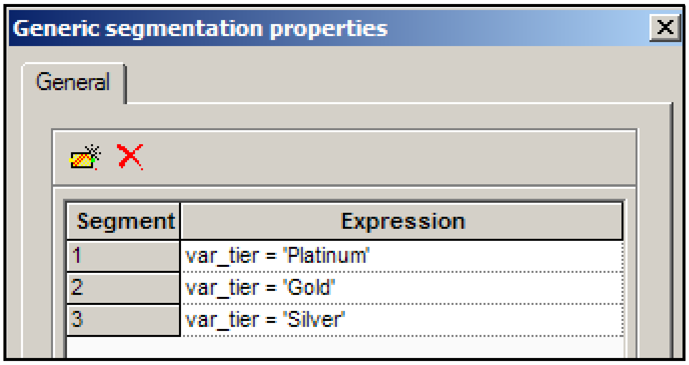
Assign output value to a variable. Output value is always considered as a string, integer or float.

var_tier

| Variable | Type | Scope |
|---|---|---|
| var_CustID | STRING | LOCAL |
| var_tier | STRING | LOCAL |

**Figure 57: Database Wizard, Assign Output Value to a Variable**

3. The next object in Figure 50 on page 52, is a Generic Segmentation object, which causes interactions to take different paths in the strategy. See Figure 58.



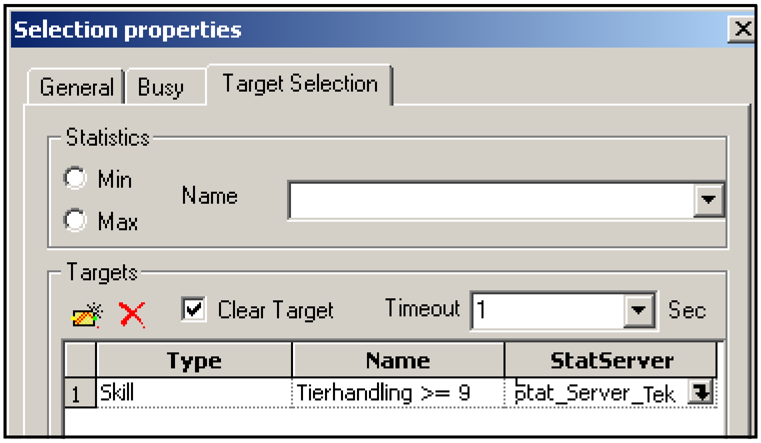
| Segment | Expression |
|---|---|
| 1 | var_tier = 'Platinum' |
| 2 | var_tier = 'Gold' |
| 3 | var_tier = 'Silver' |

**Figure 58: Generic Segmentation Object, Customer Tiers**

In this strategy, the different paths are based on the value of the `var_tier` variable specified in Figure 57. Interactions take different paths based on whether the customer belongs to the `Platinum`, `Gold`, or `Silver` tier.

4. The next three objects in the strategy shown in Figure 50 on page 52 are Routing Selection objects.

   The top Routing Selection object is used for customers of the `Platinum` tier. It routes interactions to agents based on the value of a skill expression (see Figure 59).



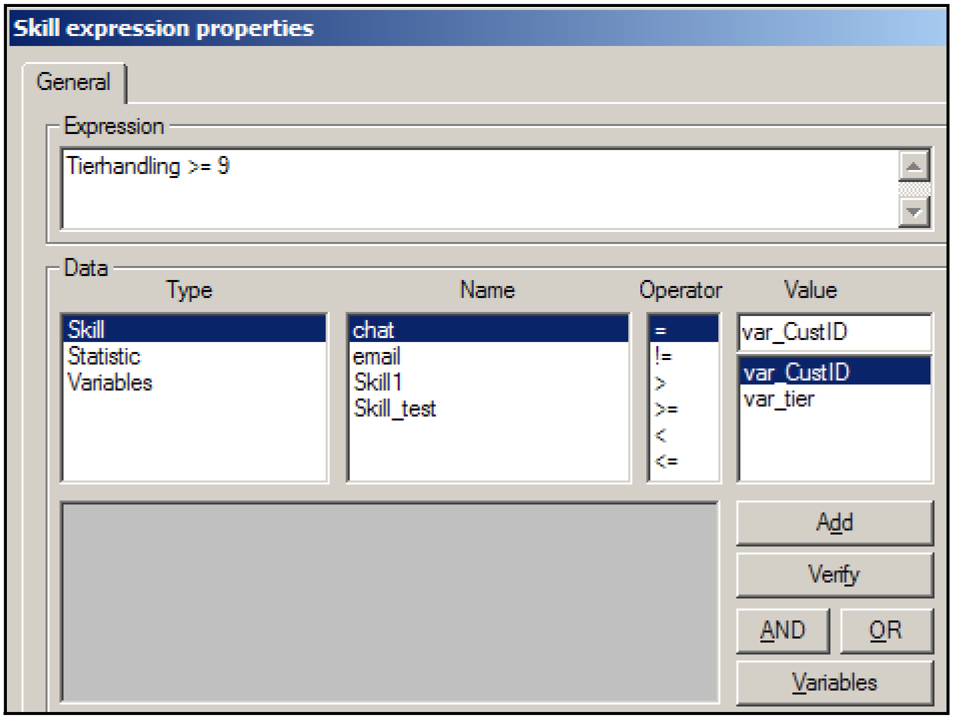
○ Min
○ Max
Name

☑ Clear Target
Timeout 1 Sec

|  | Type | Name | StatServer |
|---|---|---|---|
| 1 | Skill | Tierhandling >= 9 | Stat_Server_Tek |

**Figure 59: Routing Selection Object, Target Selection Tab**

You construct the skill expression in the Expression Builder, which opens when you click the down arrow under `Name` in Figure 59. Figure 60 shows how the expression appears in Expression Builder.



Expression: Tierhandling >= 9

Type: Skill, Statistic, Variables

Name: chat, email, Skill1, Skill_test

Operator: =, !=, >, >=, <, <=

Value: var_CustID

var_CustID, var_tier

**Figure 60: Expression Builder, Constructing Skill Expression**

The remaining Routing Selection objects in the strategy shown in Figure 50 on page 52 also route based on a skill expression.

5. If the tier was not identified correctly, or if none of the corresponding agents are immediately ready, the call is routed to the Default destination.

# Strategy 6: Using IVR Data

This simple strategy isolates calls with a particular type of interaction data. The calls are then routed to a specialized agent group. All other calls are sent to the Default destination.

The calls to be routed to the agent group `AgentGroup@Boston.GA` are identified in two stages.

1. An IVR collects the caller's account number and stores it in the User Data structure of the call under the key `FromIVRDATA`.

2. A Custom Server procedure uses the provided account number as an argument to produce a yes-or-no answer to the question regarding whether an open customer ticket is detected in a database record for this customer. The answer is returned in the format `Y` for affirmative and `N` for negative.
   - If an open ticket is found, the call is routed to the designated agent group if an agent in the group is immediately available to take the call.
   - In all other cases, the call is routed to the Default destination.

Note that the last part of the strategy is an explicit instruction to send the call to the Default destination. This step improves the strategy, but is not required for it to function correctly. If this step is omitted, the call is still sent to the Default destination after the explicit strategy instructions fail to route the call.

## The Strategy in the Routing Design Window

Figure 61 shows the `str_using_IVR_data` strategy in the IRD Routing Design window.

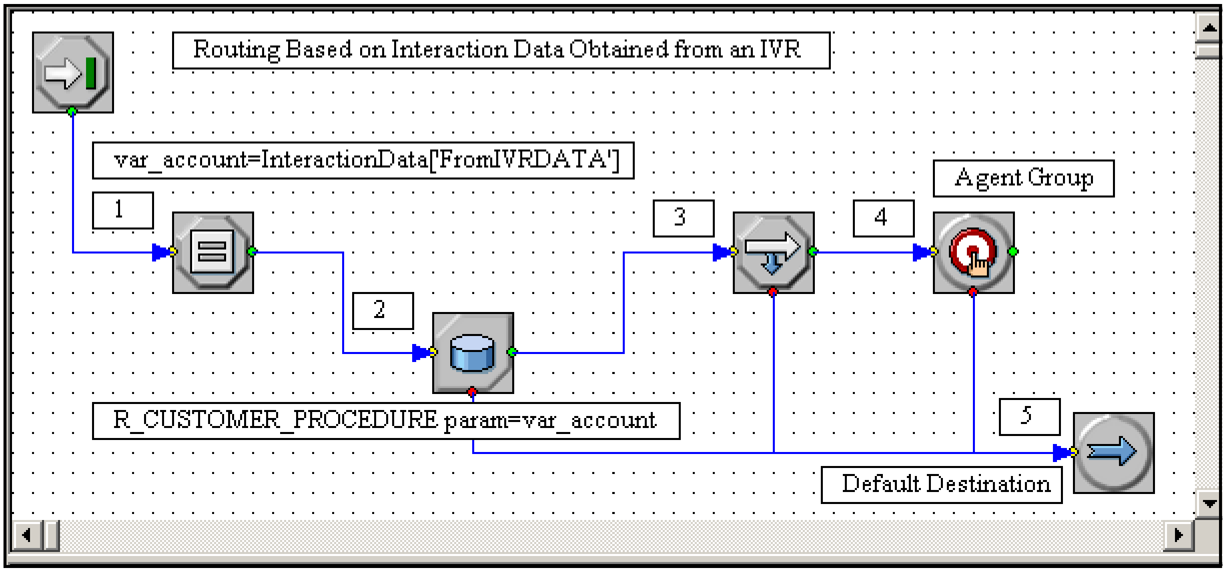

Routing Based on Interaction Data Obtained from an IVR
var_account=InteractionData['FromIVRDATA']
1
2
3
4
Agent Group
R_CUSTOMER_PROCEDURE param=var_account
5
Default Destination

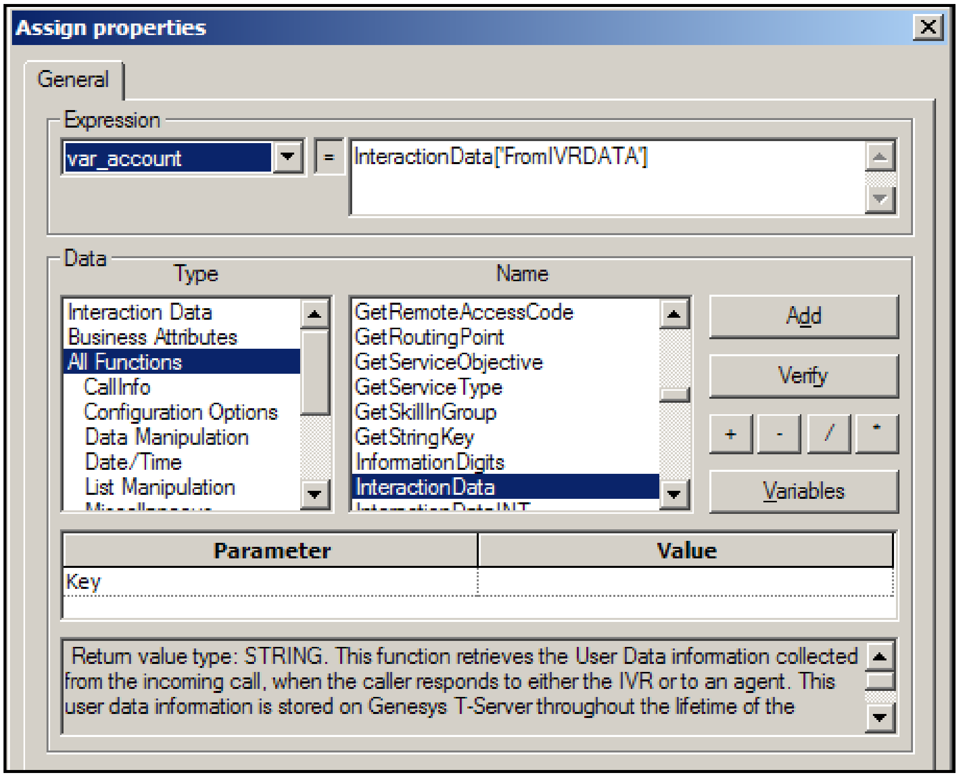
**Figure 61: Example Strategy Using Interaction Data**

### Summary of Flow

**Note:** The IRD objects that are described this section are keyed to the numbers in Figure 61 on page 59.

An IVR collects the caller's account number and stores it in the User Data structure of the call under the key `FromIVRDATA`.

1. An Assign object uses the InteractionData function to get the customer's account number from the key `FromIVRData` and assign it to a variable (see Figure 62).



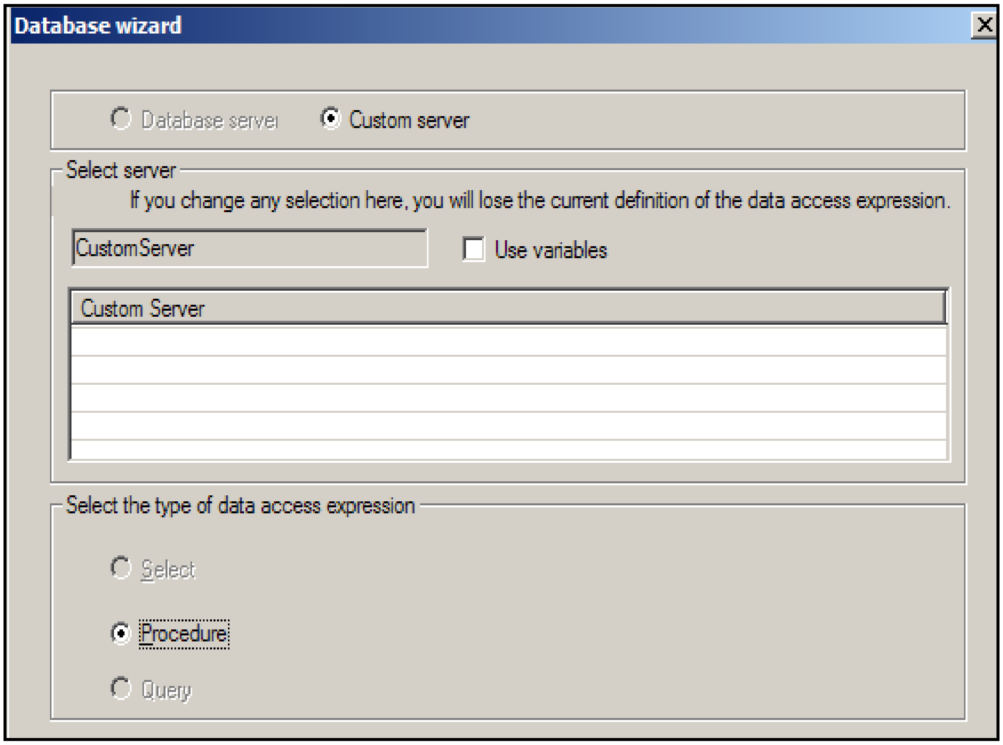
var_account = InteractionData['FromIVRDATA']

| Parameter | Value |
|---|---|
| Key | |

Return value type: STRING. This function retrieves the User Data information collected from the incoming call, when the caller responds to either the IVR or to an agent. This user data information is stored on Genesys T-Server throughout the lifetime of the

**Figure 62: Assign Object, Assigning Content of fromIVRData to Variable**

2. The first Database Wizard dialog box indicates that Custom Server and a custom procedure will get the information from the database. Genesys uses Custom Server to get information from non-SQL databases (see Figure 63).



Database wizard

○ Database server ● Custom server

Select server

If you change any selection here, you will lose the current definition of the data access expression.

CustomServer ☐ Use variables

Custom Server

Select the type of data access expression

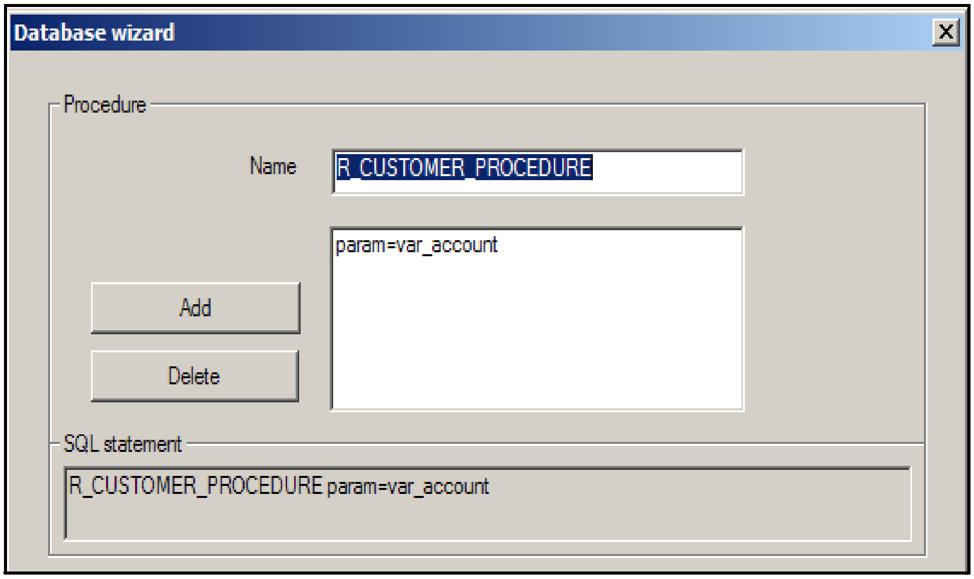
○ Select

● Procedure

○ Query

Figure 63: Database Object, Select Server

The next Database Wizard dialog box specifies the name of the Custom Server procedure. The account number contained in the `var_account` variable is specified as a parameter of the procedure (see Figure 64).



Database wizard

Procedure

Name: R_CUSTOMER_PROCEDURE

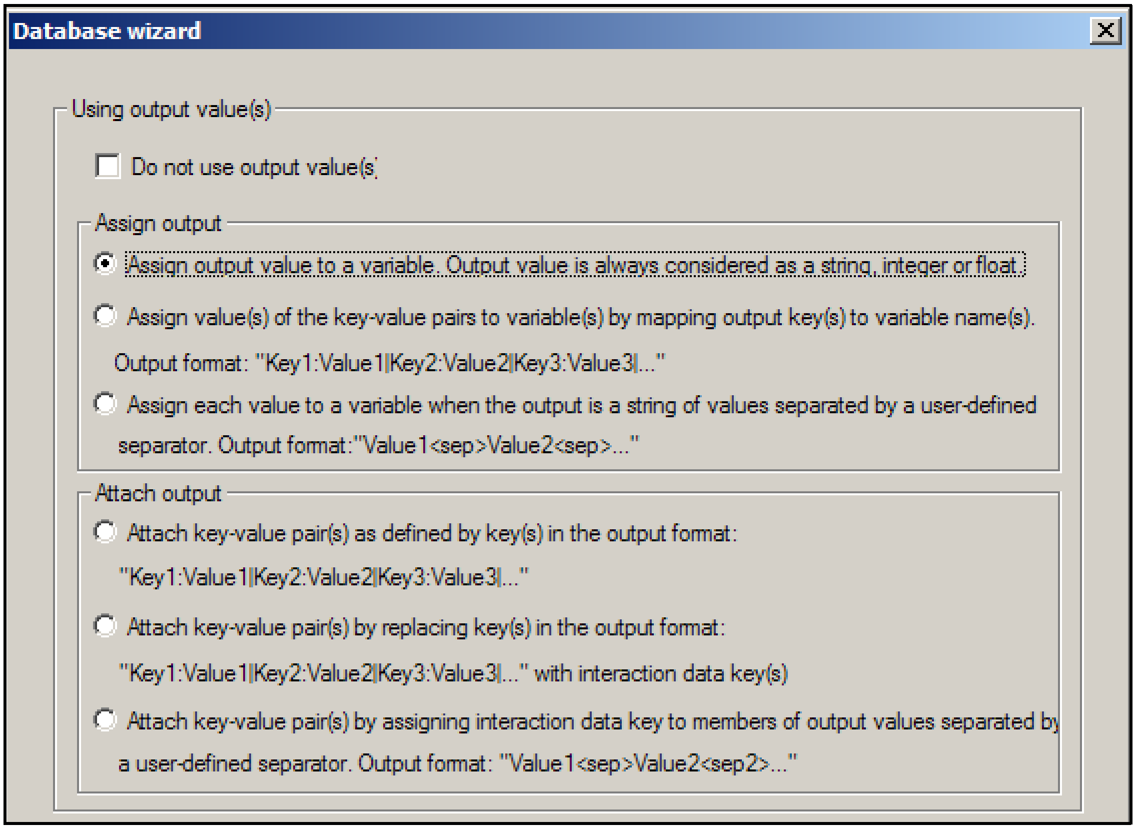
param=var_account

SQL statement

R_CUSTOMER_PROCEDURE param=var_account

Figure 64: Database Object, Procedure Specification

The next Database Wizard dialog box indicates that the output values of the procedure will be assigned to variables (see Figure 65).



Using output value(s)

□ Do not use output value(s

Assign output

● Assign output value to a variable. Output value is always considered as a string, integer or float.

○ Assign value(s) of the key-value pairs to variable(s) by mapping output key(s) to variable name(s).
Output format: "Key1:Value1|Key2:Value2|Key3:Value3|..."

○ Assign each value to a variable when the output is a string of values separated by a user-defined separator. Output format:"Value1<sep>Value2<sep>..."

Attach output

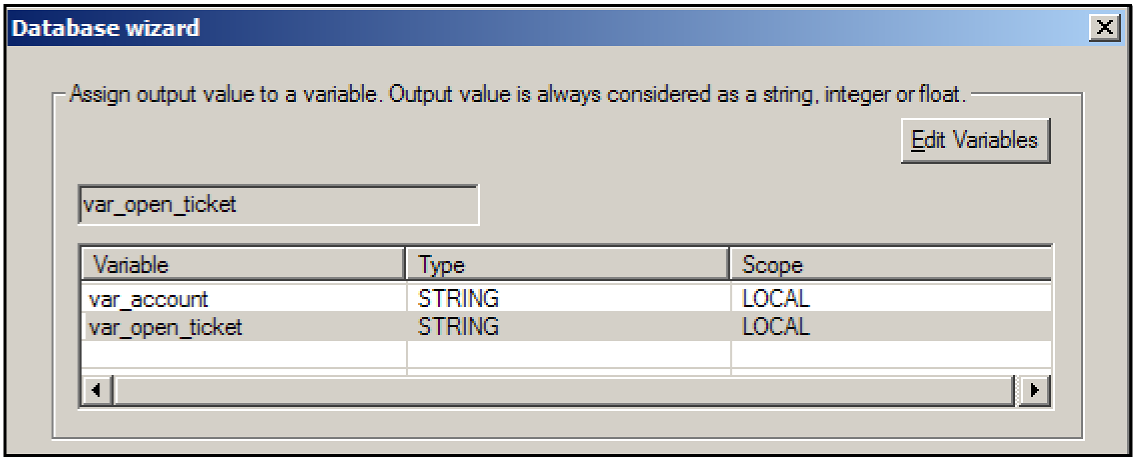
○ Attach key-value pair(s) as defined by key(s) in the output format:
"Key1:Value1|Key2:Value2|Key3:Value3|..."

○ Attach key-value pair(s) by replacing key(s) in the output format:
"Key1:Value1|Key2:Value2|Key3:Value3|..." with interaction data key(s)

○ Attach key-value pair(s) by assigning interaction data key to members of output values separated by a user-defined separator. Output format: "Value1<sep>Value2<sep2>..."

**Figure 65: Database Object, Using Output Value(s)**

The final Database Wizard dialog box names the variables (see Figure 66).



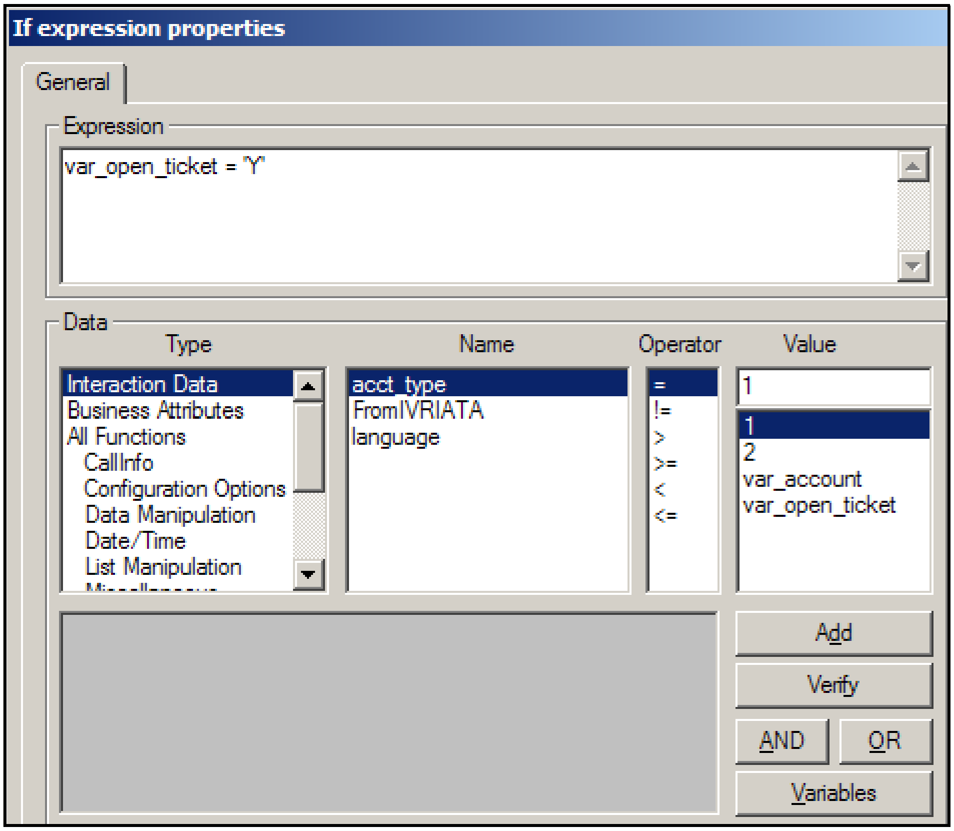
Assign output value to a variable. Output value is always considered as a string, integer or float.

var_open_ticket

| Variable | Type | Scope |
|---|---|---|
| var_account | STRING | LOCAL |
| var_open_ticket | STRING | LOCAL |

**Figure 66: Database Object, Assign Output Value to a Variable**

3. An expression in an If object is used to produce a yes-or-no answer to the question whether an open customer ticket is detected in a database record for this customer (see Figure 67).



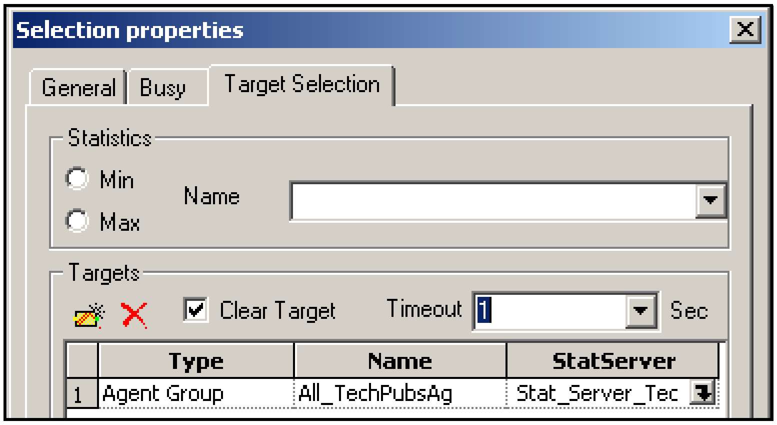
Expression: var_open_ticket = 'Y'

| Type | Name | Operator | Value |
|---|---|---|---|
| Interaction Data<br>Business Attributes<br>All Functions<br>CallInfo<br>Configuration Options<br>Data Manipulation<br>Date/Time<br>List Manipulation | acct_type<br>FromIVRIATA<br>language | =<br>!=<br>><br>>=<br><<br><= | 1<br>1<br>2<br>var_account<br>var_open_ticket |

**Figure 67: If Expression Properties Dialog Box**

The answer is returned in the format `Y` for affirmative and `N` for negative.

4. If the answer is `Y`, indicating that an open ticket is detected in a database record for this customer, the interaction goes out the green port to a Routing Selection object, which routes to an agent group (see Figure 68).



☑ Clear Target

Timeout 1 Sec

|  | Type | Name | StatServer |
|---|---|---|---|
| 1 | Agent Group | All_TechPubsAg | Stat_Server_Tec |

**Figure 68: Routing Selection Object**

5. If the answer is `N`, indicating that an open ticket is not detected in a database record for this customer, the interaction goes out the red port to the Default Destination object.

Default routing also occurs if the procedure used in the Database object does not return a value or if the Routing Selection object cannot route to the specified agent group.

**Note:** No dialog box opens for the Default destination object. You set the default destination in the `default _destination` option in the URS Application object.

# Strategy 7: Using Subroutines

This strategy demonstrates routing to various types of targets. It is a more complicated strategy than those presented so far.

## Targets Used

- Groups of agents: `Boston_brokers@Boston_statserver.GA`, `Boston_GeneralInquiry@Boston_statserver.GA`, and `SanFran_Cust_Service@Boston_statserver.GA`.
- An IVR configured as a group of places: `BOSIVR@Boston_statserver.GP`.

## Scenarios

- If the call arrives outside of the normal business hours of 8:00 a.m. and 8:00 p.m., the strategy routes it to a place group (`BOSIVR`).
- If the call arrives during normal business hours between 8:00 a.m. and 8:00 p.m., it is routed to an agent, if possible. If this is not possible, the call is routed to an IVR overflow target. The assumption is that URS operates on a machine in the Eastern Standard time zone. All other times are in the local business hours of the Boston or San Francisco office.

  During business hours, the strategy retrieves from a database the account number corresponding to the originating phone number (the keyword ANI provides access to the Automatic Number Identifier passed by T-Server in the Call Information structure). This number is retrieved from the field `AcctNum` in the table named `Customers` by matching the field `Telephone` with the ANI of the call. The Database Access Point `Boston_DAP` has access to the database to which the table belongs.The retrieved account number is stored in a variable named `AcctNum`.

  - If an account number is successfully retrieved, then the call is routed to the agent group `Boston_brokers` between 8:00 a.m. and 5:00 p.m. and to the agent group `SanFran_Cust_Service` between 5:00 p.m. and 8:00 p.m.
  - If no account number is retrieved, the call is routed to the agent group `Boston_GeneralInquiry`.

## IVR Target

The IVR is used as an overflow target as well as a target outside business hours. Routing Selection objects specify the sequence of actions when the IVR is used as an overflow target after a call fails to be routed to an agent group target. This happens when the specified waiting time elapses before the call can be routed to an agent from the group.

## The Strategy in the Routing Design Window

Figure 69 shows the `str_using_subroutines` strategy in the IRD Routing Design window.

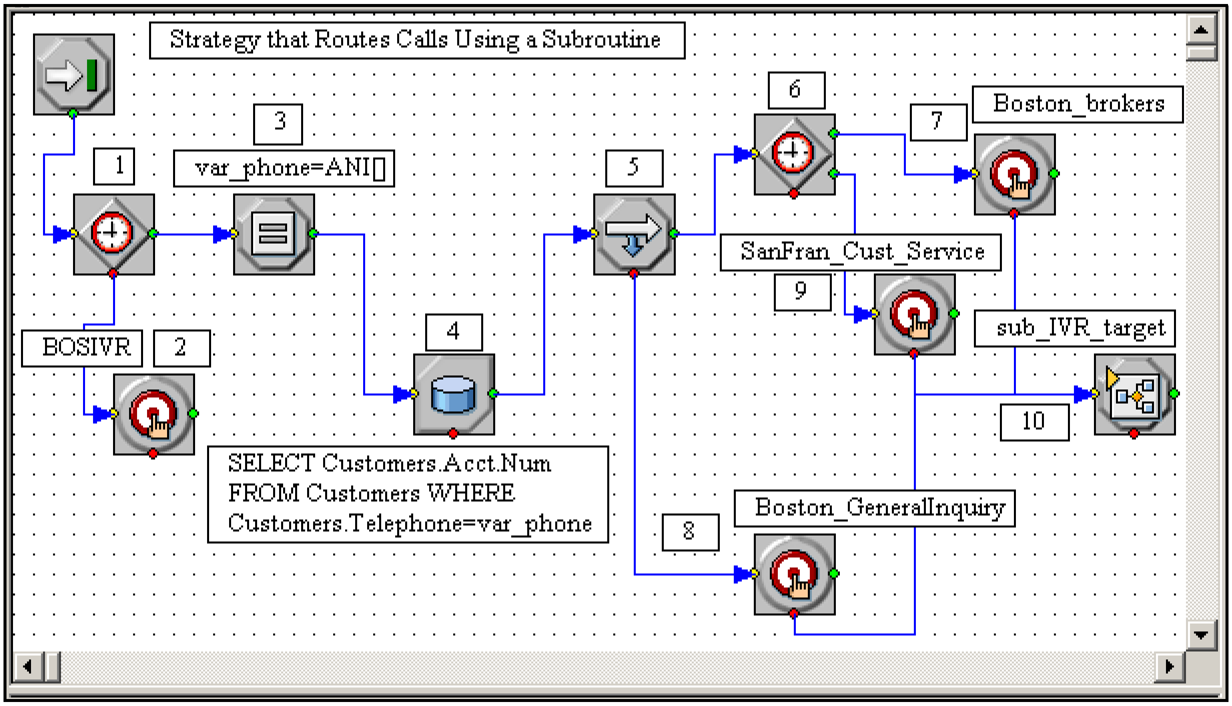
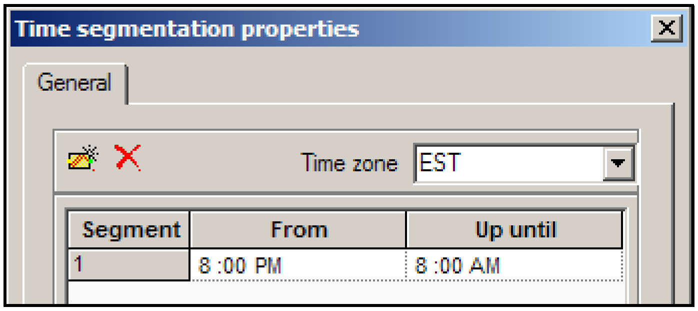


**Figure 69: Example Strategy Using Subroutines**

### Summary of Flow

**Note:** The IRD objects that are described this section are keyed to the numbers in Figure 69.

1. A Time Segmentation object causes incoming calls that arrive during Eastern Standard time zone business hours to go out the green port to an Assign object (see Figure 70).



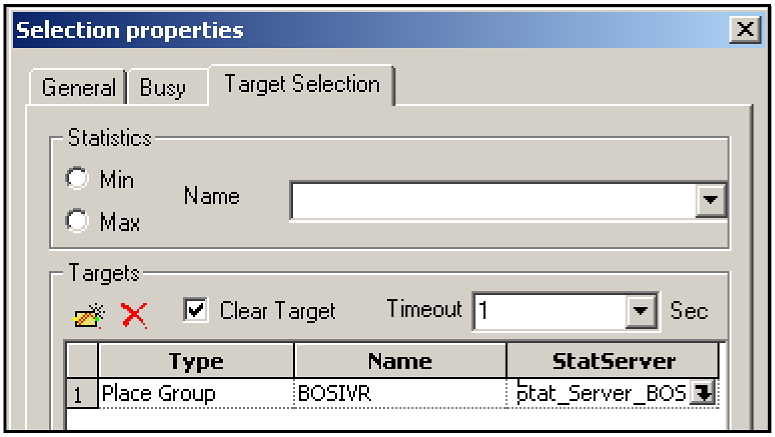
Time zone EST

| Segment | From | Up until |
|---|---|---|
| 1 | 8 :00 PM | 8 :00 AM |

**Figure 70: Time Segmentation Object, Eastern Standard Time**

If an incoming call arrives outside of business hours, it goes out the red error port to a Routing Selection object.

2. The Routing Selection object routes incoming calls that arrive outside of business hours to an IVR called `BOSIVR` place group (see Figure 71).



Statistics

○ Min

○ Max

Name

Targets

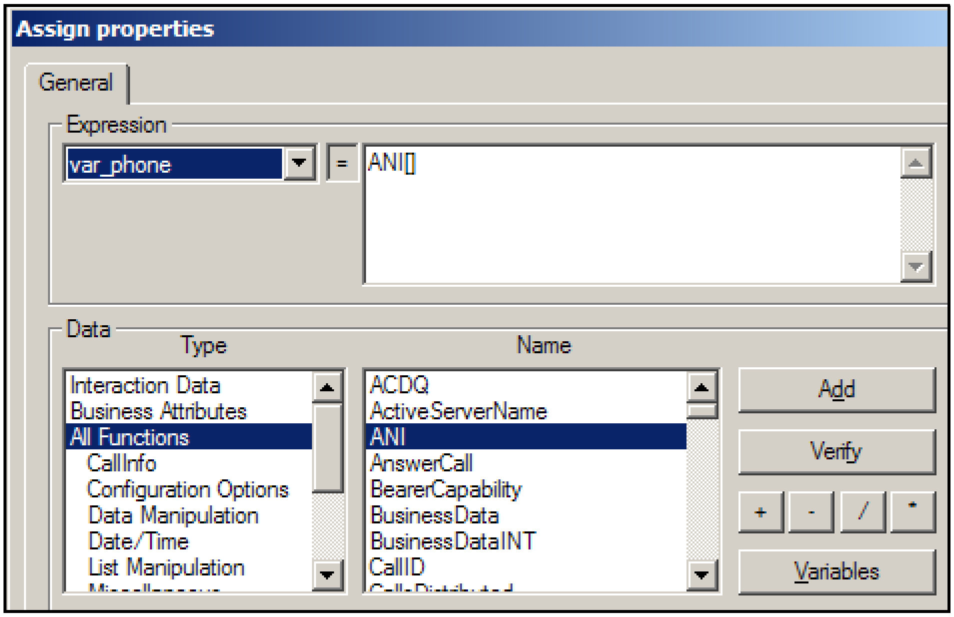
☑ Clear Target

Timeout 1 Sec

|  | Type | Name | StatServer |
|---|---|---|---|
| 1 | Place Group | BOSIVR | Stat_Server_BOS |

**Figure 71: Routing Selection Object**

3. If the call arrives during normal business hours, the Assign object assigns the originating phone number (ANI) to a variable (see Figure 72).



Assign properties

General

Expression

var_phone = ANI[]

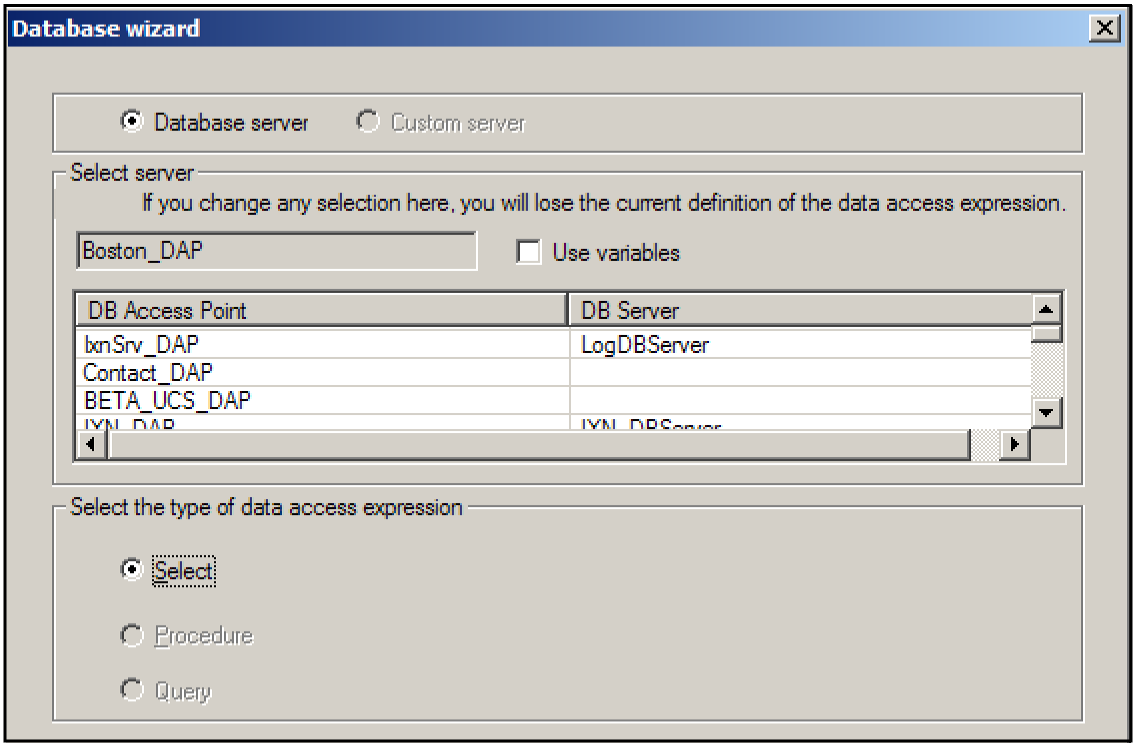
Data

| Type | Name |
|---|---|
| Interaction Data | ACDQ |
| Business Attributes | ActiveServerName |
| All Functions | ANI |
| CallInfo | AnswerCall |
| Configuration Options | BearerCapability |
| Data Manipulation | BusinessData |
| Date/Time | BusinessDataINT |
| List Manipulation | CallID |

**Figure 72: Assign Object, ANI Function**

The call then goes out the green port to a Database Wizard object.

4. The first dialog box in the Database Wizard specifies a Database Server (used for SQL databases). See Figure 73.



Database wizard

● Database server ○ Custom server

Select server

If you change any selection here, you will lose the current definition of the data access expression.

Boston_DAP ☐ Use variables

| DB Access Point | DB Server |
|---|---|
| IxnSrv_DAP | LogDBServer |
| Contact_DAP | |
| BETA_UCS_DAP | |

Select the type of data access expression

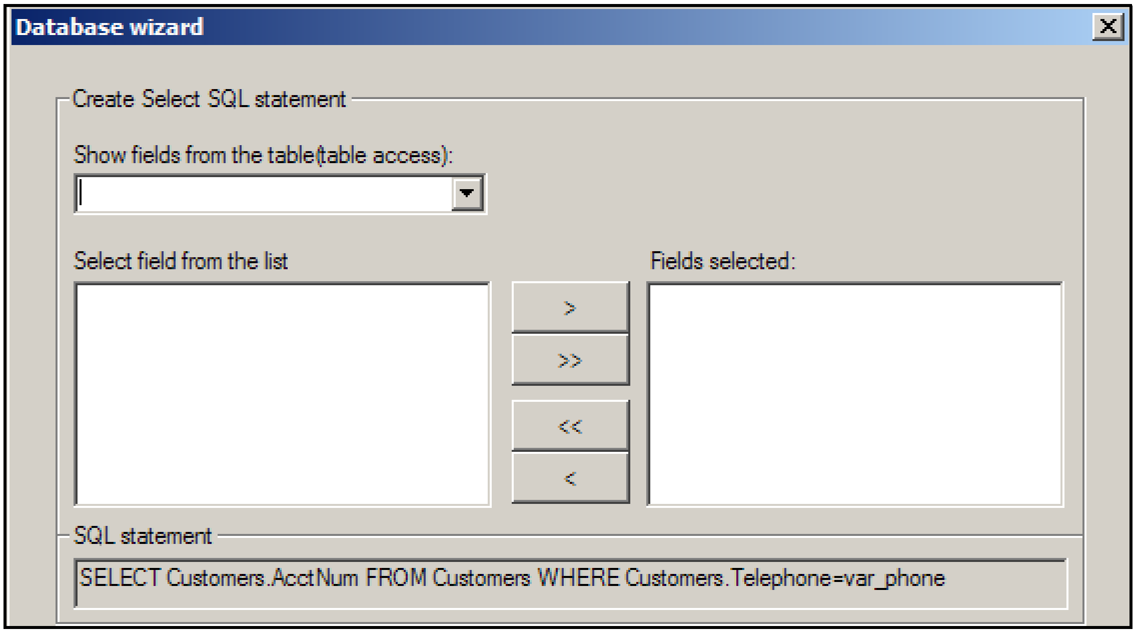
● Select

○ Procedure

○ Query

**Figure 73: Database Wizard, Type of Server and Data Access Point**

The next Database Wizard dialog box starts the SQL `SELECT` statement (see Figure 74).



Database wizard

Create Select SQL statement

Show fields from the table(table access):

Select field from the list

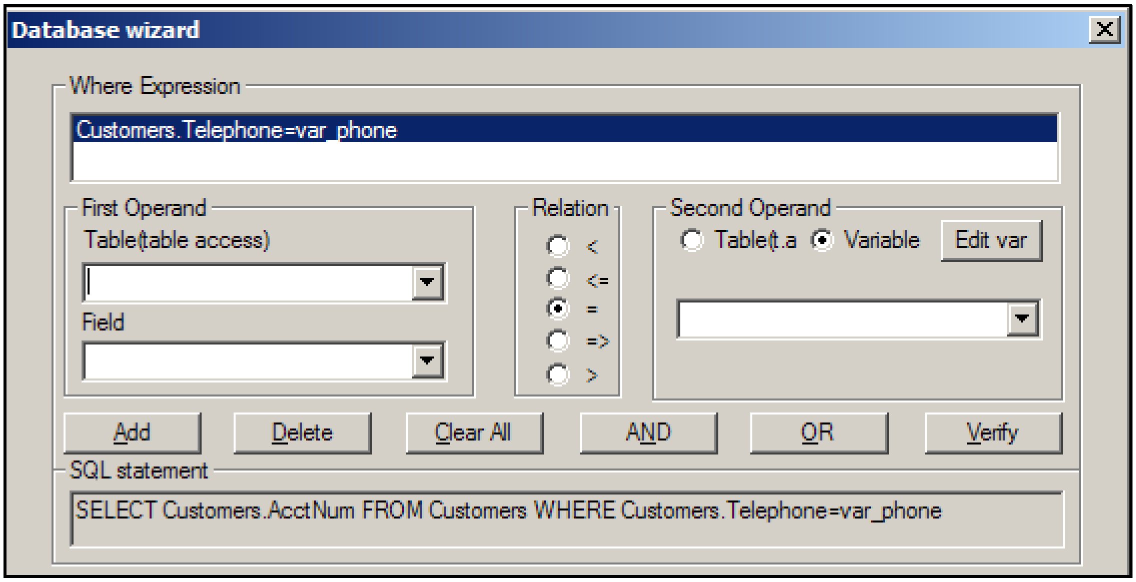
Fields selected:

SQL statement

SELECT Customers.AcctNum FROM Customers WHERE Customers.Telephone=var_phone

Figure 74: Database Wizard, Create SQL Select Statement

The next Database Wizard dialog box creates the `WHERE` clause (see Figure 75).



Database wizard

Where Expression

Customers.Telephone=var_phone

First Operand

Table(table access)

Field

Relation

○ <
○ <=
● =
○ =>
○ >

Second Operand

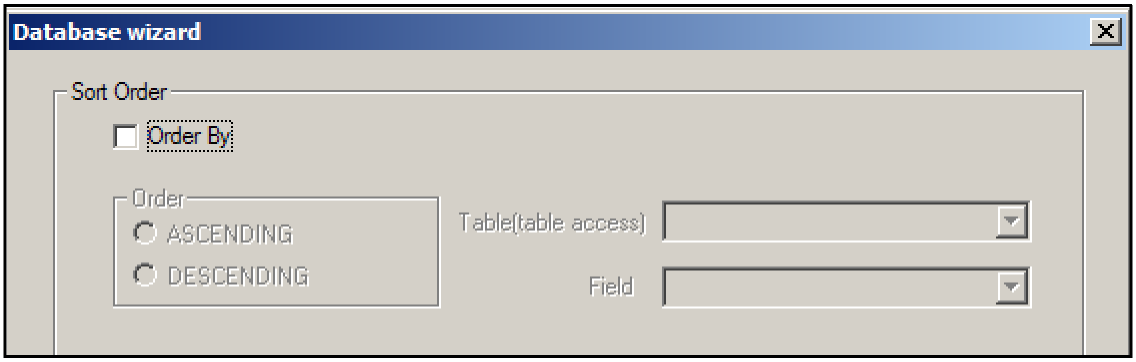
○ Table(t.a ● Variable

SQL statement

SELECT Customers.AcctNum FROM Customers WHERE Customers.Telephone=var_phone

Figure 75: Database Wizard, Where Expression

The next Database Wizard dialog box enables you to specify a sort order. This example does not use a sort order (see Figure 76).



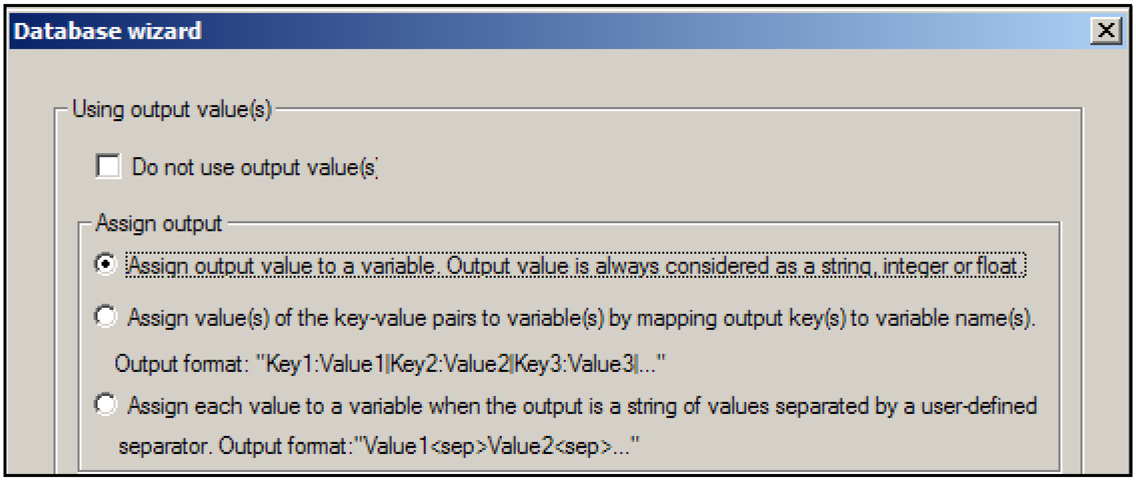
☐ Order By

○ ASCENDING

○ DESCENDING

**Figure 76: Database Wizard, Sort Order**

The next Database Wizard dialog box assigns the output from the database to a variable (see Figure 77).



☐ Do not use output value(s

● Assign output value to a variable. Output value is always considered as a string, integer or float.

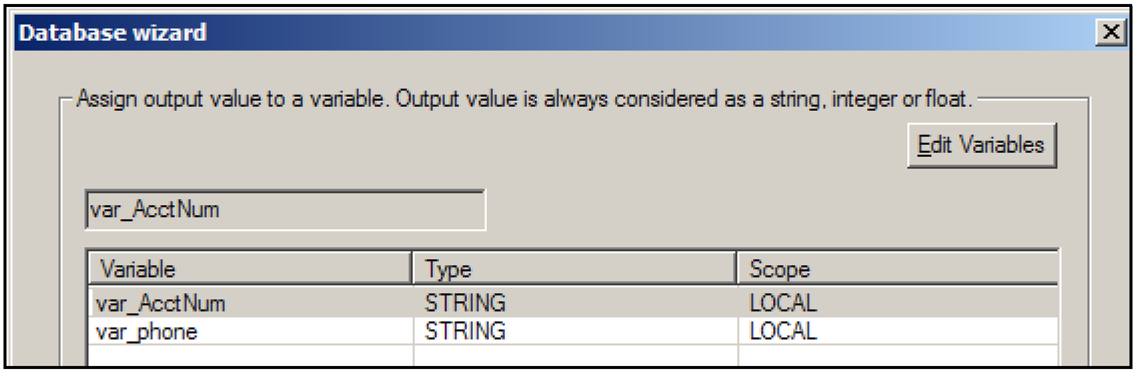
○ Assign value(s) of the key-value pairs to variable(s) by mapping output key(s) to variable name(s).

Output format: "Key1:Value1|Key2:Value2|Key3:Value3|..."

○ Assign each value to a variable when the output is a string of values separated by a user-defined separator. Output format:"Value1<sep>Value2<sep>..."

**Figure 77: Database Wizard, Using Output Values**

The last Database Wizard dialog box defines the variables (see Figure 78).



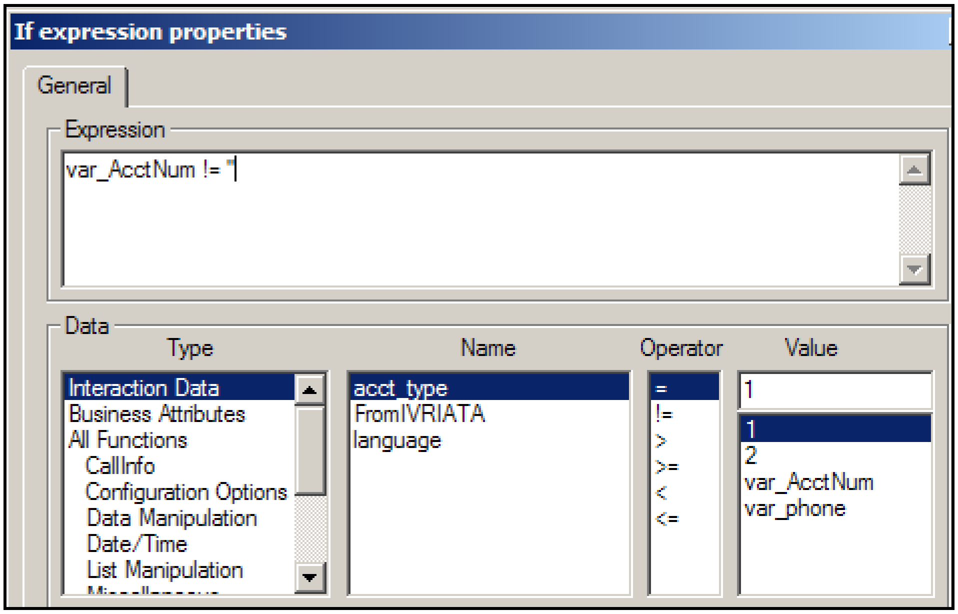
Assign output value to a variable. Output value is always considered as a string, integer or float.

var_AcctNum

| Variable | Type | Scope |
|---|---|---|
| var_AcctNum | STRING | LOCAL |
| var_phone | STRING | LOCAL |

**Figure 78: Database Wizard, Assign Output Value to a Variable**

**5.** An If object creates an expression used to determine whether an account number was successfully retrieved from the database. If the account number was successfully retrieved, the variable `var_AcctNum` contains the account number (see Figure 79).



If expression properties

General

Expression

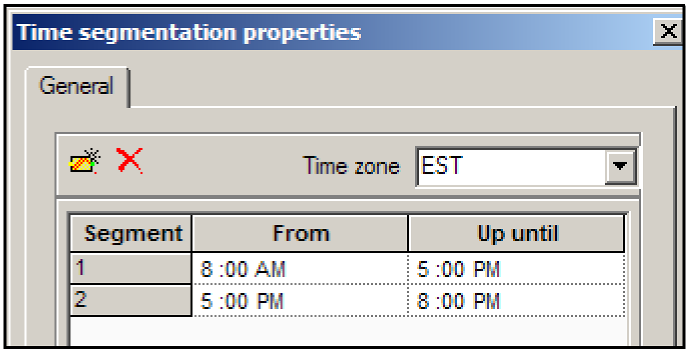
var_AcctNum != ''

Data

| Type | Name | Operator | Value |
|---|---|---|---|
| Interaction Data | acct_type | = | 1 |
| Business Attributes | FromIVRIATA | != | 1 |
| All Functions | language | > | 2 |
| CallInfo | | >= | var_AcctNum |
| Configuration Options | | < | var_phone |
| Data Manipulation | | <= | |
| Date/Time | | | |
| List Manipulation | | | |

Figure 79: If Expression Properties Dialog Box Using var_AcctNum

**6.** If an account number was successfully retrieved from the database, the interaction goes out the green port to a Time Segmentation object. The first row in the Time segmentation properties dialog box corresponds to normal business hours. The second row is used for calls that arrive outside of business hours (see Figure 80).



Time segmentation properties

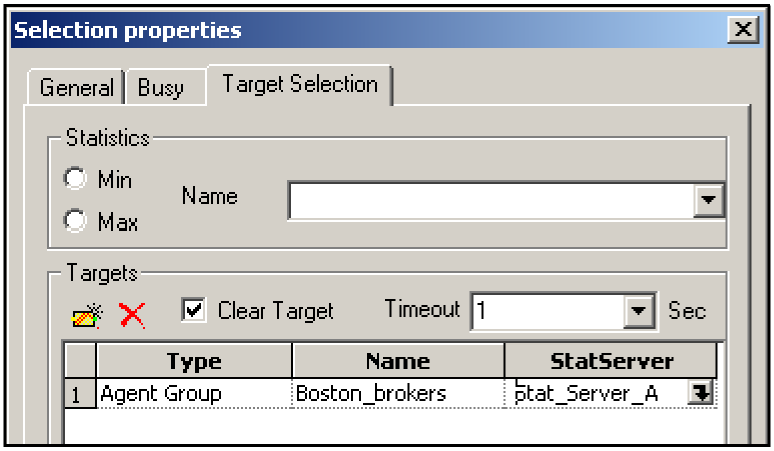
General

Time zone EST

| Segment | From | Up until |
|---|---|---|
| 1 | 8 :00 AM | 5 :00 PM |
| 2 | 5 :00 PM | 8 :00 PM |

Figure 80: Time Segmentation Object

**7.** Calls that arrive between 8:00 a.m. and 5:00 p.m. go to a Routing Selection object that specifies the `Boston_brokers` agent group (see Figure 81).



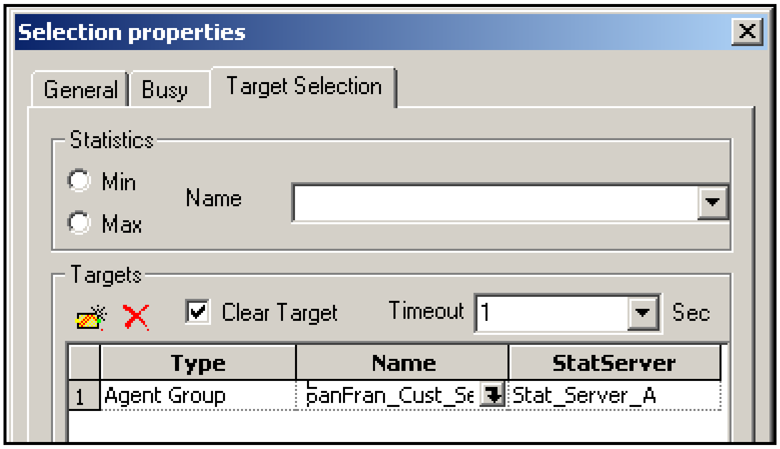
○ Min ○ Max Name

☑ Clear Target Timeout 1 Sec

|   | Type | Name | StatServer |
|---|---|---|---|
| 1 | Agent Group | Boston_brokers | Stat_Server_A |

**Figure 81: Routing Selection Object, Routing to Boston_brokers**

8. Calls that arrive between 5:00 p.m. and 8:00 p.m. go to a Routing Selection object that specifies the `SanFran_Cust_Service` agent group (see Figure 82).



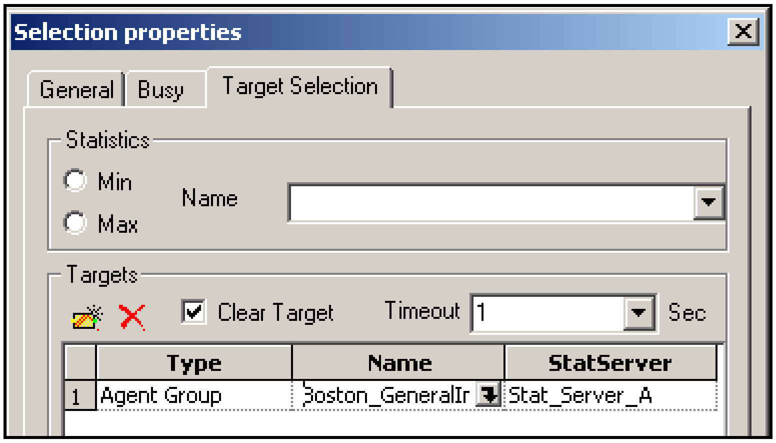
○ Min ○ Max Name

☑ Clear Target Timeout 1 Sec

|   | Type | Name | StatServer |
|---|---|---|---|
| 1 | Agent Group | SanFran_Cust_Se | Stat_Server_A |

**Figure 82: Routing Selection Object, Routing to SanFran_Cust_Service**

9. If an account number was not successfully retrieved from the database (see Figure 79 on page 70), the call goes out the red error port to a Routing Selection object that specifies the `Boston_GeneralInquiry` agent group (see Figure 83).



Statistics

○ Min

○ Max

Name

Targets

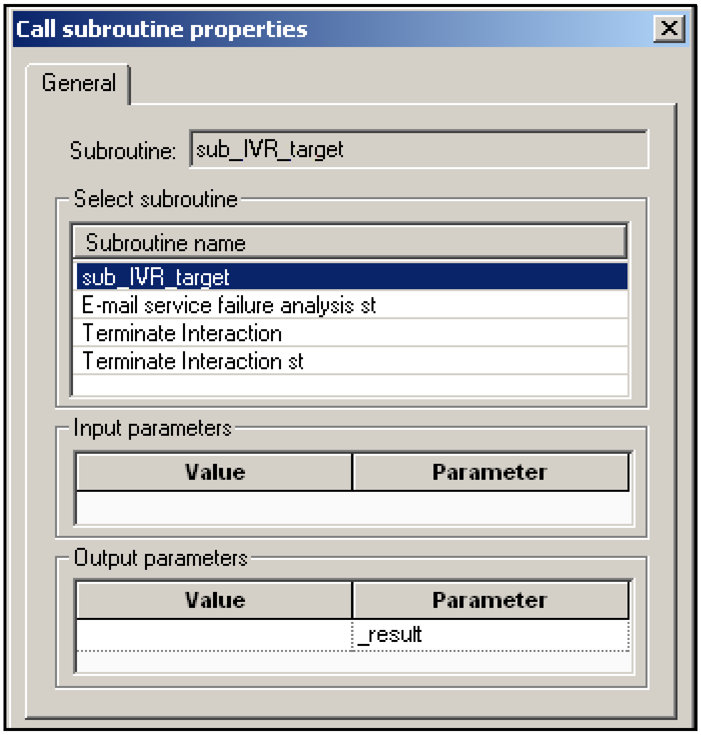
☑ Clear Target

Timeout 1 Sec

|   | Type | Name | StatServer |
|---|---|---|---|
| 1 | Agent Group | Boston_GeneralIr | Stat_Server_A |

Figure 83: Routing Selection Object, Boston_GeneralInquiry

10. If a call is not successfully routed to any of the above agent groups, they are routed to a Call Subroutine object, which specifies an IVR target (see Figure 84).



Subroutine: sub_IVR_target

Select subroutine

| Subroutine name |
|---|
| sub_IVR_target |
| E-mail service failure analysis st |
| Terminate Interaction |
| Terminate Interaction st |

Input parameters

| Value | Parameter |
|---|---|

Output parameters

| Value | Parameter |
|---|---|
|  | _result |

Figure 84: Call Subroutine Properties Dialog Box

Chapter

# 3 Multimedia Samples

This chapter describes multimedia (non-voice) business process samples that are supplied in the Genesys Interaction Workflow Samples component.

- The *eServices (Multimedia) 8.0 Deployment Guide* describes how to install the Interaction Workflow Samples.
- The *Universal Routing 8.1 Deployment Guide* describes how to have the Interaction Design shortcut bar appear in IRD, if it has not appeared automatically.
- The *Universal Routing 8.1 Business Process User's Guide* provides an in-depth discussion of business processes.
- The *Universal Routing 8.1 Interaction Routing Designer Help* describes how to create, save, import and export a business process, and how to load the strategies that comprise the business process.

### About the Graphics

Some graphics showing business processes and strategies in this chapter have been arranged to make all the included objects fit in the available space. When actually creating business processes and strategies, Genesys recommends that you allow adequate space between the objects for clarity. The actual workspaces provided by IRD are much larger than the areas shown in these graphics.

### About Server Names in Object Properties Dialog Boxes

No server is selected in objects appearing in these sample strategies in which you have the option to select a specific server to handle interactions. This was done deliberately to ensure that the samples will function in any environment. If you specify a server name, URS or Interaction Server (depending on the object) searches for the specific server and returns an error if it cannot find it.

If you do not select a specific server, URS or Interaction server looks at its connections list and sends interactions to all servers of the correct type, using load balancing to determine how many interactions to send to each server.

When you create actual strategies in your environment, you might prefer to specify which server that should handle interactions for each strategy.

**Note:** For detailed information on business processes, see the *Universal Routing 8.1 Business Process User's Guide*.

This chapter contains the following sections:

# Business Processes and Strategies

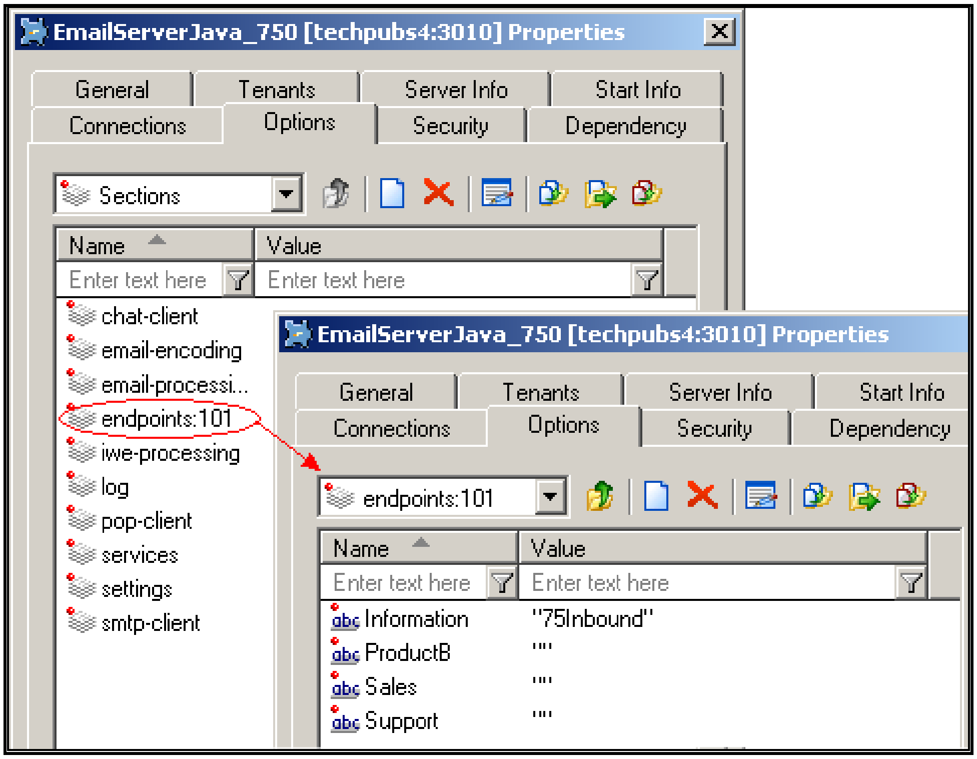
A *business process* directs customer interactions arriving at the contact center through various processing objects, including *routing strategies*. It controls what happens to customer interactions from the point of arrival to the point of completion. An *interaction workflow* comprises a group of business processes.

The types of processing applied to interactions varies based on the media type and the contact center's business logic. In all cases, the goal is to generate an appropriate response for the customer.

- In the case of an e-mail interaction, an appropriate response might be an e-mail answering the customer's questions.
- In the case of a chat interaction, an appropriate response might be mailing product brochures to the customer.
- In the case of a fax interaction, an appropriate response might be an e-mail stating the requested materials had been received, and so on.

In general, a business process works as follows:

- Interaction Server places an inbound interaction in a queue. In the case of an e-mail interaction, the appropriate initial inbound queues are defined using the `endpoints:<tenant_DBID>` section in the E-mail Server Application object (see Figure 85):



EmailServerJava_750 [techpubs4:3010] Properties

Sections

- chat-client
- email-encoding
- email-processi...
- endpoints:101
- iwe-processing
- log
- pop-client
- services
- settings
- smtp-client

EmailServerJava_750 [techpubs4:3010] Properties

endpoints:101

| Name | Value |
|---|---|
| Information | "75Inbound" |
| ProductB | "" |
| Sales | "" |
| Support | "" |

**Figure 85: E-mail Server Endpoint Configuration**

- The interaction is then taken out of the queue and submitted to a routing strategy that uses various processing objects.
- A routing strategy performs the processing specified and eventually routes the interaction to a target, but not necessarily the final target. For example, an e-mail interaction may be placed in an agent queue for construction of a response.
- The target processes the interaction and places it into another queue where another strategy may process it. For example, a strategy may send the interaction to a queue for Quality Assurance checking.
- The cycle of going from queue to routing strategy to queue continues until processing is stopped or the interaction reaches some final (outbound) queue.

## Conceptual Diagram

Figure 86 shows a conceptual diagram of an inbound e-mail business process that ends in Outbound Quality Assurance (QA) review.

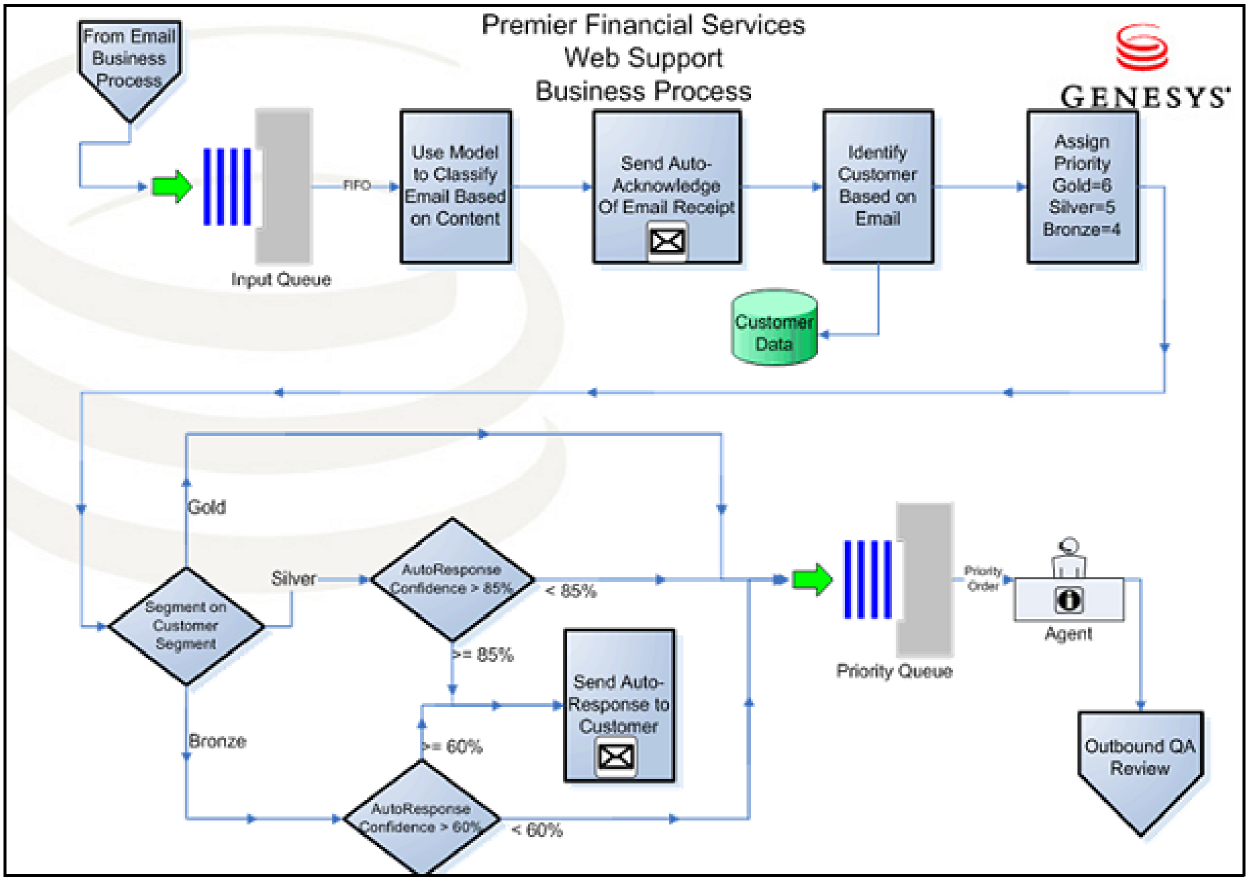

Premier Financial Services
Web Support
Business Process
GENESYS
From Email Business Process
FIFO
Input Queue
Use Model to Classify Email Based on Content
Send Auto-Acknowledge Of Email Receipt
Identify Customer Based on Email
Assign Priority Gold=6 Silver=5 Bronze=4
Customer Data
Gold
Silver
Segment on Customer Segment
AutoResponse Confidence > 85%
< 85%
>= 85%
Send Auto-Response to Customer
>= 60%
Bronze
AutoResponse Confidence > 60%
< 60%
Priority Order
Priority Queue
Agent
Outbound QA Review

**Figure 86: Business Process Conceptual Diagram: Inbound E-mail**

# Sample Business Process

An actual business process created in Interaction Routing Designer resembles a diagram. Figure 87 shows an example business process that forwards a chat transcript to a customer.

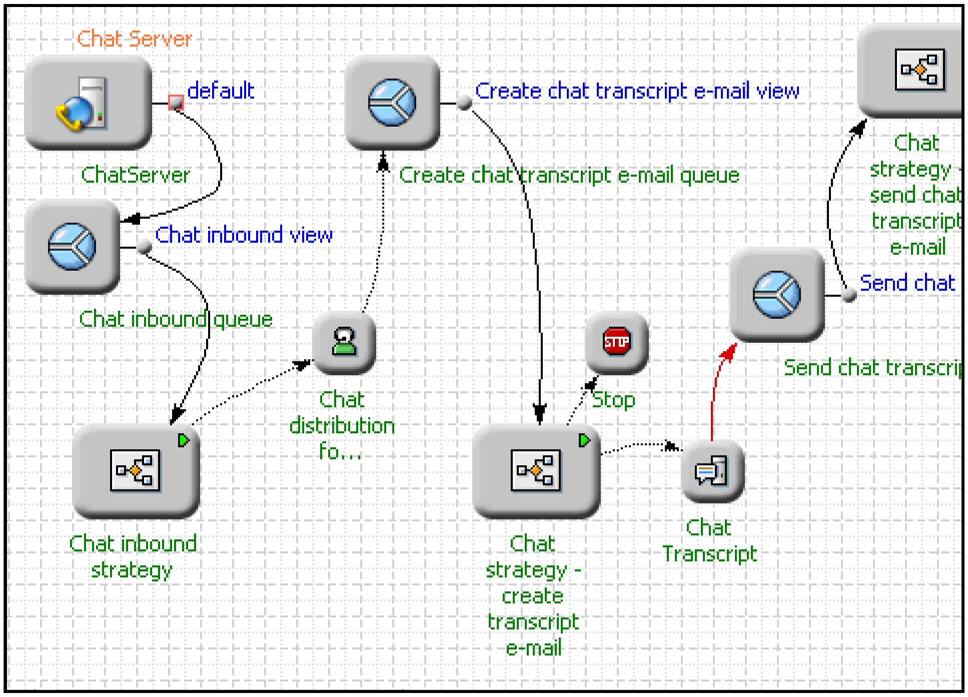


**Figure 87: Example Business Process for Forwarding Chat Transcript**

Objects of several types, together with their properties, make up a business process created in IRD's Interaction Design window:

- The primary objects are *Queue* (rounded squares in Figure 87), *View* (single small circles in Figure 87), and *Strategy* (rounded rectangles in Figure 87) objects.
- Some objects— output queues, agents, agent groups, workbins, and servers—appear because they are specified as targets in routing strategies. These are strategy-linked nodes, as described in the *Universal Routing 8.1 Business Process User's Guide*.
- A *Stop* node appears if a routing strategy contains a Stop object.
- Curved lines from queues to routing strategies represent *Submitter* processes.
- Curved lines flowing out from routing strategies point to targets (smaller nodes, such as `Chat Transcript` in Figure 87).

## Interaction Server

Interaction Server executes business processes and in doing so, communicates with Universal Routing Server (URS), which executes the routing strategies contained in business processes.

When Interaction Server executes a business process, interactions are placed in queues, which are represented by Queue objects. Interaction Server then selects interactions from a queue for further processing, based on the selection criteria specified in the View object associated with the Queue object. Submitter objects represent the process of moving an interaction to a Strategy object (routing strategy).

**Note:** When multiple routing strategies are used in a business process, you may use queues as "stepping stones" between strategies. For more on this, see Appendix A in *Universal Routing 8.1 Business Process User's Guide*.

# Objects Used in This Chapter

Table 2 lists strategy-building objects (see page 28) used in multimedia routing.

**Table 2: IRD Objects Used in the Samples**

| Multimedia objects (page 31) | Routing objects (page 30) | Segmentation objects (page 30) | Miscellaneous objects (page 31) |
|---|---|---|---|
| Acknowledgement (see pages 102, 181, 194, 207) | Route Interaction (see pages 94, 123, 127, 147, 158, 161, 168, 184, 190) | Screen Segmentation (see pages 184, 194, 207) | Assign (see pages 94, 127, 147, 175, 177, 180, 190) |
| Autoresponse (see pages 102, 194, 207) | Workbin (see pages 184, 190) | Generic Segmentation (see pages 89, 102, 112, 115, 119, 127, 138, 142, 180, 181, 184) | MultiAssign (see pages 89, 94, 127, 168, 175, 180, 190) |
| Chat Transcript (see page 152) | Queue Interaction (see pages 89, 94, 102, 115, 127, 138, 142, 175, 180, 184, 190, 194, 200, 207) | | Function (see pages 102, 112, 115, 119, 127, 138, 147, 152, 155, 184) |

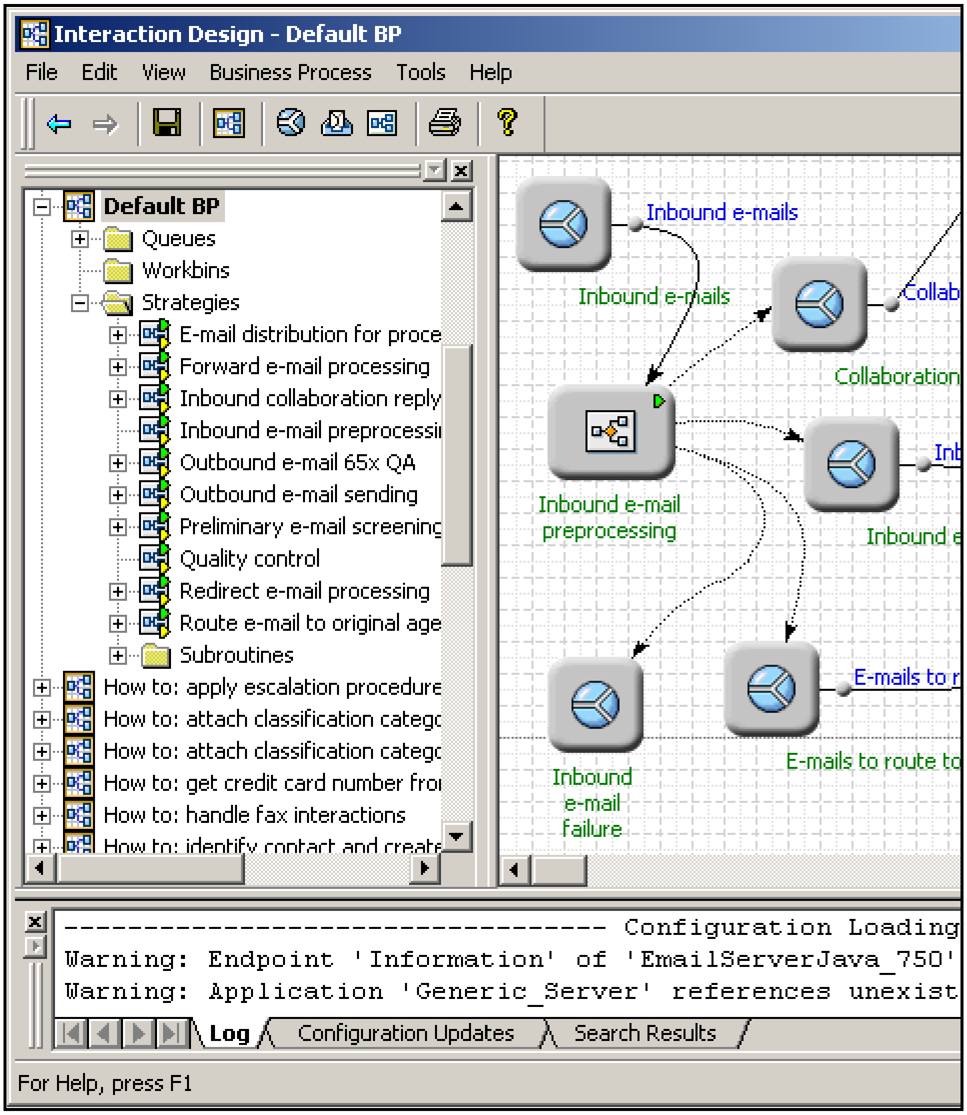
### Table 2: IRD Objects Used in the Samples (Continued)

| Multimedia objects (page 31) | Routing objects (page 30) | Segmentation objects (page 30) | Miscellaneous objects (page 31) |
|---|---|---|---|
| Create Interaction (see pages 158, 161, 167, 200) | | | If (see pages 94, 102, 127, 138, 147, 152, 167, 168, 169, 170, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 184, 190, 200) |
| Forward E-Mail (see page 115) | | | Call Subroutine (see pages 102, 112, 115, 119, 138, 142, 167, 168, 169, 170, 171, 172, 173, 174, 176, 177, 178, 180, 184) |
| Identify Contact (see pages 167, 200) | | | |
| MultiScreen (see pages 184, 194, 207) | | | |
| Redirect E-Mail (see page 112) | | | |
| Reply E-Mail From External Resource (see page 119) | | | |
| Screen (see pages 102, 138, 184) | | | |
| Send E-Mail (see pages 142, 155) | | **Data and Services objects (page 30)** | |
| Send SMS Out (see page 163) | | External Service (see page 147) | |
| Stop Interaction (see pages 102, 112, 115, 119, 142, 152, 155, 164, 173, 182, 194, 207) | | | |

In addition to the objects in Table 2, a routing strategy that is called by a business process can use the Database Wizard object that is discussed in “Strategy 5: Database Lookup” on page 51.

# Interaction Workflow Samples

If you install the Genesys eServices (formerly Multimedia) software components (as described in the *eServices (Multimedia) 8.0 Deployment Guide)*, you have the option of installing a component called *Interaction Workflow Samples.* Figure 88 shows the Interaction Design window when the `Strategies` folder of one of the supplied business processes, `Default BP`, is expanded.


Interaction Design - Default BP
File
Edit
View
Business Process
Tools
Help
Default BP
Queues
Workbins
Strategies
E-mail distribution for proce
Forward e-mail processing
Inbound collaboration reply
Inbound e-mail preprocessi
Outbound e-mail 65x QA
Outbound e-mail sending
Preliminary e-mail screening
Quality control
Redirect e-mail processing
Route e-mail to original age
Subroutines
How to: apply escalation procedure
How to: attach classification catego
How to: attach classification catego
How to: get credit card number fro
How to: handle fax interactions
How to: identify contact and create
Inbound e-mails
Inbound e-mails
Collab
Collaboration
Inbound e-mail preprocessing
Inb
Inbound e
E-mails to r
E-mails to route to
Inbound e-mail failure
Log
Configuration Updates
Search Results
For Help, press F1

```
------------------------------- Configuration Loading
Warning: Endpoint 'Information' of 'EmailServerJava_750'
Warning: Application 'Generic_Server' references unexist
```

**Figure 88: Interaction Workflow Samples in the Interaction Design Window**

Because each company has a specific environment and different requirements, implementing business processes that accurately reflect all real-word business scenarios is impossible. Therefore, do not consider these samples suitable for a production environment, but instead as demonstrative samples that contain a set of common functionality. Use them as a starting point to develop your own customized business processes that are adjusted to your company's needs.

Through its Multimedia Interaction Workflow Samples component, Genesys provides the following sample business processes:

- ABC Simple BP is a basic business process useful to get the general concept of business processes and check the samples installation.
- ABC Simple Chat BP offers minimal chat interaction processing.
- ABC Simple MMS offers minimal inbound MMS (Multimedia Messaging Service) processing.
- ABC Simple SMS Paging offers minimal SMS (Short Messaging Service) processing.
- WebCallback offers web callback processing.
- Default BP is a complex business process that incorporates many types of interaction handling. See "Sample Functionalities" on page 82.
- The 10 samples beginning with Step 0. Common Components and ending with Step 4. Outbound sending represent a breakdown of Default BP into functionally-themed components.
- How to: apply escalation procedure
- How to: attach classification categories and use the Attach Categories object
- How to: attach classification categories and use the Multi-Screen object
- How to: get credit card number from the e-mail
- How to: handle fax interactions
- How to: identify contact and create interaction in UCS
- How to: place the interaction in the workbin
- How to: screen with multiple rules and use the screening switch

## Routing Outbound Open Media Interactions

The Outbound toolbar in the IRD Routing Design window includes a number of objects that enable you to route outbound open media interactions that are of the outbound_preview type. These objects are used for proactive routing.

For detailed information about proactive routing, including strategy samples that demonstrate the use of the Outbound toolbar objects, see the *Genesys 7.6 Proactive Routing Solution Guide*. The strategy samples appear in the Appendix to this document.

## Sample Functionalities

As described in Appendix A of the *Universal Routing 8.1 Business Process User's Guide*, the `DefaultBP` business process that is included with the Interaction Workflow Samples component supplies the following functionalities:

- Pre-routing based on interaction sub-type
- Routing interactions to the original agent
- Screening of inbound interactions
- Attaching classification categories
- Processing of attached data
- Redirecting e-mail
- Forwarding e-mail
- Collaboration reply sending
- Automatic treatment with an acknowledgement e-mail
- Autoresponse e-mail when applicable
- Placing interactions in workbins
- Escalating overdue e-mails to supervisor workbins
- Routing to agents
- Assigning failure codes to interactions
- Promoting an interaction that failed pre-routing to the next process
- Routing interactions for QA review
- Skill-based review of agent response
- Re-processing interactions that failed QA review
- Quality control for outbound e-mails based on screening
- Re-processing interactions that failed quality control
- Sending e-mail responses to customers
- Re-processing interactions that failed sending
- Stopping an interaction with a reason code

**Note:** While queues hold interactions, and submitters and views extract interactions from queues, *routing strategy objects perform the specific processing operations.* For this reason, to understand a sample business process fully, you must also understand its routing strategies.

# The Default BP Sample

Another function of the samples is to show how interaction processing can be contained in one large business process or broken down into smaller business processes connected via queues.

The `Default BP` business process shown in Figure 88 on page 80 demonstrates “real world” workflow complexity. In order to help you understand `Default BP`, a group of samples isolate its functional areas. For each functional area of `Default BP`, such as pre-routing, the samples supply a corresponding step-numbered business process (see Figure 91 on page 85).

For example, Figure 89 shows the objects that handle the pre-routing functionality in `Default BP`.

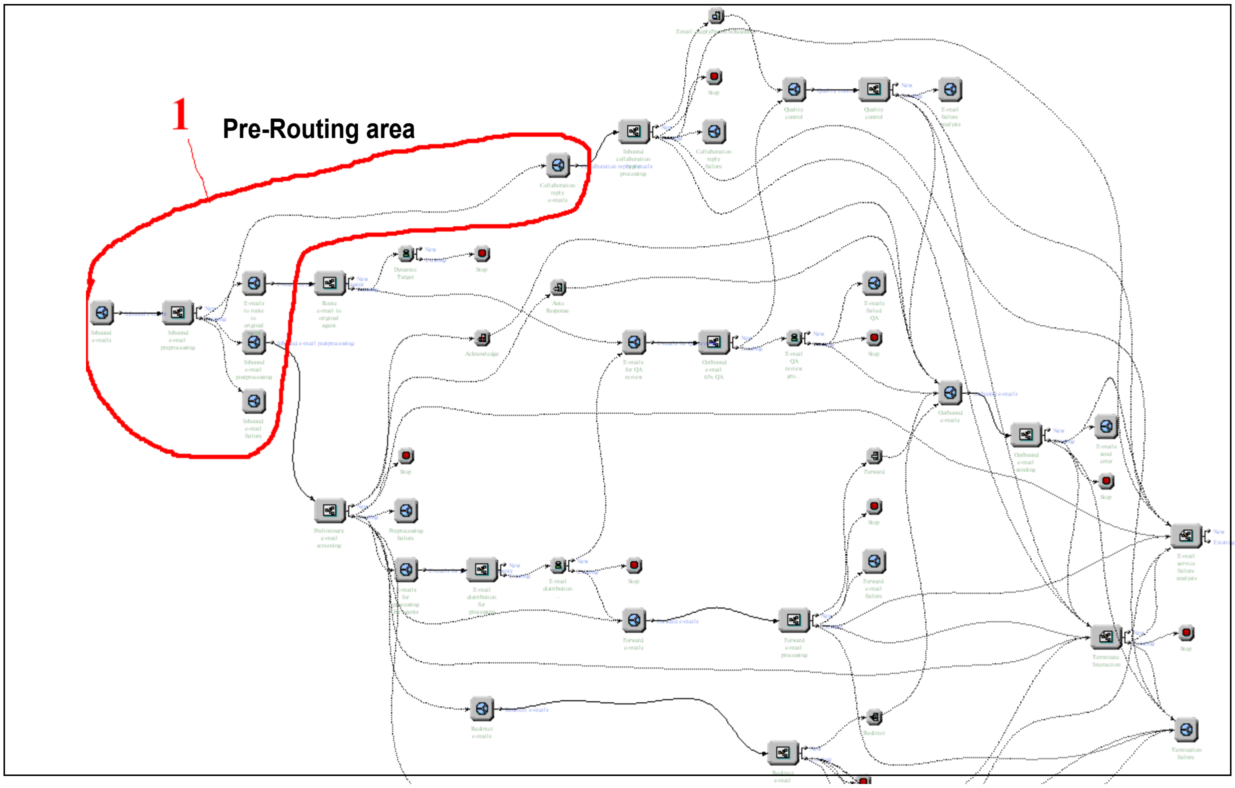

1
Pre-Routing area
Inbound e-mails
Inbound e-mail preprocessing
New
E-mails to route to original agent
Inbound e-mail postprocessing
Inbound e-mail failure
Route e-mail to original agent
Dynamic Target
Stop
Auto Response
Acknowledge
Collaboration reply e-mails
Preliminary e-mail screening
Preprocessing failure

**Figure 89: Pre-Routing Objects in Default BP**

The pre-routing objects in `Default BP` in Figure 89 correspond to the `Step 1. Pre-routing` Business Process shown in Figure 90.

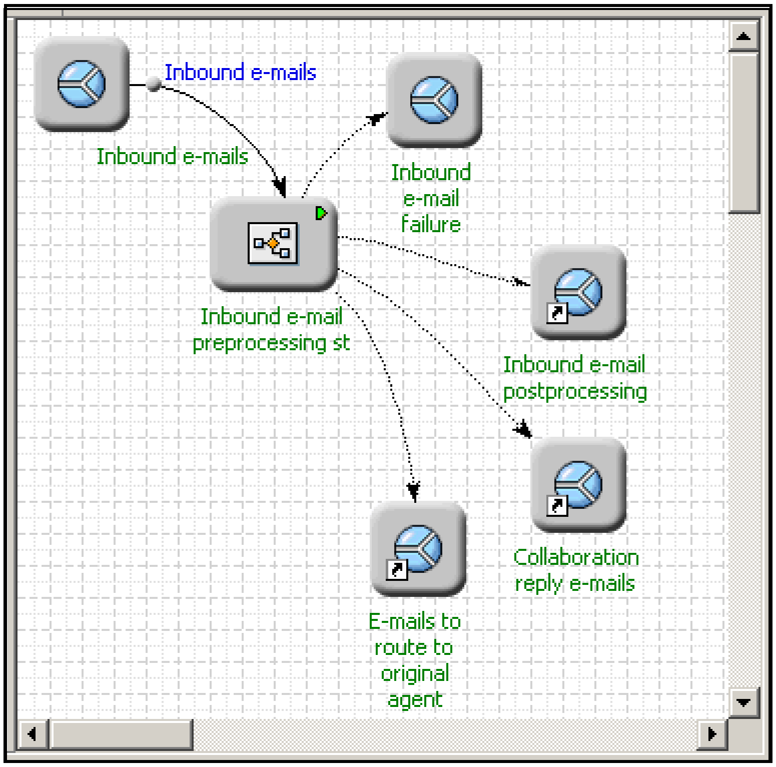

Inbound e-mails
Inbound e-mails
Inbound e-mail failure
Inbound e-mail preprocessing st
Inbound e-mail postprocessing
Collaboration reply e-mails
E-mails to route to original agent

Figure 90: Step 1. Pre-Routing Business Process

## Queues Connect Business Processes

As can be seen in Figure 90, `Step 1. Pre-routing` sends interactions out to four queues:

1. `Inbound e-mail postprocessing`, which is the input queue for `Step 2.3. New Inbound E-mails`.
2. `Collaboration reply e-mails`, which is the input queue for `Step 2.2. Inbound Collaboration Reply`.
3. `E-mails to route to original agent`, which is the input queue for `Step 2.1. NDR Handling`.
4. `Inbound e-mail failure`.

You can connect business processes in this manner to create an *interaction workflow*.

## Step-Numbered Business Processes

The business processes discussed in this chapter that have the same functionality as those within `Default_BP` are named using a step number:

- `Step 0: Common Components`
- `Step 1: Pre-Routing`

- Step 2.1: No Destination Reached (NDR) handling
- Step 2.2: Inbound Collaborator Reply Handling
- Step 2.3: New Inbound E-mails Handling
- Step 3.1: Processing By Agents
- Step 3.2: QA Review of Agent Responses
- Step 3.3: Forwarding E-mails
- Step 3.4: Redirecting E-mails
- Step 4: Outbound E-mail Sending

You can view these step-numbered business processes in the Interaction Design window (see Figure 91).

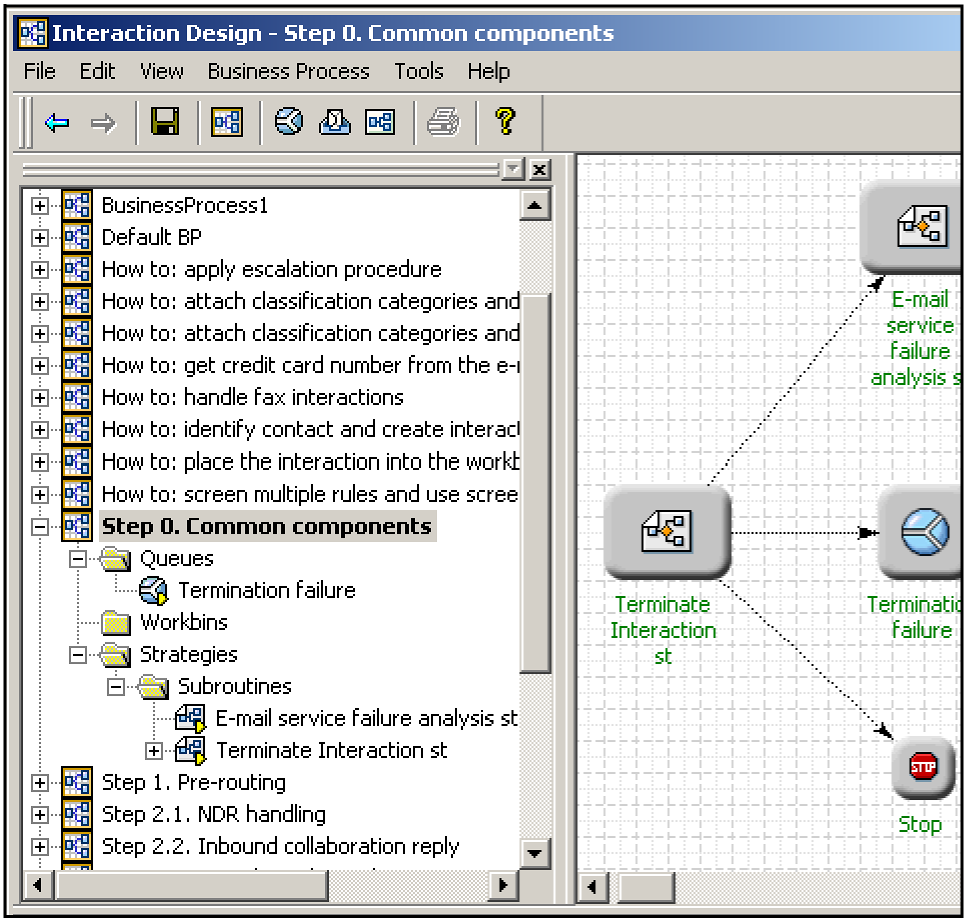

Interaction Design - Step 0. Common components
File Edit View Business Process Tools Help
BusinessProcess1
Default BP
How to: apply escalation procedure
How to: attach classification categories and
How to: attach classification categories and
How to: get credit card number from the e-
How to: handle fax interactions
How to: identify contact and create interact
How to: place the interaction into the workb
How to: screen multiple rules and use scree
Step 0. Common components
Queues
Termination failure
Workbins
Strategies
Subroutines
E-mail service failure analysis st
Terminate Interaction st
Step 1. Pre-routing
Step 2.1. NDR handling
Step 2.2. Inbound collaboration reply
E-mail service failure analysis s
Terminate Interaction st
Terminatio failure
STOP
Stop

Figure 91: Step-Numbered Business Processes

**Note:** In order to fully understand the strategy samples contained in this chapter, you must view each sample in the context of a business process. This can be the current business process, a business process that came before the strategy, or a business process that comes after the strategy. If not viewed in this context, interaction processing may seem fragmented and/or incomplete.

### What's Next

This chapter continues by presenting the routing strategies that are used to demonstrate the various functional areas that are listed on page 82.

# Pre-Routing Based on Interaction Subtype

The business process `Step 1: Pre-Routing` contains the strategy `Inbound e-mail preprocessing`, which is described in this section (see Figure 92).

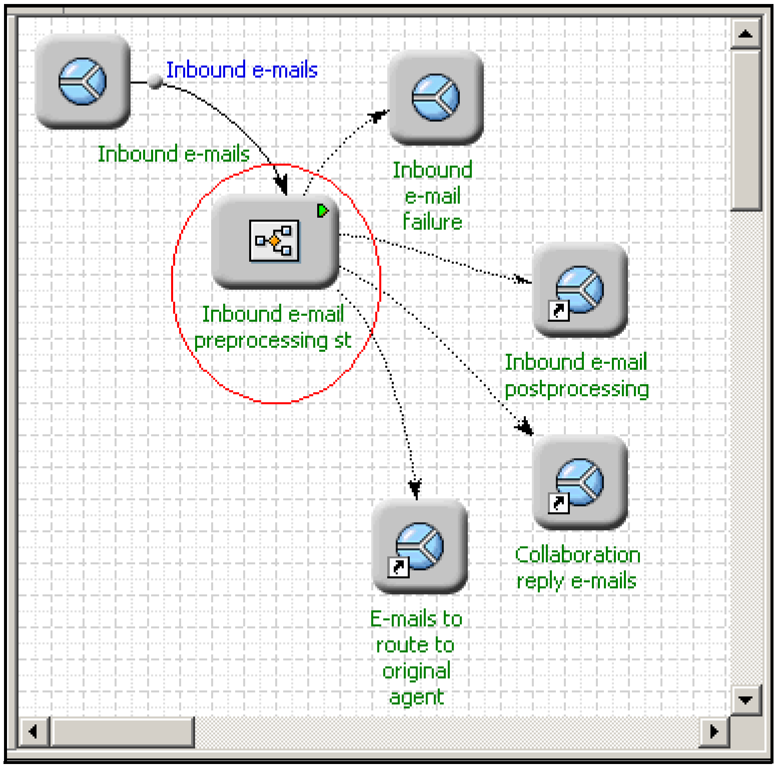

Inbound e-mails
Inbound e-mails
Inbound e-mail failure
Inbound e-mail preprocessing st
Inbound e-mail postprocessing
Collaboration reply e-mails
E-mails to route to original agent

**Figure 92: Step 1: Pre-Routing Business Process**

## Inbound E-mail Preprocessing Strategy

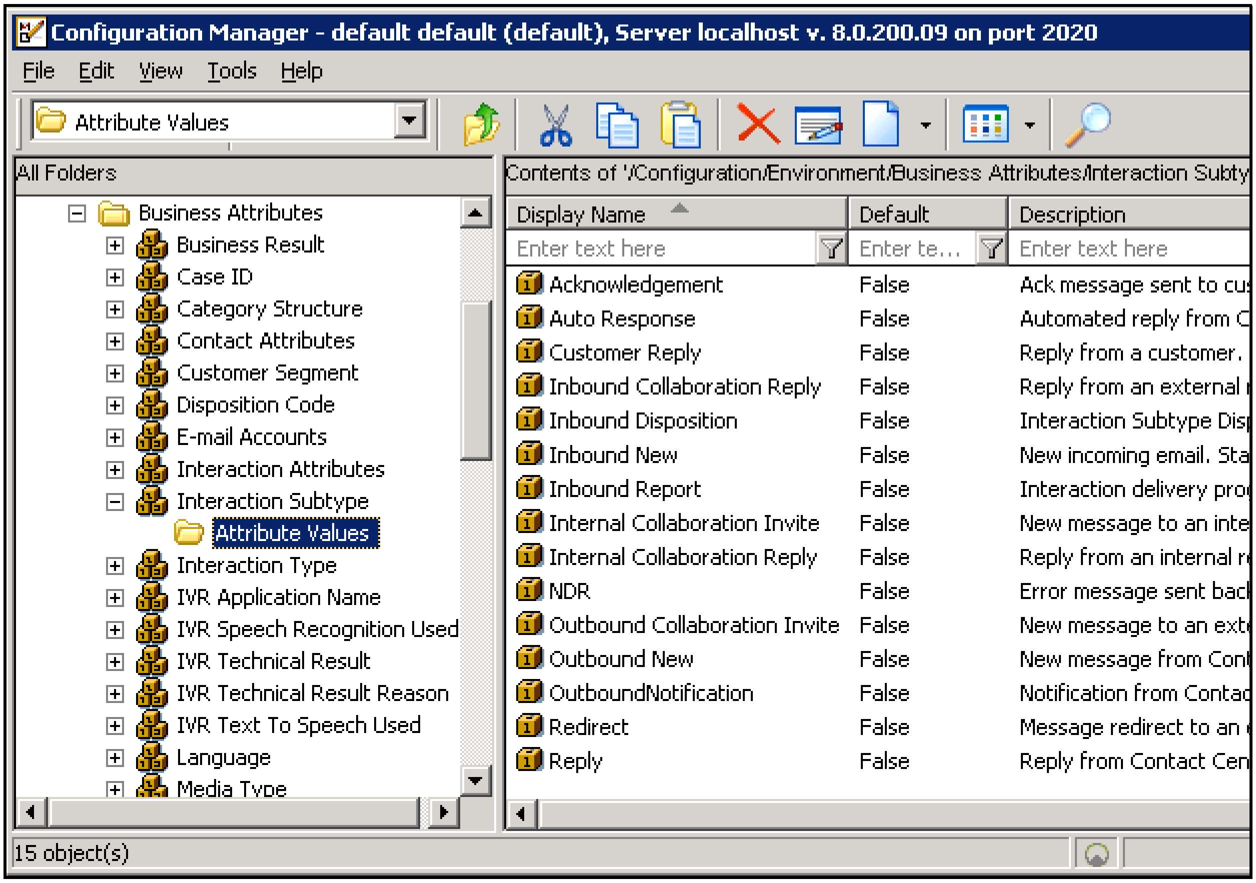
This strategy processes inbound e-mails from an initial inbound queue. A view object (named `Inbound e-mails` in Figure 92) extracts e-mails from the `Inbound` `e-mails` queue in Figure 92 and a submitter object submits them to the `Inbound e-mail preprocessing` strategy.

Incoming e-mails may be new e-mails from customers or e-mails that have already gone through Genesys e-mail processing.

- If the e-mail is new, E-mail Server assigns an `Inbound New Interaction Subtype`.
- If the e-mail has already been processed by Genesys, the previous strategy will have assign an `Interaction Subtype` (see below) or an error code that uses the `Gem_Failure` key (see page 91).

## Interaction Sub-types

Available `Interaction Subtypes` that can be assigned to interactions can be seen in Configuration Manager under `Business Attributes` (see Figure 93).



Configuration Manager - default default (default), Server localhost v. 8.0.200.09 on port 2020

Attribute Values

All Folders

- Business Attributes
  - Business Result
  - Case ID
  - Category Structure
  - Contact Attributes
  - Customer Segment
  - Disposition Code
  - E-mail Accounts
  - Interaction Attributes
  - Interaction Subtype
    - Attribute Values
  - Interaction Type
  - IVR Application Name
  - IVR Speech Recognition Used
  - IVR Technical Result
  - IVR Technical Result Reason
  - IVR Text To Speech Used
  - Language
  - Media Type

Contents of '/Configuration/Environment/Business Attributes/Interaction Subty

| Display Name | Default | Description |
|---|---|---|
| Enter text here | Enter te... | Enter text here |
| Acknowledgement | False | Ack message sent to cu |
| Auto Response | False | Automated reply from C |
| Customer Reply | False | Reply from a customer. |
| Inbound Collaboration Reply | False | Reply from an external r |
| Inbound Disposition | False | Interaction Subtype Disp |
| Inbound New | False | New incoming email. Sta |
| Inbound Report | False | Interaction delivery pro |
| Internal Collaboration Invite | False | New message to an inte |
| Internal Collaboration Reply | False | Reply from an internal r |
| NDR | False | Error message sent bacl |
| Outbound Collaboration Invite | False | New message to an exte |
| Outbound New | False | New message from Cont |
| OutboundNotification | False | Notification from Contac |
| Redirect | False | Message redirect to an e |
| Reply | False | Reply from Contact Cen |

15 object(s)

**Figure 93: Interaction Subtypes in Configuration Manager**

**Note:** The right pane lists the Display Name; the object name as given in the properties dialog box that opens when you double-click the display name may be different. For example, `NDR` appears in the Display Name column but the object name is `InboundNDR`.

### Example EventRouteRequest Message

Interaction Server uses T-Library protocol to communicate with URS. All of the interaction properties, including attached data, are presented to URS in a `UserData` property of the T-Event structure. The following is partial view of an example `EventRouteRequest` message that provides interaction data to URS:

```
17:05:21.462 Trc 24102 Sending to router: universal_router_showtime:
'EventRouteRequest' message:
        AttributeExtensions [bstr] = TKVList:
           'STRATEGY_ID' [str] = "strategy_first"
        AttributeThisQueue [str] = "asl_strategy_first"
        AttributeThisDN [str] = "asl_strategy_first"
        AttributeCallType [int] = 0
        AttributeCallID [int] = 3829
        AttributeMediaType [int] = -1
        AttributeConnID [long] =
        AttributeCustomerID [str] = "Genesys"
        AttributeUserData [bstr] = TKVList:
           'AccountBalance' [int] = 0
           'CustomNumber2' [int] = 0
           'CustomNumber3' [int] = 0
           'Priority' [int] = 0
           'ServiceObjective' [int] = 0
           'InteractionId' [str] = "tst00004e20"
           'TenantId' [int] = 101
           'MediaType' [str] = "email"
           'InteractionType' [str] = "Inbound"
           'InteractionSubtype' [str] = "InboundNew"
           'InteractionState' [int] = 1
           'IsOnline' [int] = 0
           'IsLocked' [int] = 0
           'Queue' [str] = "queue_first"
           'SubmittedBy' [str] = "Workflow Engine Tester"
           'ReceivedAt' [str] = "2004-04-14T22:10:35Z"
           'SubmittedAt' [str] = "2004-04-14T22:10:34Z"
           'DeliveredAt' [str] = "2004-04-26T23:30:02Z"
           'PlacedInQueueAt' [str] = "2004-04-23T18:04:10Z"
```

**Note:** Although not shown in the partial view above, the UserData attribute can also contain customer-defined attributes, such as `Gem_Failure` (see Figure 99 on page 91) and `GD_OriginalAgentEmployeeId` (see Figure 105 on page 95).

### Interaction Subtype Assignment

The flow of an e-mail through the `Inbound e-mail preprocessing` strategy is determined by `Interaction Subtype` assignment.

- If it is assigned the `NDRInteraction` subtype, the e-mail is routed to a queue for e-mails where no destination can be reached.
- If it is assigned the `InboundCollaborationReply` subtype, the e-mail is routed to a queue handling replies from one agent to another (collaboration replies).
- If no specific `Interaction Subtype` is assigned, the strategy determines whether the e-mail has an inbound processing failed status. If yes, the e-mail is routed to a queue for those types of e-mails. If not, the e-mail is routed to a queue for further processing.

Figure 94 shows the sample `Inbound email preprocessing` strategy in IRD.

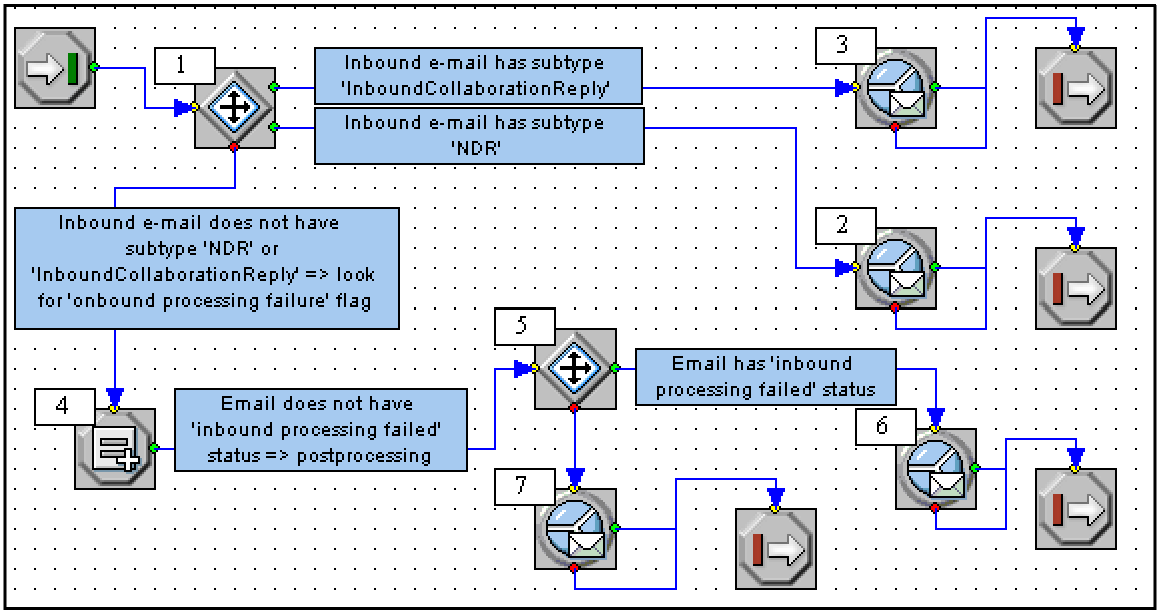

1
Inbound e-mail has subtype 'InboundCollaborationReply'
Inbound e-mail has subtype 'NDR'
3
2
Inbound e-mail does not have subtype 'NDR' or 'InboundCollaborationReply' => look for 'onbound processing failure' flag
5
4
Email does not have 'inbound processing failed' status => postprocessing
Email has 'inbound processing failed' status
6
7

**Figure 94: Inbound E-mail Preprocessing Strategy**

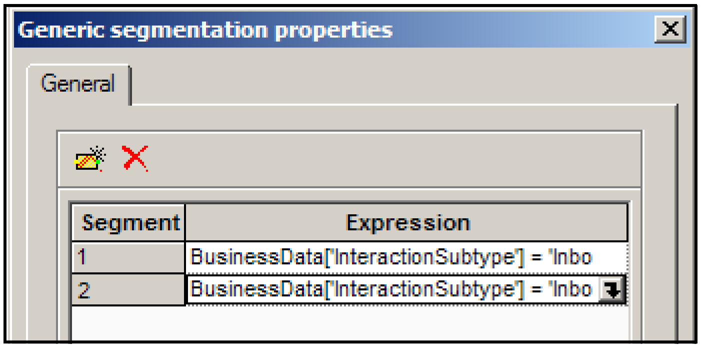
## Summary of Flow

**Note:** The IRD objects that are described in this section are keyed to the numbers in Figure 94.

The general flow for the `inbound-email-preprocessing` strategy is as follows:

1. A Generic Segmentation object uses the BusinessData function (described in the *Universal Routing 8.1 Reference Manual*) to retrieve one of the following `Interaction Subtypes` from the e-mail:
   - `InboundNDR` (No Destination Reached)
   - `InboundCollaborationReply`

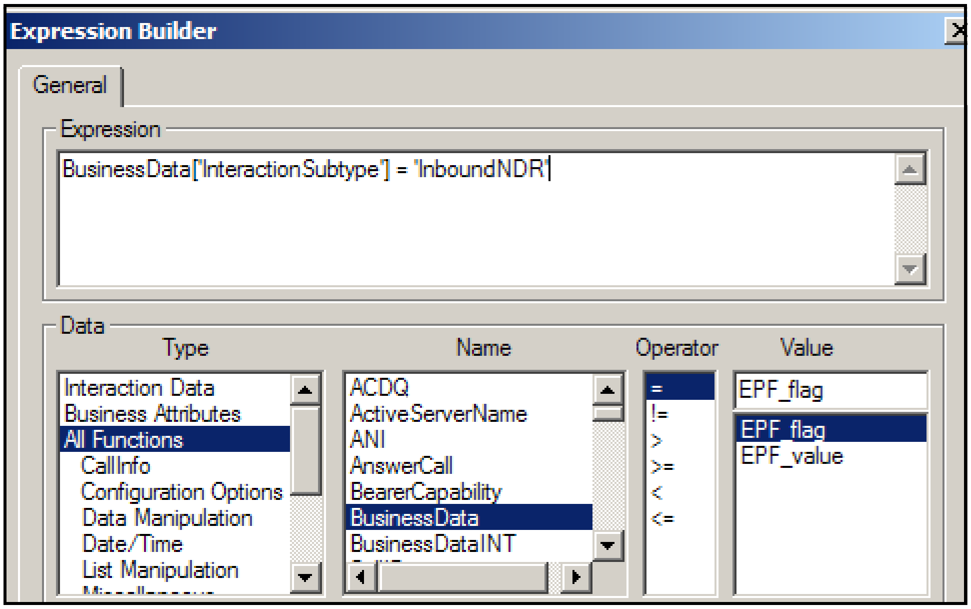
Figure 95 shows the Generic Segmentation Properties dialog box.



| Segment | Expression |
|---|---|
| 1 | BusinessData['InteractionSubtype'] = 'Inbo |
| 2 | BusinessData['InteractionSubtype'] = 'Inbo |

**Figure 95: Generic Segmentation Properties Dialog Box**

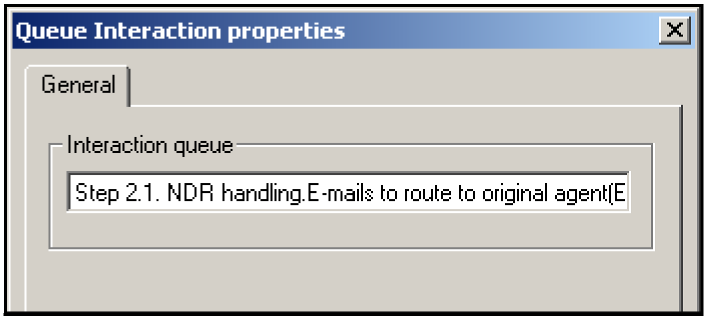
Figure 96 shows how the second expression in the dialog box was created in Expression Builder, which opens when you click the down arrow shown in Figure 95.



Expression: BusinessData['InteractionSubtype'] = 'InboundNDR'

**Figure 96: Expression Builder Dialog Box, InboundNDR**

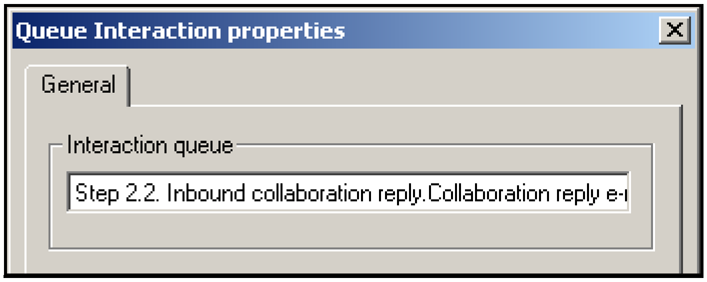
2. If an e-mail contains the `InboundNDR Interaction Subtype`, it goes through the green port (see Figure 3 on page 18) of the Generic Segmentation object to an interaction queue specified in a Queue Interaction object (see Figure 97).



Interaction queue: Step 2.1. NDR handling.E-mails to route to original agent(E

Figure 97: Queue Interaction Properties—NDR Handling

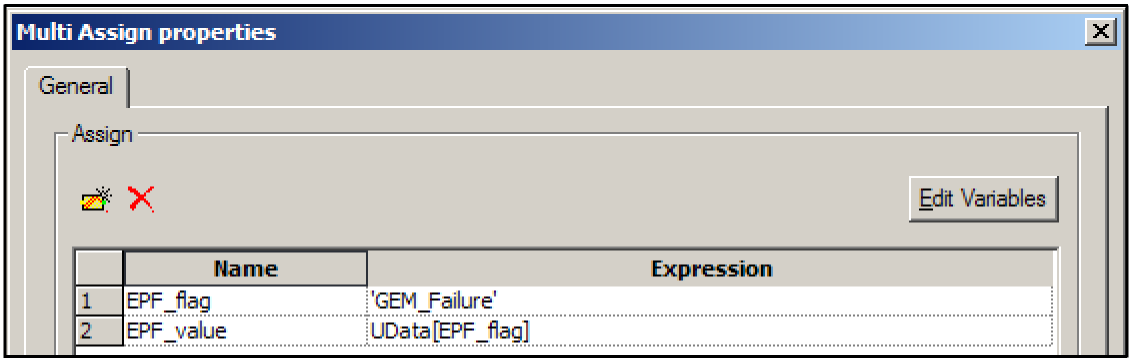
3. If an e-mail contains the `InboundCollaborationReply Interaction Subtype`, it also goes through the green port (see Figure 3 on page 18) of the Generic Segmentation object to an interaction queue specified in another Queue Interaction object (see Figure 98).



Interaction queue: Step 2.2. Inbound collaboration reply.Collaboration reply e-

Figure 98: Queue Interaction Properties—Inbound Collaboration Reply

4. If the inbound e-mail does not have either of these `Interaction Subtypes` attached, it goes through the red port of the Generic Segmentation object to a MultiAssign object, which assigns values to two variables: `EPF_flag` and `EPF_value` (see Figure 99).



| | Name | Expression |
|---|---|---|
| 1 | EPF_flag | 'GEM_Failure' |
| 2 | EPF_value | UData[EPF_flag] |

Figure 99: MultiAssign Properties Dialog Box

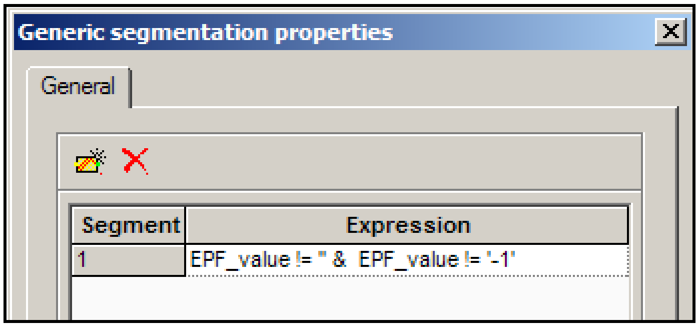
## Gem_Failure

In Figure 99, `Gem_Failure` is the key for a customer-defined field in the interaction (see “Example EventRouteRequest Message” on page 88). It contains an error code used for failed e-mail processing.

- For a new incoming e-mail (that E-mail Server just retrieved from the corporate e-mail server), this attached data contains an error code representing the reason the inbound e-mail was detected as failed. Although not shown in this example, the strategy also assumes that a key called `GEM_FailureMsg` contains a more explicit message as attached data.
- For an e-mail that previously went though processing by an IRD strategy, `Gem_Failure` contains the error code returned by the last IRD service called. Again, the `GEM_FailureMsg` attached data contains a more explicit message.

In the MultiAssign Properties dialog box shown in Figure 99, the presence and the value of this attached data are tested to check if the current incoming e-mail is a failed one. A failed e-mail requires different processing than a normal e-mail.

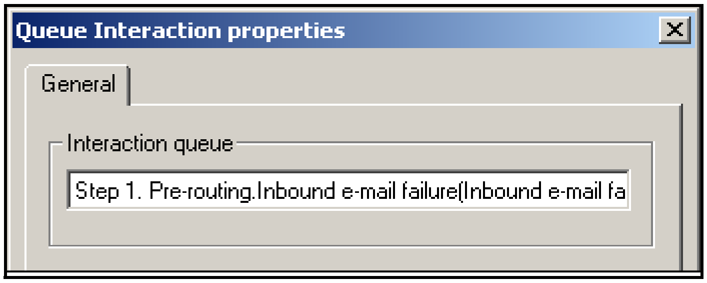
5. Once the variables are assigned, the interaction goes through the green port (see Figure 3 on page 18) of the MultiAssign object to another Generic Segmentation object, which directs the e-mail based on whether an expression is true or false (see Figure 100).



| Segment | Expression |
|---|---|
| 1 | EPF_value != '' &  EPF_value != '-1' |

Figure 100: Generic Segmentation Properties Dialog Box

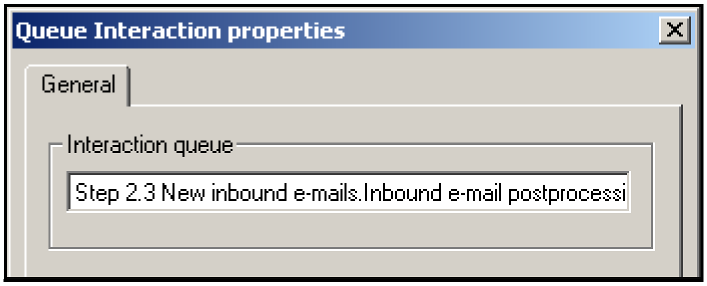
6. If the expression is true (`-1`), then the interaction has an E-mail Processing Failed (EPF) status. In this case, the e-mail goes through the green port of the Generic Segmentation object to a Queue Interaction object that specifies a queue used for analyzing e-mail failures (see Figure 101).



Interaction queue: Step 1. Pre-routing.Inbound e-mail failure(Inbound e-mail fa

Figure 101: Queue Interaction Properties—Inbound E-Mail Failure

7. If the expression in Figure 100 is false, then the interaction does not have an E-mail Processing Failed status and needs further processing. In this case, the e-mail goes through the red port of the Generic Segmentation object to a Queue Interaction object that specifies a queue for inbound e-mail post processing (see Figure 102).



Interaction queue: Step 2.3 New inbound e-mails.Inbound e-mail postprocessi

**Figure 102: Queue Interaction Properties—New Inbound E-Mails**

**Note:** The `Step 2.3 New inbound e-mails.inbound e-mail postprocessing` queue is connected to another business process, `Step 2.3, New Inbound E-mails Handling` (see Figure 113 on page 100).

# Routing E-mails To the Original Agent

The business process `Step 2.1: NDR Handling` contains the strategy `Route e-mail to original agent`, which is described in this section (see Figure 103).

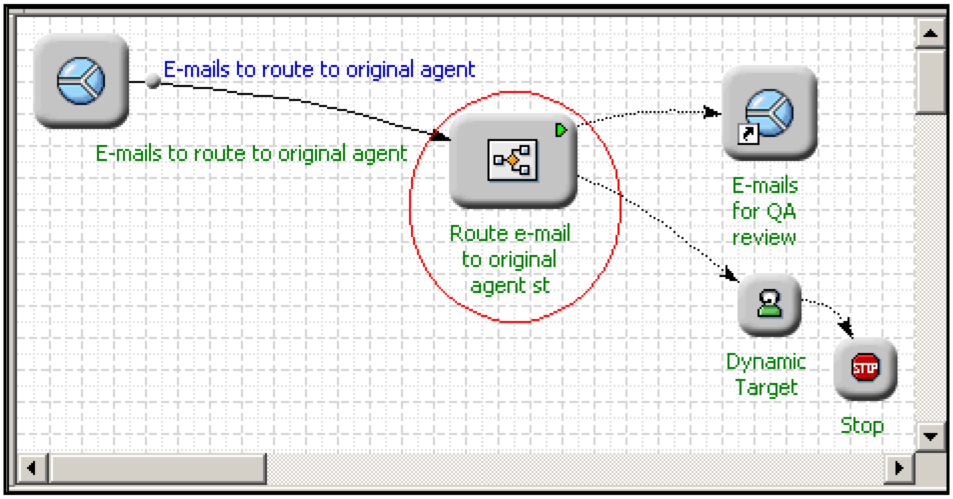

E-mails to route to original agent
E-mails to route to original agent
Route e-mail to original agent st
E-mails for QA review
Dynamic Target
Stop

**Figure 103: Step 2.1 NDR Handling Business Process**

The `E-mails to route to original agent` queue in Figure 103 gets interactions from the `Inbound e-mail preprocessing` strategy shown in Figure 92 on page 86. A view attached to the `E-mails to route to original agent` queue extracts e-mails. A submitter submits e-mails to the `E-mails route to original agent` strategy shown in Figure 103.

**Note:** For more information on Interaction Design queues, views, and submitters, see the Interaction Workflow Designer section of the *Universal Routing 8.1 Interaction Routing Designer Help* and the *Universal Routing 8.1 Business Process User's Guide*.

# Route E-mail to Original Agent Strategy

The `Route email to original agent` strategy demonstrates how to route a customer reply e-mail back to the agent who handled the original e-mail. It assumes the employee ID of the original agent or QA reviewer is contained in the interaction attached data (see page 88). Figure 104 shows the `Route-email-to-original-agent` strategy.

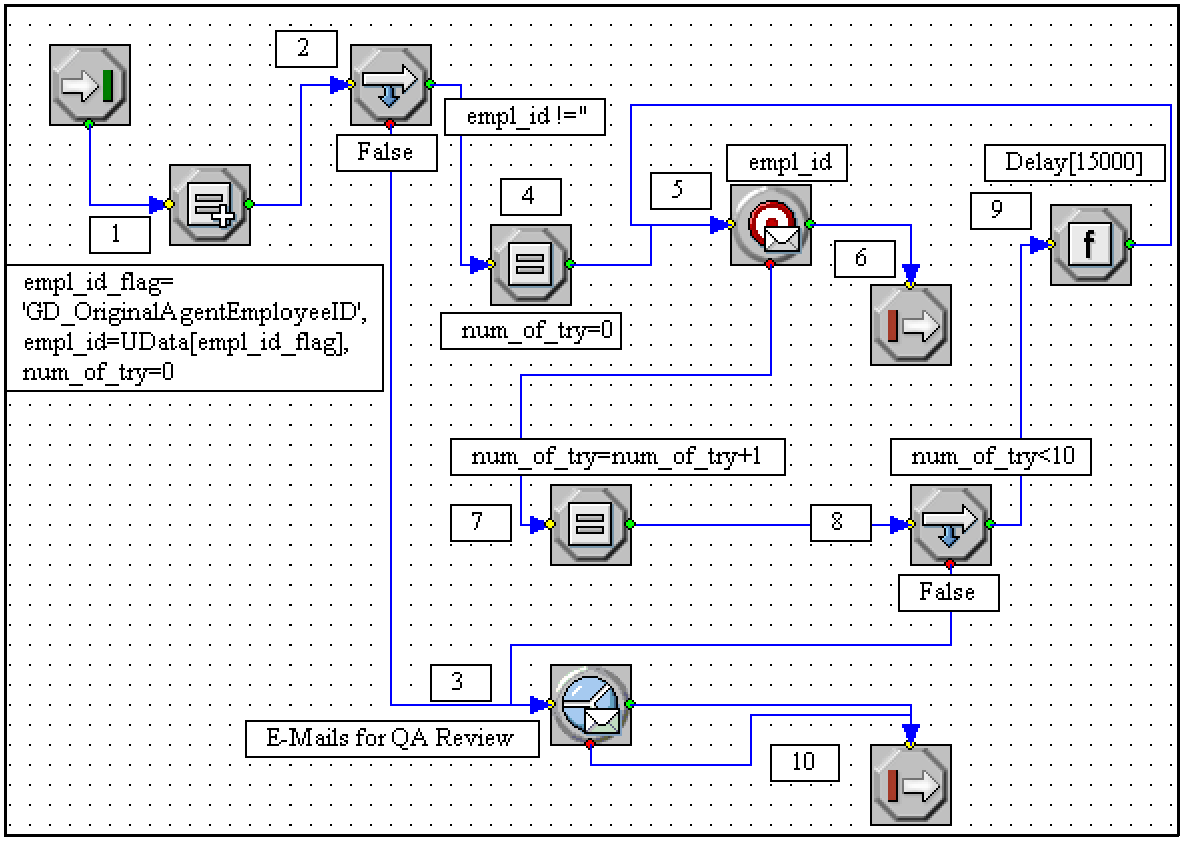
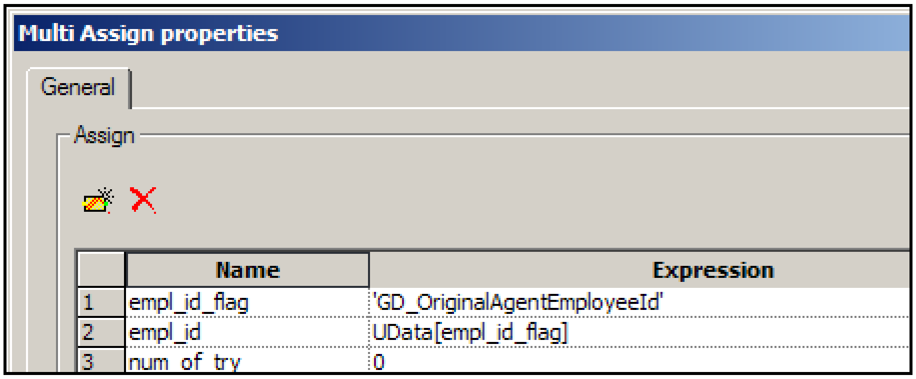


**Figure 104: Route E-mail to Original Agent Strategy (IRD View)**

## Summary of Flow

**Note:** The IRD objects that are described this section are keyed to the numbers in Figure 104

1. A MultiAssign object assigns e-mail interaction attributes to three predefined variables (see Figure 105).



| | Name | Expression |
|---|---|---|
| 1 | empl_id_flag | 'GD_OriginalAgentEmployeeId' |
| 2 | empl_id | UData[empl_id_flag] |
| 3 | num_of_try | 0 |

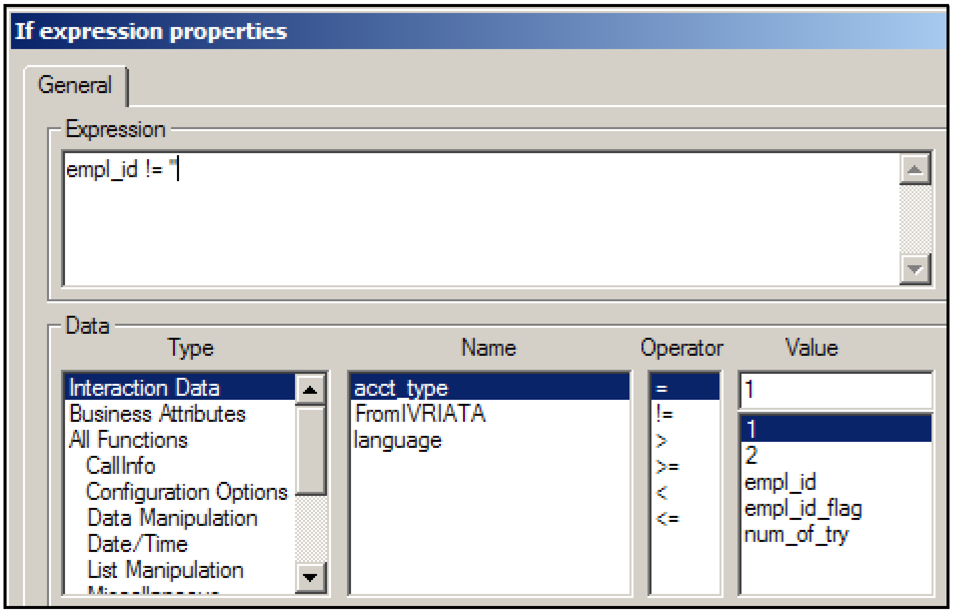
**Figure 105: Multi Assign Properties Dialog Box**

### Interaction Attached Data

In Figure 105 under Expression:

- `GD_OriginalAgentEmployeeID` is an attached data key that may contain the identifier of the QA person who reviewed the original agent's e-mail response (see "Strategy Assumptions" on page 128).
- The UData function attempts to get a value from the attached data key `empl_id_flag` and write it to a variable. If this key contains a value, it indicates there is an original agent to whom the interaction can be routed.
- The variable `num_of_tries` is initialized to `0`. It functions as a counter used to control the number of tries the strategy makes to route to the original agent.

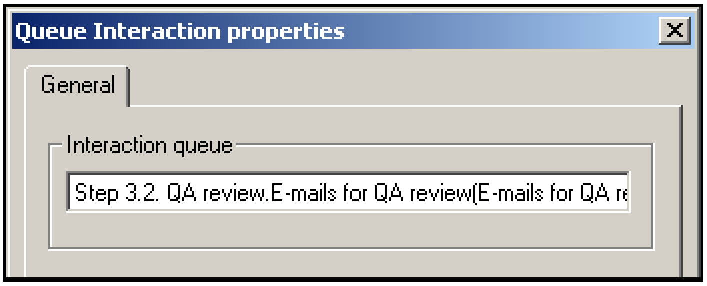
**2.** After setting the variables, an If object initializes the `empl_id` variable so it does not contain a value (see Figure 106).



Expression: empl_id != ''

**Figure 106: If Properties Dialog Box, Employee ID Initialized**

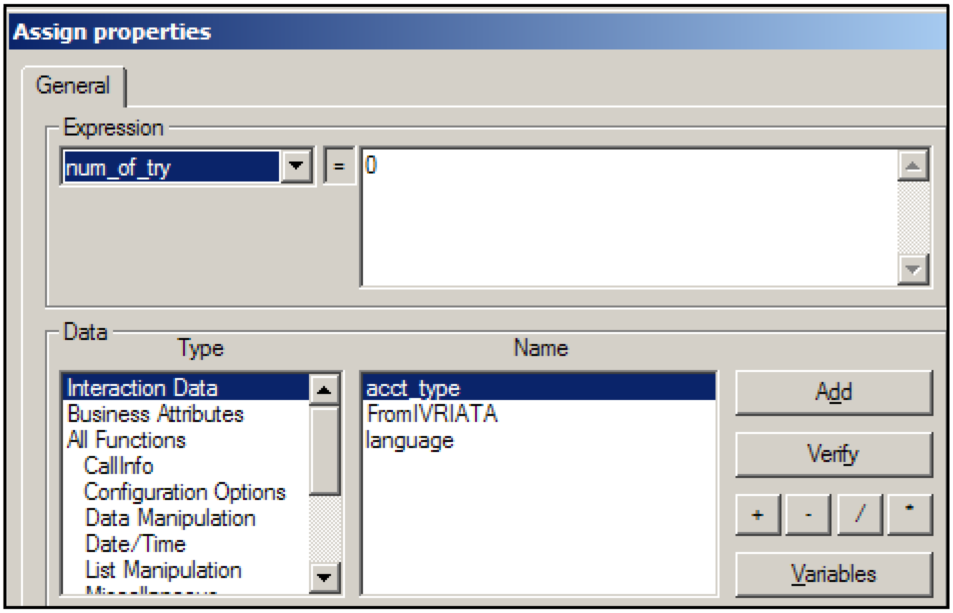
3. If the expression in Figure 106 on page 95 is true (if there is no original agent), the e-mail goes out the red port of the If object to a Queue Interaction object specifying a queue for QA review (see Figure 107).



Interaction queue: Step 3.2. QA review.E-mails for QA review(E-mails for QA re

Figure 107: Queue Interaction Properties—E-mails for QA Review

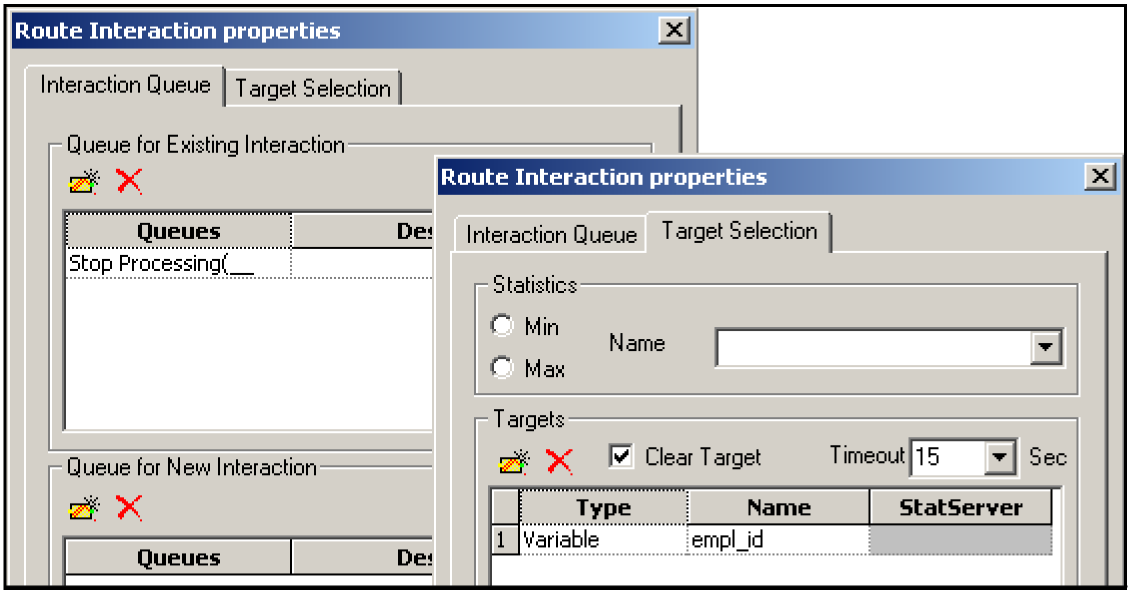
4. If the expression in Figure 106 on page 95 is false (if there is an original agent), the e-mail goes out the green port of the If object to an Assign object, which sets a counter (variable `num_of_try`) to zero (see Figure 108).



Expression: num_of_try = 0

Figure 108: Assign Properties Dialog Box, Number of Tries Counter

5. A Route Interaction object attempts to route the interaction to the original agent contained in the `empl_id` variable (see Figure 109).


Route Interaction properties
Interaction Queue
Target Selection
Queue for Existing Interaction
Queues
Des
Queue for New Interaction
Route Interaction properties
Statistics
Min
Max
Name
Targets
Timeout
Sec

Queues: Stop Processing(__

☑ Clear Target

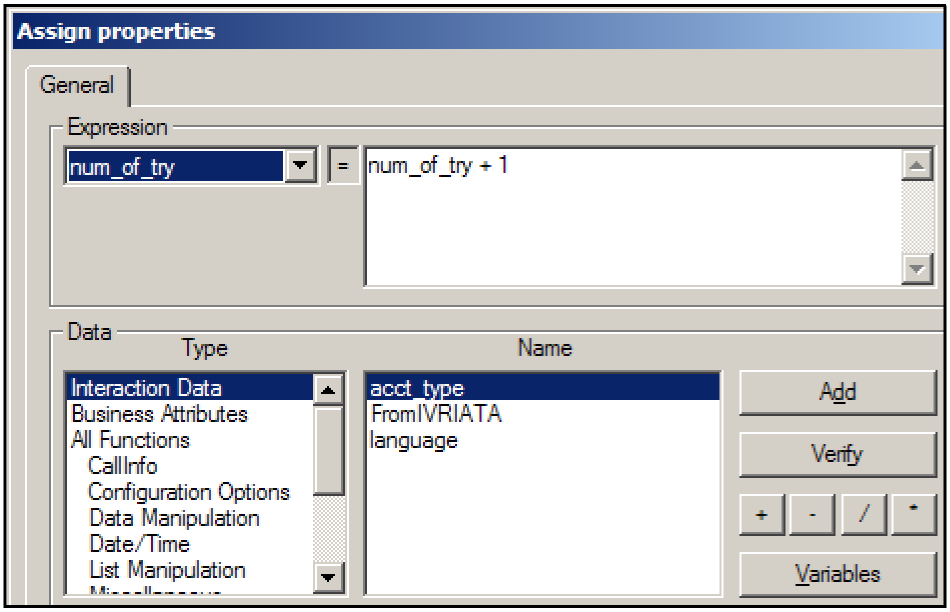
Timeout: 15 Sec

| | Type | Name | StatServer |
|---|---|---|---|
| 1 | Variable | empl_id | |

Figure 109: Route Interaction Properties—Routing Based On Variable

6. If the `empl_id` variable contains the name of the original agent, the e-mail goes out the green port of the Route Interaction object to an Exit object.

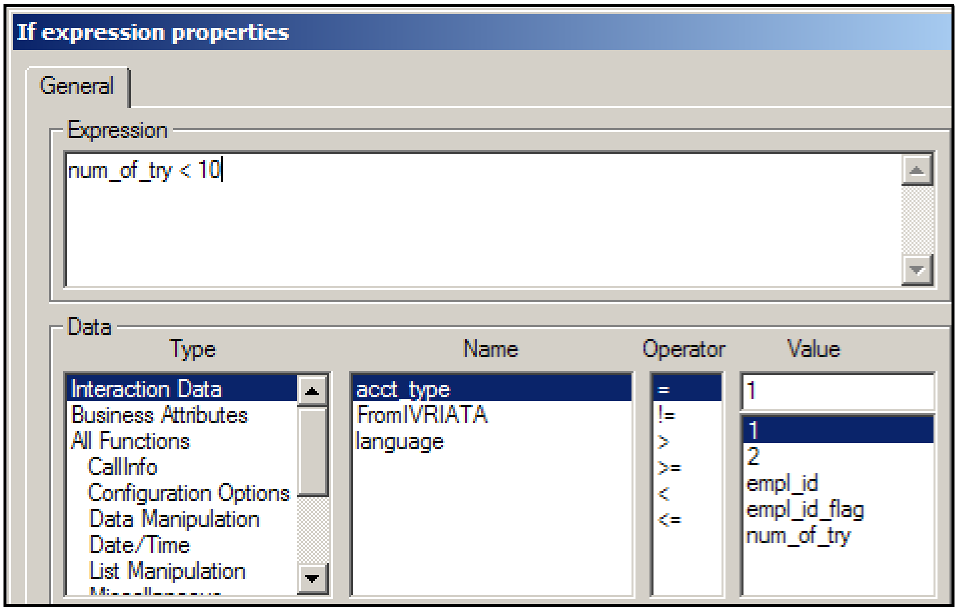
7. If the `empl_id` variable does not contain the name of the original agent, the e-mail goes out the red port of the Route Interaction object to an Assign object, which increments the `num_of_try` counter (see Figure 110).



Expression: num_of_try = num_of_try + 1

Figure 110: Assign Properties Dialog Box, Number of Tries

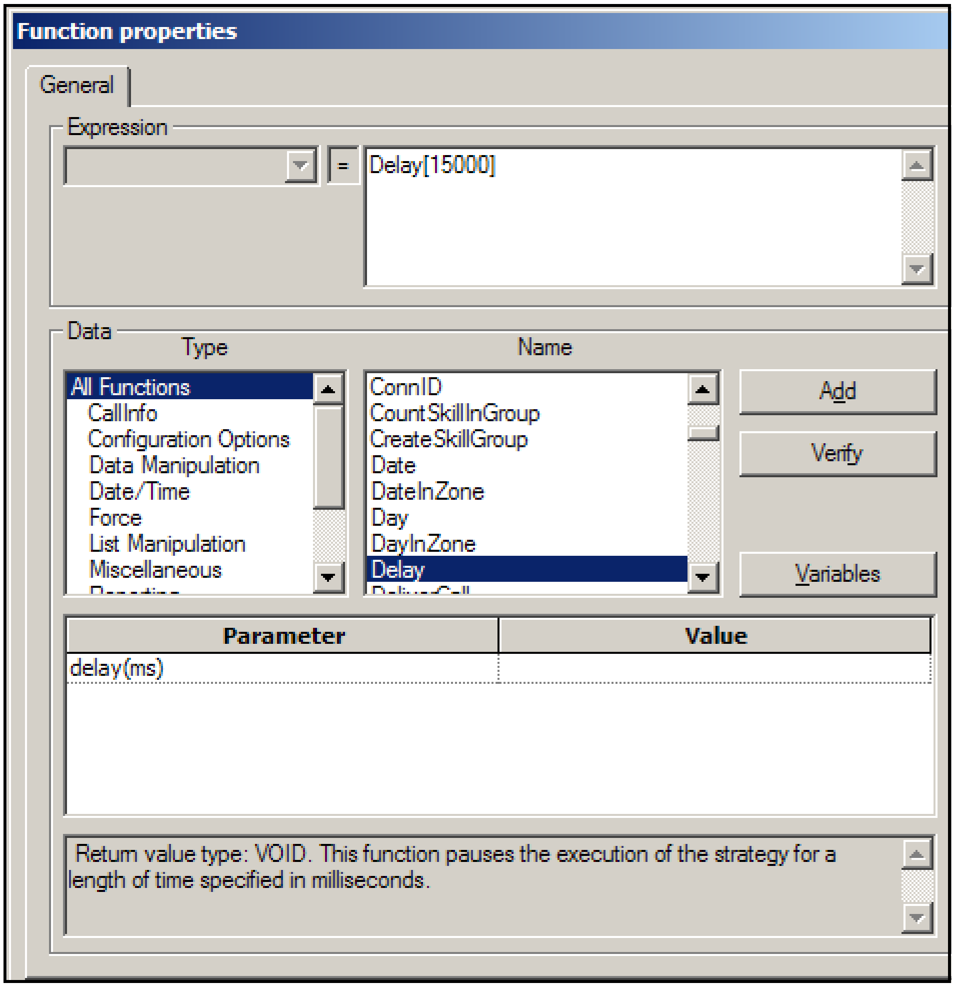
8. The e-mail then goes out the green port of the Assign object to an If object, which is used to decide whether there have been more than 10 processing cycles to determine the original agent (see Figure 111).



Expression: num_of_try < 10

**Figure 111: If Expression Properties Dialog Box, Number of Tries**

9. If the number of tries to route to the original agent is less than `10`, the interaction goes out the green port of the If object to a Function object used to delay strategy execution (see Figure 112).



Delay[15000]

| Parameter | Value |
|---|---|
| delay(ms) | |

Return value type: VOID. This function pauses the execution of the strategy for a length of time specified in milliseconds.

**Figure 112: Function Properties Dialog Box, Delay Function**

Once the delay is established, the e-mail goes out the green port of the Function object back to the Route Interaction object in Figure 109 on page 97.

10. If the expression in Figure 111 on page 98 is false (number of tries is 10 or greater), the e-mail goes out the green port of the If object to the Queue Interaction object (see Figure 107 on page 96).

# Screening of Inbound E-mails

The business process `Step 2.3: New Inbound E-mails Handling` (see page 85) contains the strategy `Preliminary e-mail screening`, which is described in this section (shown in Figure 113).

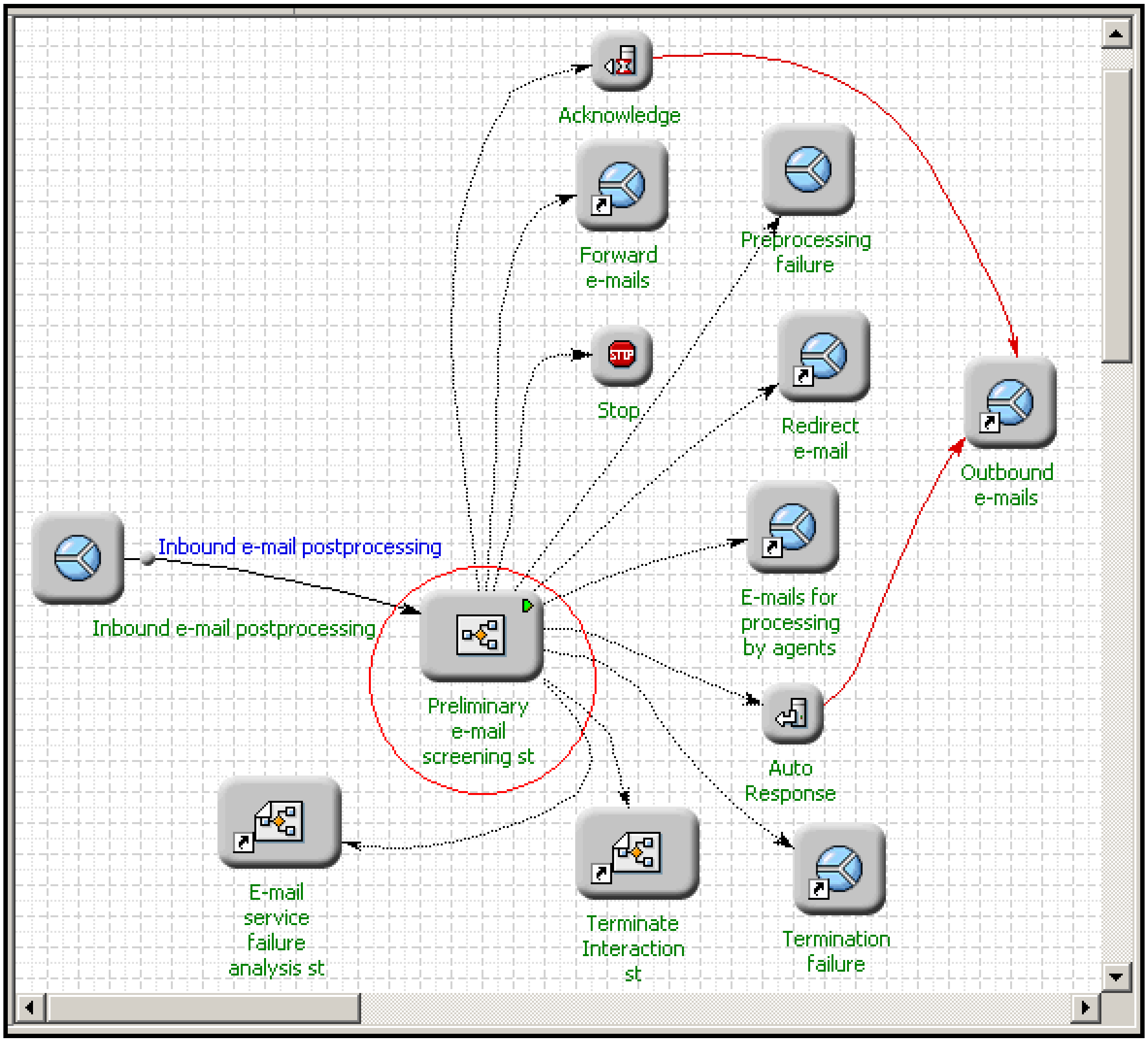

Acknowledge
Forward e-mails
Preprocessing failure
Stop
Redirect e-mail
Outbound e-mails
Inbound e-mail postprocessing
Inbound e-mail postprocessing
E-mails for processing by agents
Preliminary e-mail screening st
Auto Response
E-mail service failure analysis st
Terminate Interaction st
Termination failure

**Figure 113: Step 2.3: New Inbound E-mails Handling**

## Preliminary E-mail Screening Strategy

This strategy uses three different IRD Screen objects to analyze incoming e-mail content using three different screening rules:

1. `Auto Response Available`: To determine if a standard response can be used to respond to the customer.
2. `Warranty Problem`: To determine if the e-mail content is about product warranties.
3. `Tech Support`: To determine if the e-mail content indicates the e-mail should be redirected to Technical Support.

## Auto Response Available Screening Rule

An initial screening is performed using the `Auto Response Available` rule. The results are written to a variable, which is subsequently used in an If expression:

- If the expression is true (screening rule match is found), an Autoresponse object selects a standard response.
  - If a standard response is generated, a Stop object notifies Interaction Server that processing for the strategy has stopped.
  - If a standard response e-mail cannot be generated, a subroutine is called that performs failure analysis and generates an error code.
- If the expression is false, a second screening occurs.

If the initial screening could not be performed, the interaction goes out the red port to a Call Subroutine object for failure analysis.

## Warranty Problem Screening Rule

If there are no screening results using the `Auto Response Available` rule, a second screening is performed using the `Warranty Problem` rule. Screening results are written to a variable, which is subsequently used in an If expression.

- If the expression is true (screening rule match is found), the e-mail goes to a queue for e-mails to be forwarded.
- If the expression is not true (screening rule match is not found), a third screening is performed with the `Tech Support` screening rule.

If the second screening cannot be performed, the interaction goes out the red port to a Call Subroutine object for failure analysis.

## Tech Support Screening Rule

If there are no screening results using the `Warranty Problem` screening rule, a third screening is performed with the `Tech Support` screening rule. Screening results are written to a variable, which is subsequently used in an If expression.

- If the expression is true (screening rule match is found), the e-mail goes to a queue for e-mails to be redirected.
- If the expression is not true (screening rule match is not found) the e-mail must be processed by an agent. The e-mail goes out the red port to an Acknowledgement object. After an acknowledgement e-mail is generated, the e-mail goes to a queue for processing by agents.

If screening cannot be performed with the `Tech Support` screening rule, the interaction goes out the red port to a Call Subroutine object for failure analysis.

Figure 114 shows the start of the `Preliminary e-mail screening` routing strategy that performs the initial screening using the `Auto Response Available` screening rule.

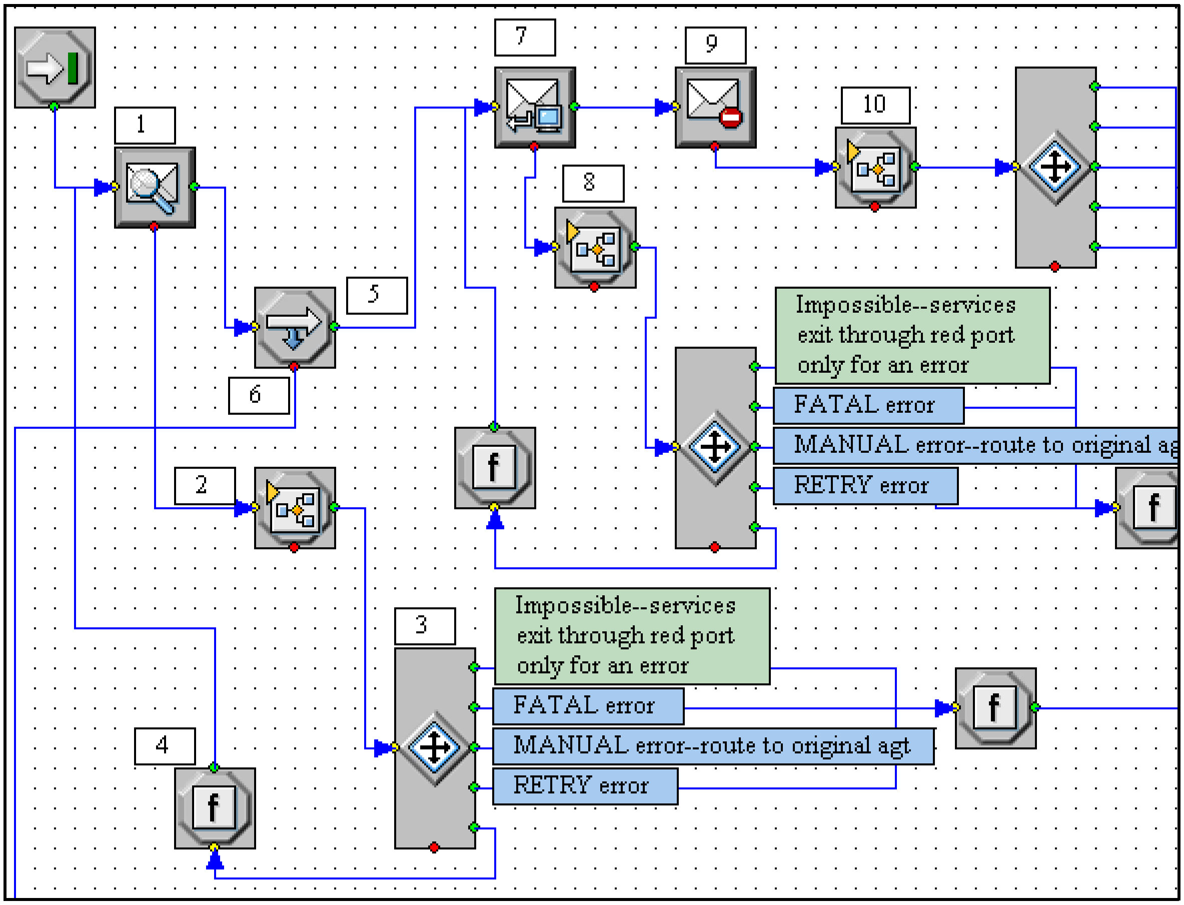

7
9
10
1
8
5
6
2
Impossible--services exit through red port only for an error
FATAL error
MANUAL error--route to original ag
RETRY error
f
3
Impossible--services exit through red port only for an error
FATAL error
MANUAL error--route to original agt
RETRY error
4
f

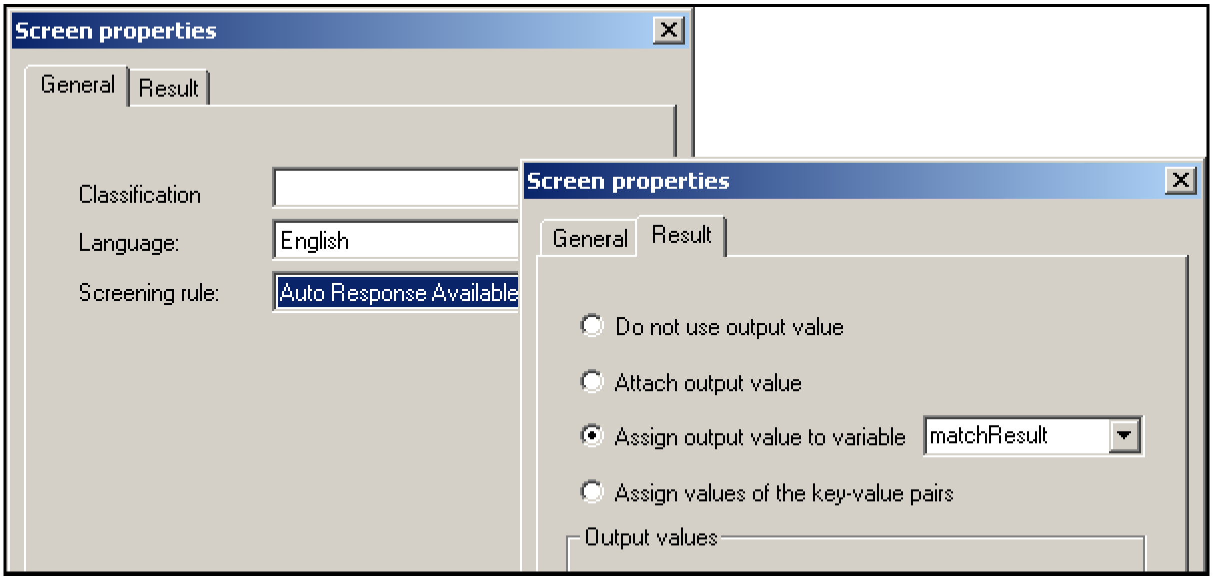
**Figure 114: Preliminary E-mail Screening Routing Strategy (IRD View)**

The "Summary of Flow" section below is limited to the start of the strategy, which screens using the `Auto Response Available` rule, and the end of the strategy, which uses the `Tech Support` screening rule and an Acknowledgement object (see Figure 122 on page 108).

## Summary of Flow

**Note:** The IRD objects that are described this section are keyed to the numbers in Figure 114.

1. A Screen object uses the `Auto Response Available` screening rule to determine if the e-mail can be handled with pre-written text from the Knowledge Manager standard response library. The results are written to a variable, `matchResult` (see Figure 115).



Classification:
Language: English
Screening rule: Auto Response Available

○ Do not use output value
○ Attach output value
● Assign output value to variable matchResult
○ Assign values of the key-value pairs
Output values

**Figure 115: Screen Object Properties Dialog Boxes, Auto Response Available**

### Screening Rule Output Example

Assume you write screening results to the `matchResult` variable and the variable is equal to:

```
ScreenRuleName:""|Id:""|ScreenRuleMatch:false|return:ok
```

This type of variable content indicates the e-mail did not match the rule.

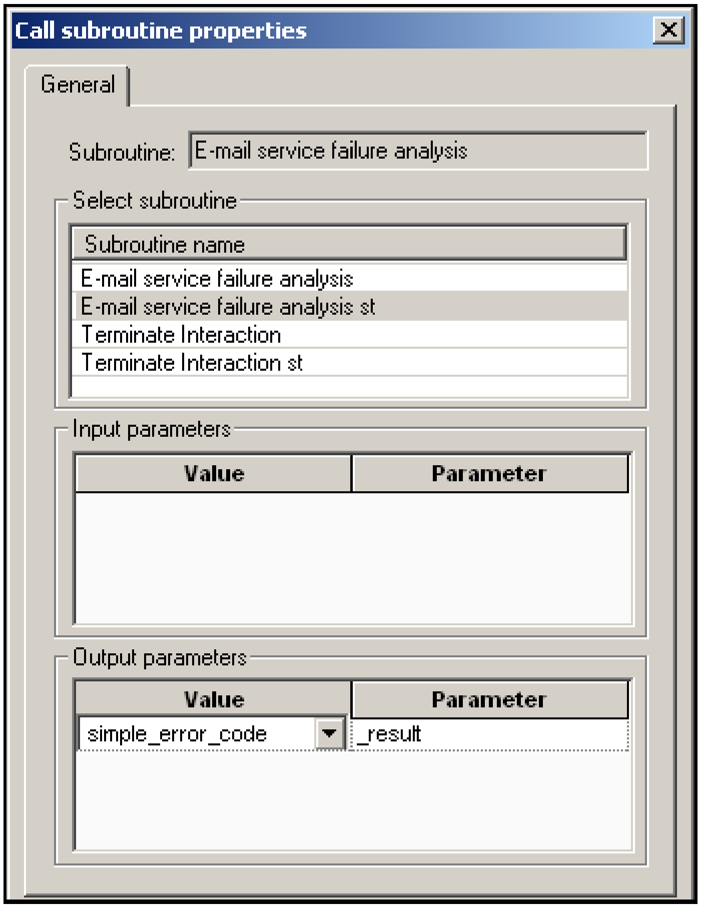
In contrast, assume you write the results to the `matchResult` variable and the variable is equal to:

```
00001a05F5U900QW:AutoReponseAvailable|ScreenRuleName:AutoResponseAv
ailble|Id:00001a05F5U900QW|ScreenRuleMatch:true|return:ok
```

This type of variable content indicates the e-mail matched the rule.

### Error Handling for Screening #1

2. If the initial screening could not occur, the interaction goes out the red port to a subroutine, which generates an error code. The subroutine is specified in a Call Subroutine object (see Figure 116).



Call subroutine properties

General

Subroutine: E-mail service failure analysis

Select subroutine

| Subroutine name |
|---|
| E-mail service failure analysis |
| E-mail service failure analysis st |
| Terminate Interaction |
| Terminate Interaction st |

Input parameters

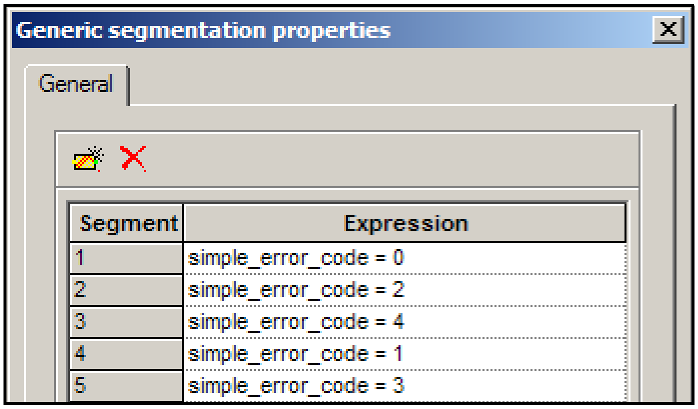
| Value | Parameter |
|---|---|

Output parameters

| Value | Parameter |
|---|---|
| simple_error_code | _result |

**Figure 116: Call Subroutine Properties—Failure Analysis**

3. A Generic Segmentation object directs interactions based on the error code returned by the subroutine (see Figure 117).



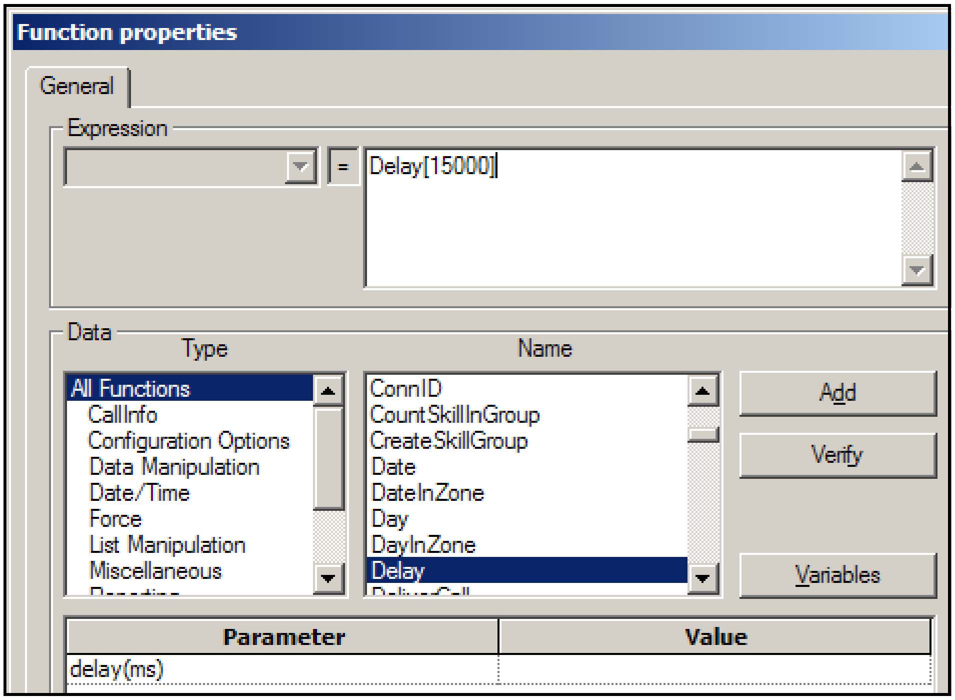
Generic segmentation properties

General

| Segment | Expression |
|---|---|
| 1 | simple_error_code = 0 |
| 2 | simple_error_code = 2 |
| 3 | simple_error_code = 4 |
| 4 | simple_error_code = 1 |
| 5 | simple_error_code = 3 |

**Figure 117: Generic Segmentation Properties—Error Codes**

4. If the returned error code equals `3`, the Delay function delays execution of the strategy and automatic response generation is tried again (see Figure 118 on page 105).



Function properties

Expression: = Delay[15000]

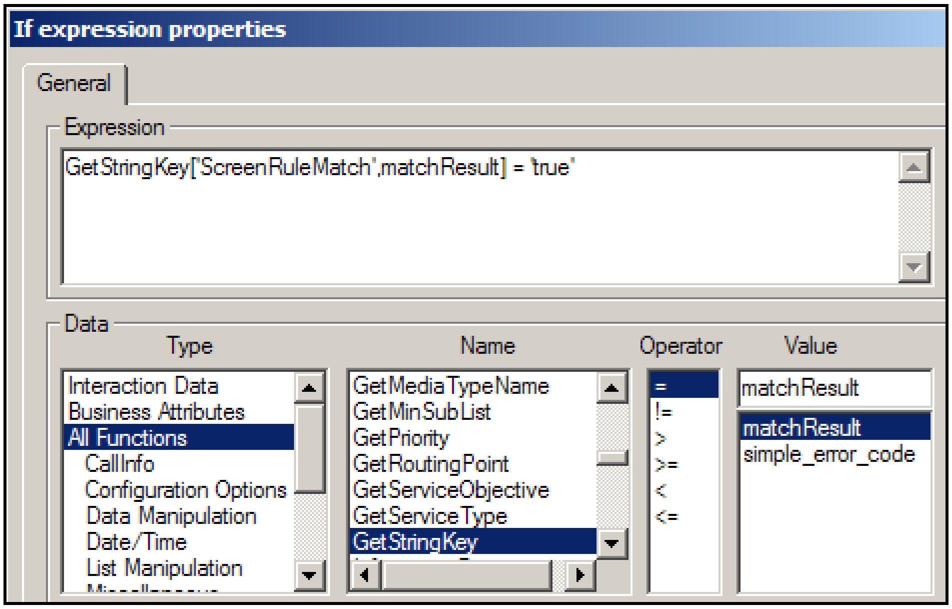
Data Type: All Functions

Name: Delay

| Parameter | Value |
|---|---|
| delay(ms) | |

Figure 118: Function Object Properties Dialog Box, Delay Function

5. If the screening is successful, the results in the `matchResult` variable in Figure 115 on page 103 are used in an If expression (see Figure 119).



If expression properties

Expression: GetStringKey['ScreenRuleMatch',matchResult] = 'true'

Data Type: All Functions

Name: GetStringKey

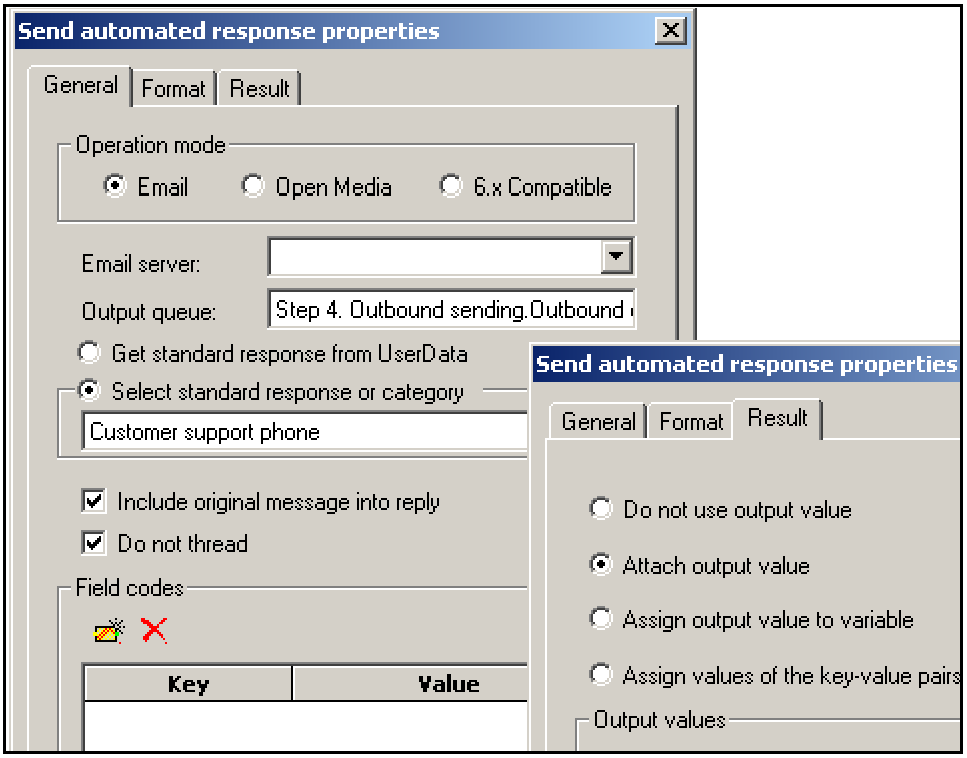
Operator: =

Value: matchResult

Figure 119: If Expression Properties Dialog Box

6. If the expression in Figure 119 is false, the e-mail goes to the second Screen object, which screens using the `Warranty Problem` rule.

7. If the expression is true (see Figure 119 on page 105), an Autoresponse object selects a standard response and attaches it to the interaction (see Figure 120).



Operation mode: ● Email ○ Open Media ○ 6.x Compatible

Email server:

Output queue: Step 4. Outbound sending.Outbound

○ Get standard response from UserData

● Select standard response or category

Customer support phone

☑ Include original message into reply

☑ Do not thread

○ Do not use output value

● Attach output value

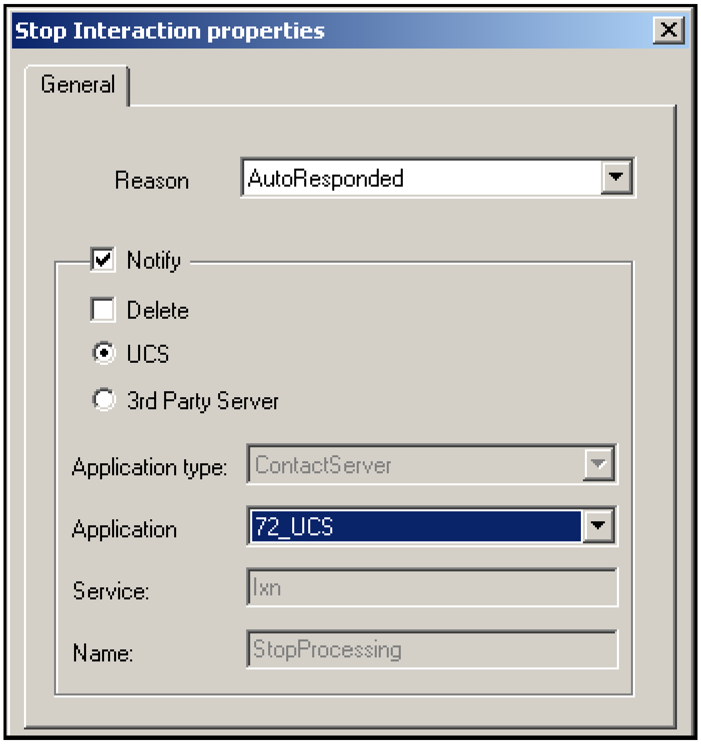
○ Assign output value to variable

○ Assign values of the key-value pairs

Figure 120: Autoresponse Properties Dialog Box

8. If an autoresponse e-mail cannot be generated, the e-mail goes out the red error port to a subroutine, which generates an error code. The name of the subroutine is specified in a Call Subroutine object (see Figure 116 on page 104).

9. If an automatic response is generated, a Stop Interaction object halts e-mail processing, supplies a Reason Code, and optionally notifies UCS or a 3rd-party server (see Figure 121).



Stop Interaction properties

General

Reason: AutoResponded

☑ Notify

□ Delete

● UCS

○ 3rd Party Server

Application type: ContactServer

Application: 72_UCS

Service: Ixn

Name: StopProcessing

**Figure 121: Stop Interaction Properties Dialog Box**

10. If UCS cannot be notified, the e-mail goes out the red error port to a subroutine, which generates an error code. The name of the subroutine is specified in a Call Subroutine object.

### Screening #2

If there are no screening results using the `Auto Response Available` rule, a second screening is performed using the `Warranty Problem` rule. Screening results are written to a variable, which is subsequently used in an If expression.

- If the expression is true (screening rule match found), the e-mail goes to a queue for e-mails to be forwarded.
- If the expression is not true (screening rule match not found), a third screening is performed with the `Tech Support` screening rule.

If the screening cannot be performed, the interaction goes out the red port to a Call Subroutine object for failure analysis.

### Screening #3

11. If no results with the `Warranty Problem` screening rule, the strategy then screens using the `Tech Support` screening rule (see Figure 122).

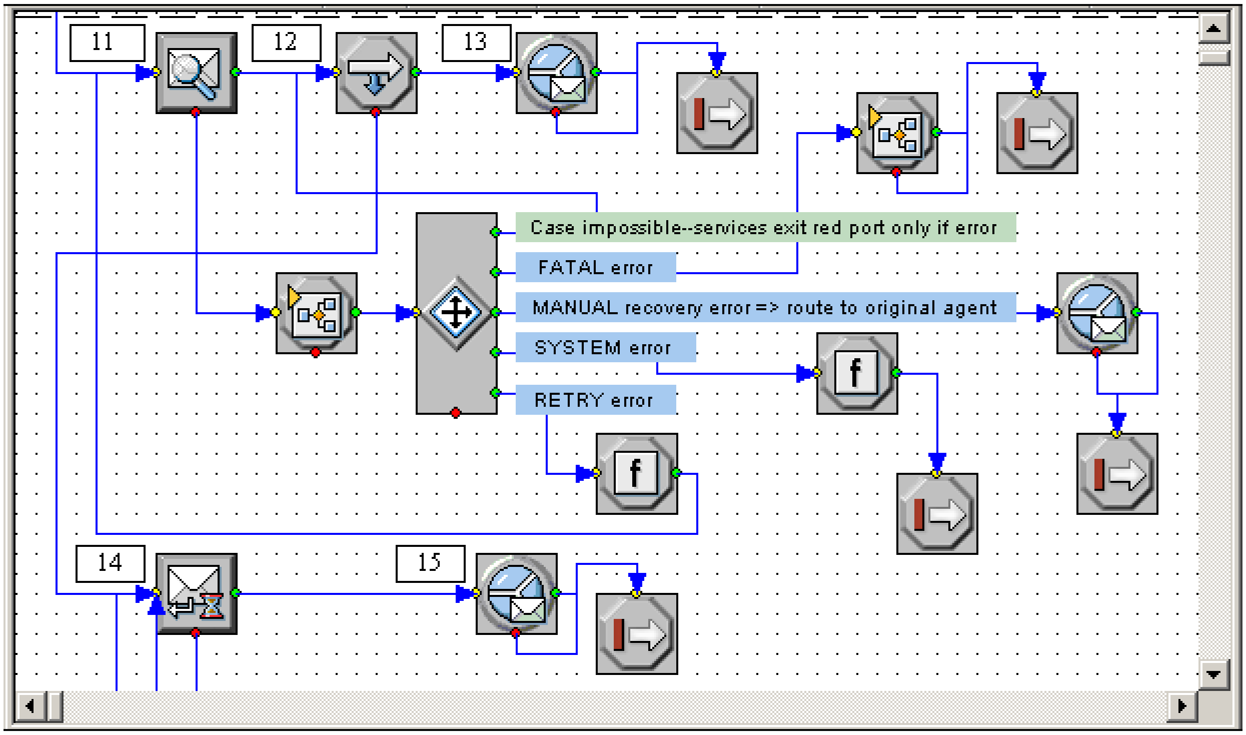
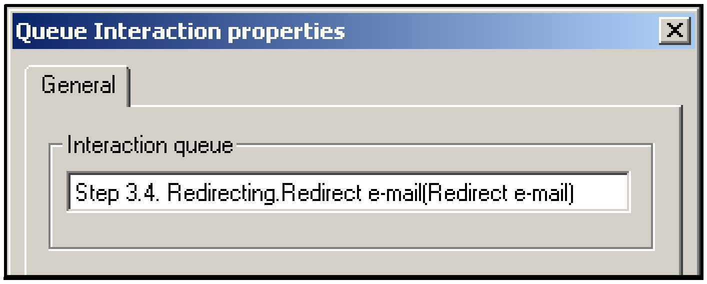

11
12
13
Case impossible--services exit red port only if error
FATAL error
MANUAL recovery error => route to original agent
SYSTEM error
RETRY error
f
14
15

**Figure 122: End of Preliminary E-mail Screening Routing Strategy**

12. If the e-mail matches the rule, screening results contained in the `matchResult` variable (see Figure 115 on page 103) are used in an If expression using an IRD If object (see Figure 119 on page 105).

13. If the expression is true, the e-mail goes to a queue for redirected e-mails (see Figure 123).


Queue Interaction properties

General

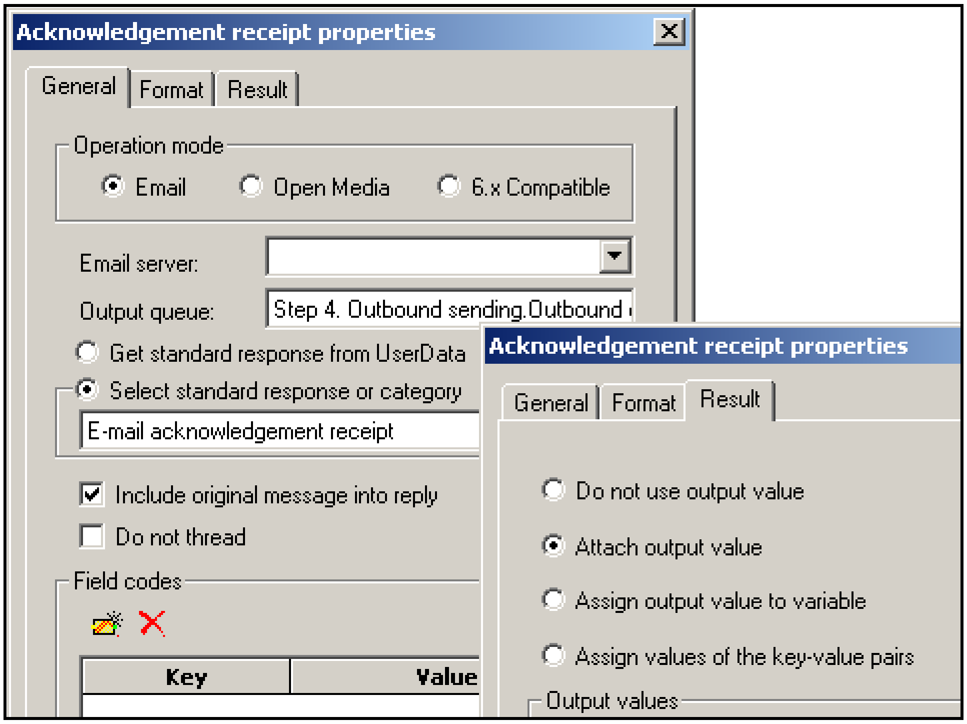
Interaction queue

Step 3.4. Redirecting.Redirect e-mail(Redirect e-mail)

**Figure 123: Queue Interaction Properties—Redirected E-mail Queue**

See “Redirecting an E-mail” on page 110 for additional information.

14. If the expression is false, the e-mail must be processed by an agent. In this case, an Acknowledgement object selects a standard response indicating that the customer’s e-mail has been redirected to Technical Support and attaches it to the interaction (see Figure 124).


Acknowledgement receipt properties
General
Format
Result

Operation mode: ● Email ○ Open Media ○ 6.x Compatible

Email server:

Output queue: Step 4. Outbound sending.Outbound

○ Get standard response from UserData

● Select standard response or category

E-mail acknowledgement receipt

☑ Include original message into reply

□ Do not thread

Field codes

| Key | Value |
|---|---|

Acknowledgement receipt properties
General
Format
Result

○ Do not use output value

● Attach output value

○ Assign output value to variable

○ Assign values of the key-value pairs

Output values

**Figure 124: Acknowledgement Properties Dialog Box**

15. The resulting e-mail is placed in an output queue for processing by agents.

**Note:** You can also use the Multi-Screen object. It does not require a conditional test after a screening. For more information, see “How To: Screen Multiple Rules and Use Screening Switch” on page 192.

# Processing of E-mail Attached Data

The Interaction Workflow Samples (see page 80) supply the following routing strategies that demonstrate processing of e-mail attached data:

- “Inbound E-mail Preprocessing Strategy” on page 86 (see “Interaction Sub-types” on page 87 and “Gem_Failure” on page 91).
- “Route E-mail to Original Agent Strategy” on page 94
- “Outbound E-mail 65X QA Strategy” on page 126

# Stopping an E-mail With a Reason Code

The Interaction Workflow Samples (see page 80) supply the following routing strategies that demonstrate stopping an e-mail with a Reason Code:

- "Preliminary E-mail Screening Strategy" on page 100 (see Figure 121 on page 107).
- "Redirect E-mail Processing Strategy" on page 110 (see Figure 129 on page 113).
- "Forward E-mail Processing Strategy" on page 115 (see Figure 134 on page 117).
- "Inbound Collaboration Reply Processing Strategy" on page 118 (see Figure 138 on page 121).
- "Chat Strategy - Create Transcript E-mail" on page 152 (see Figure 182 on page 154).

# Redirecting an E-mail

The business process `Step 3.4: Redirecting` (see page 85) contains the strategy `Redirect e-mail processing`, which is described in this section (see Figure 125).

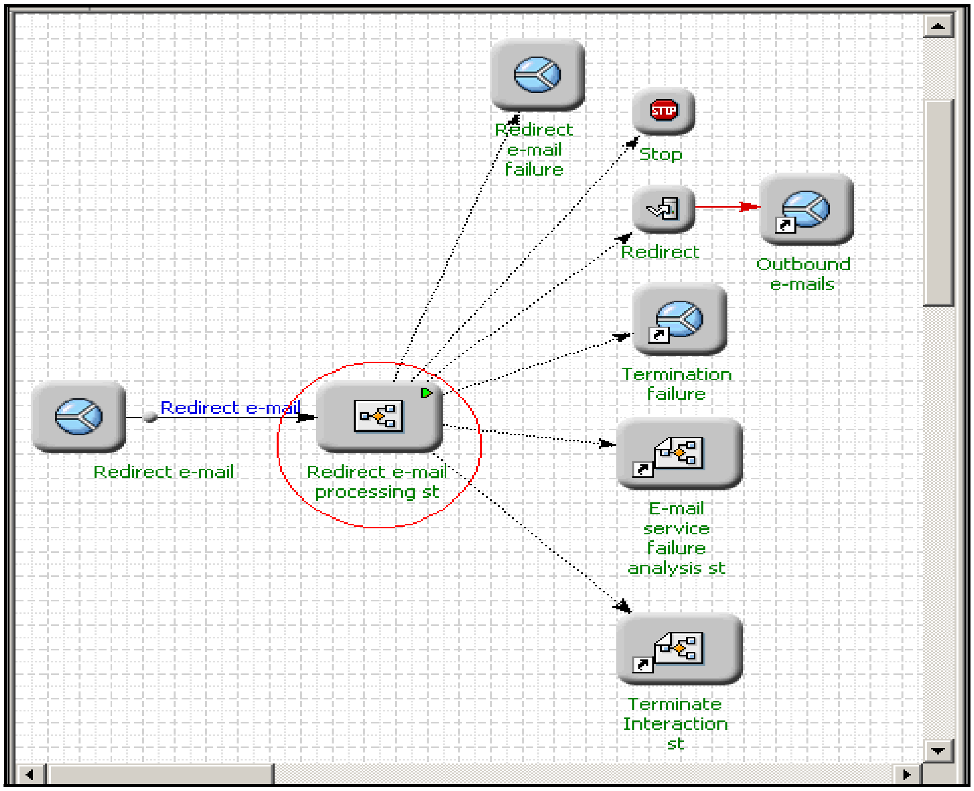


**Figure 125: Step 3.4: Redirecting E-mail Business Process**

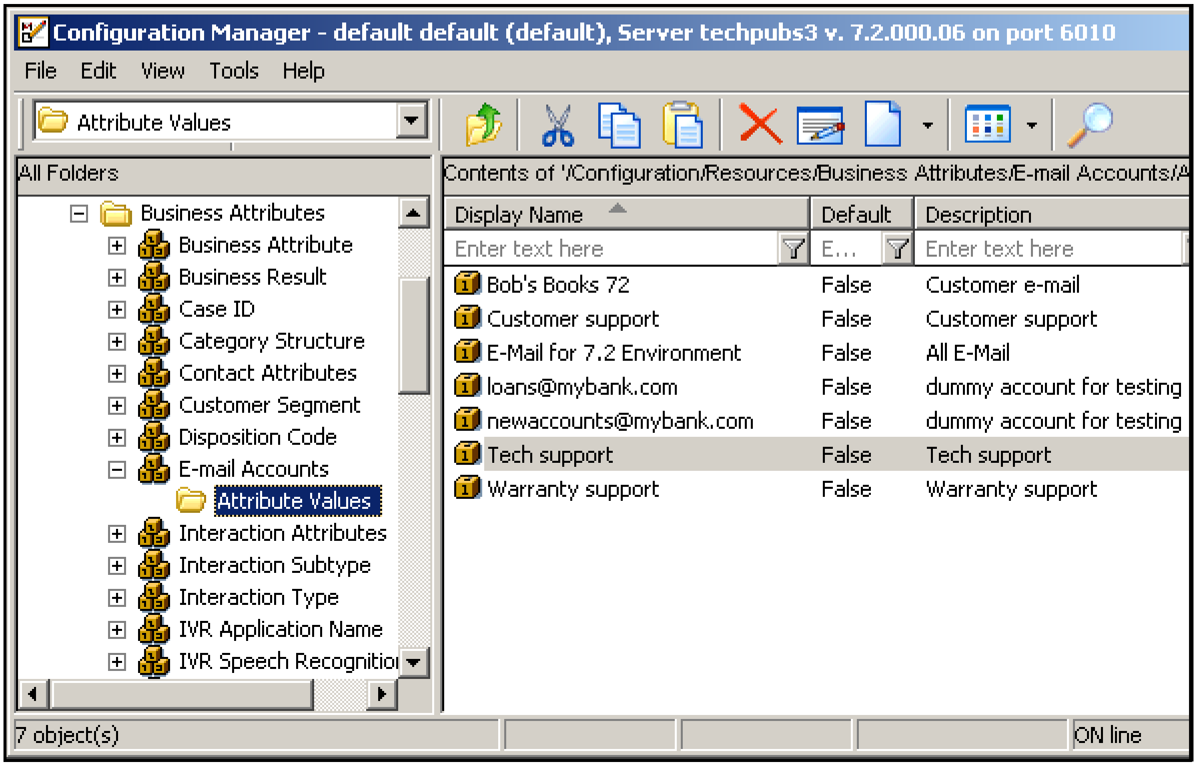
## Redirect E-mail Processing Strategy

The `Redirect e-mail` queue in Figure 125 gets interactions from the `Preliminary e-mail screening` strategy shown in Figure 113 on page 100, which determined that the e-mail matched the Tech support screening rule and therefore needs to be redirected. A view (Redirect e-mail) attached to the Redirect e-mail queue in Figure 125 extracts e-mails. A submitter submits them to the Redirect e-mail processing strategy.

**Note:** For more information on Interaction Design queues and views, see the *Universal Routing 8.1 Business Process User's Guide*.

## E-mail Accounts Business Attributes

The Redirect e-mail processing strategy requests E-mail Server to create an e-mail using an external address associated with the Tech Support E-mail Accounts business attribute (see Figure 126).


Configuration Manager - default default (default), Server techpubs3 v. 7.2.000.06 on port 6010
File
Edit
View
Tools
Help
Attribute Values
All Folders
Contents of '/Configuration/Resources/Business Attributes/E-mail Accounts/A

- Business Attributes
  - Business Attribute
  - Business Result
  - Case ID
  - Category Structure
  - Contact Attributes
  - Customer Segment
  - Disposition Code
  - E-mail Accounts
    - Attribute Values
  - Interaction Attributes
  - Interaction Subtype
  - Interaction Type
  - IVR Application Name
  - IVR Speech Recognitio

| Display Name | Default | Description |
|---|---|---|
| Enter text here | E... | Enter text here |
| Bob's Books 72 | False | Customer e-mail |
| Customer support | False | Customer support |
| E-Mail for 7.2 Environment | False | All E-Mail |
| loans@mybank.com | False | dummy account for testing |
| newaccounts@mybank.com | False | dummy account for testing |
| Tech support | False | Tech support |
| Warranty support | False | Warranty support |

7 object(s) ON line

Figure 126: Configuration Manager, E-mail Accounts Business Attributes

Once the e-mail is generated, it goes into a queue for outbound e-mails.

Figure 127 shows the Redirect e-mail processing strategy in the IRD Routing Design window.

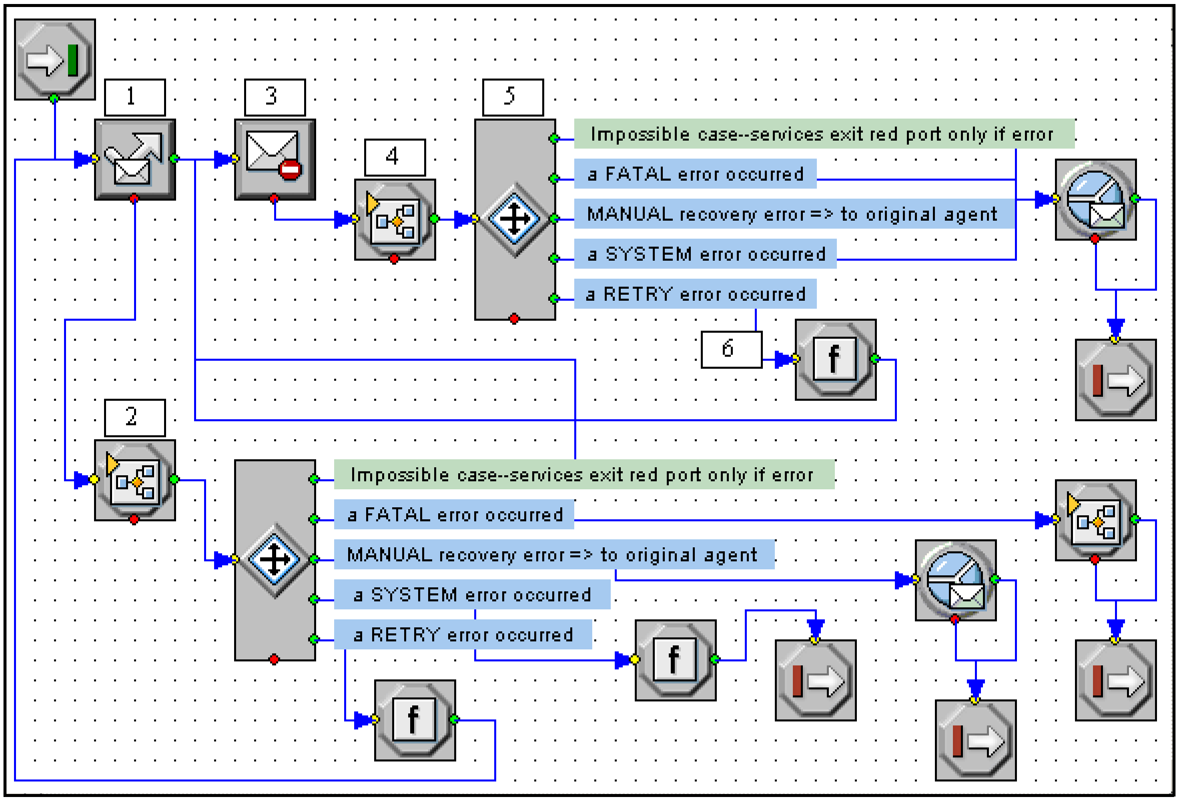
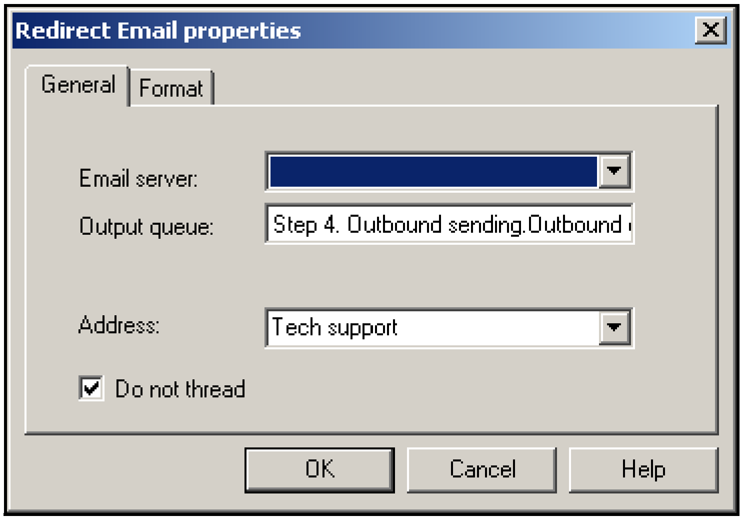

1
3
4
5
6
2
Impossible case--services exit red port only if error
a FATAL error occurred
MANUAL recovery error => to original agent
a SYSTEM error occurred
a RETRY error occurred
Impossible case--services exit red port only if error
a FATAL error occurred
MANUAL recovery error => to original agent
a SYSTEM error occurred
a RETRY error occurred

**Figure 127: Redirect E-mail Processing Strategy**

## Summary of Flow

**Note:** The IRD objects that are described this section are keyed to the numbers in Figure 127. The summary excludes descriptions of some error processing objects.

1. A Redirect E-Mail object requests E-mail Server (through Interaction Server) to create an e-mail using the external address associated with the `Tech Support E-mail Accounts` business attribute (see Figure 126 on page 111). It also instructs to put in the e-mail in a queue called `Step 4.Outbound sending.Outbound e-mails` (see Figure 128).



Redirect Email properties

Email server:

Output queue: Step 4. Outbound sending.Outbound

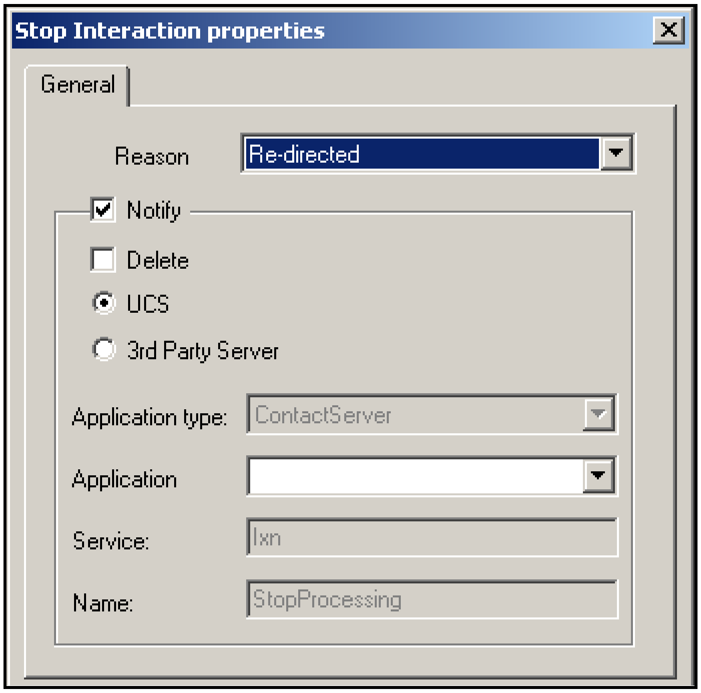
Address: Tech support

☑ Do not thread

**Figure 128: Redirect E-Mail Properties—Tech Support External Address**

**Note:** `Do not thread` in Figure 128 instructs E-mail Server not to create a record for the new e-mail in the Universal Contact Server database.

2. If the e-mail cannot be created, the interaction goes out the red error port to a subroutine, which generates an error code. The name of the subroutine is specified in a Call Subroutine object (see Figure 116 on page 104).
3. If the e-mail can be created, a Stop Interaction object notifies UCS through Interaction Server that processing is finished and supplies a reason code (see Figure 129).



Stop Interaction properties

Reason: Re-directed

☑ Notify

□ Delete

● UCS

○ 3rd Party Server

Application type: ContactServer

Application:

Service: Ixn

Name: StopProcessing

**Figure 129: Stop E-mail Properties—Redirected Reason Code**

4. If Interaction Server cannot notify UCS, the interaction goes out the red error port to a subroutine, which generates an error code. The name of the subroutine is specified in a Call Subroutine object (see Figure 116 on page 104).
5. A Generic Segmentation object directs interactions based on the error code returned by the subroutine (see Figure 117 on page 104).
6. If the returned error code equals `3,` a Function object delays execution of the strategy and Interaction Server tries again to notify UCS (see Figure 118 on page 105).

# Forwarding an E-mail

The business process `Step 3.3: Forwarding` contains the strategy `Forward e-mail processing` described in this section (see Figure 130).

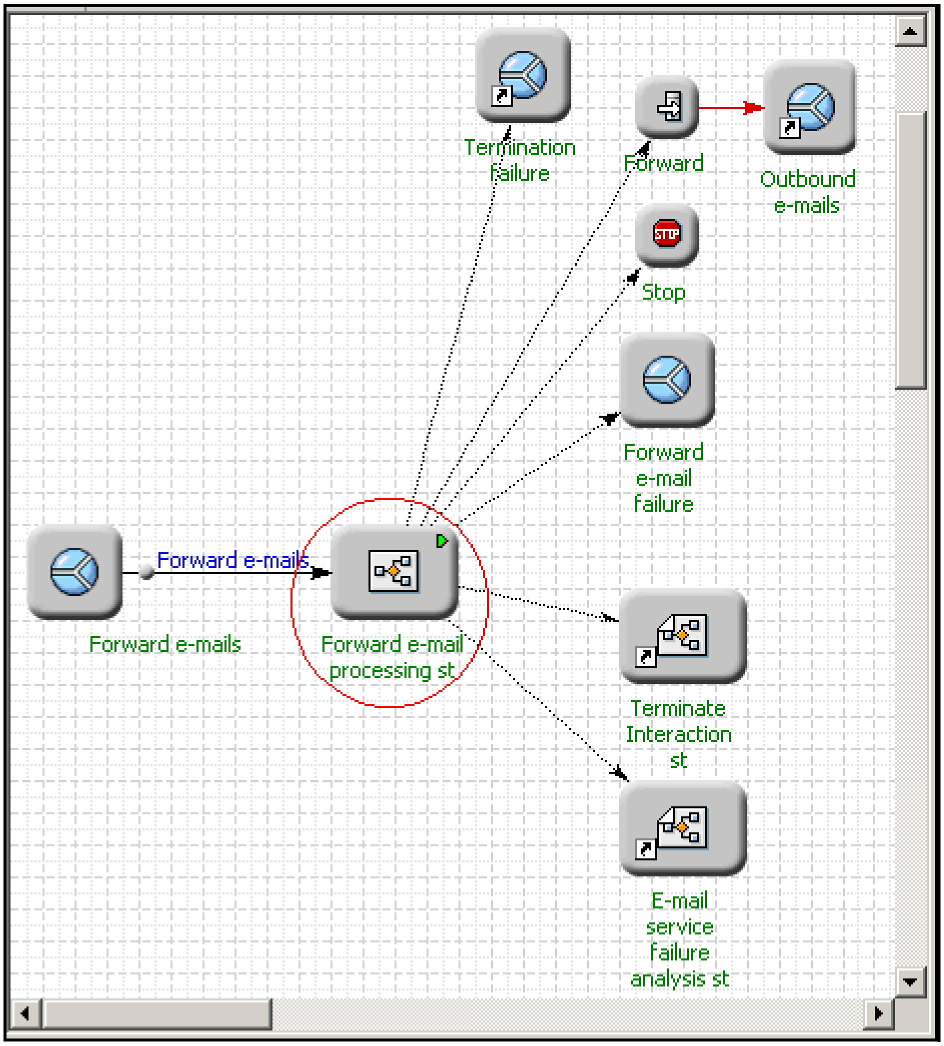


**Figure 130: Step 3.3: Forwarding E-mail Business Process**

# Forward E-mail Processing Strategy

The Forward e-mails queue in Figure 130 on page 114 gets interactions from the Preliminary e-mail screening strategy shown in Figure 113 on page 100. It determined that the e-mail matched the Warranty support screening rule and therefore needs to be forwarded. A view attached to the Forward e-mails queue extracts e-mails. A submitter object submits them to the Forward e-mail processing strategy shown in Figure 130 on page 114.

## E-mail Accounts Business Attributes

The Forward e-mail processing strategy requests E-mail Server to create an e-mail using to external addresses associated with the Warranty Support E-mail Accounts business attribute (see Figure 126 on page 111). Once the e-mail is generated, it goes into a queue for outbound e-mails (see Figure 130 on page 114).

Figure 131 shows the Forward e-mail processing strategy in the IRD Routing Design window.

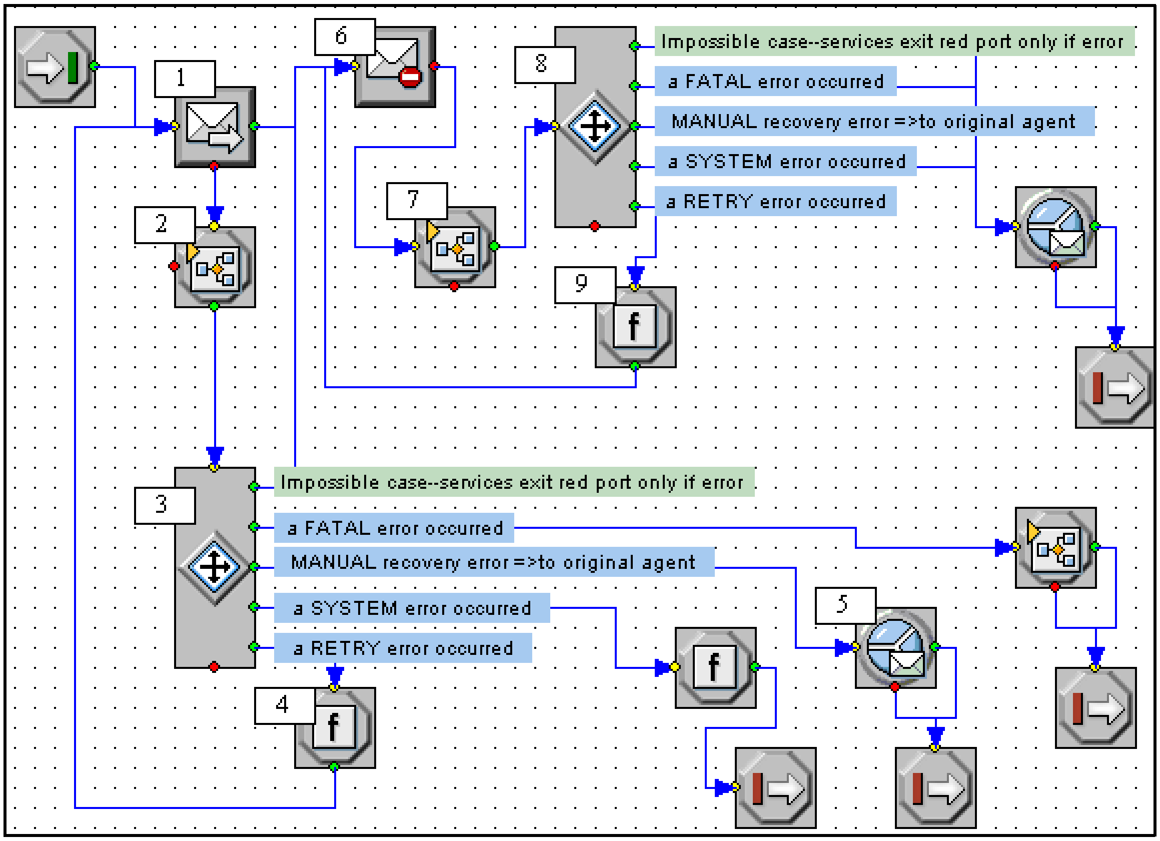
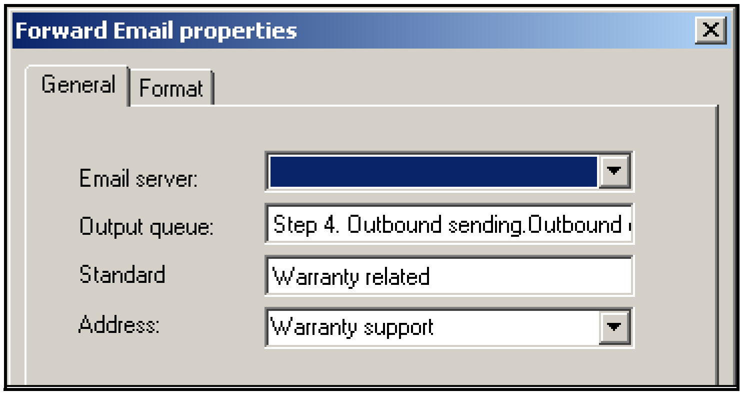

6
1
8
Impossible case--services exit red port only if error
a FATAL error occurred
MANUAL recovery error =>to original agent
a SYSTEM error occurred
a RETRY error occurred
2
7
9
f
3
Impossible case--services exit red port only if error
a FATAL error occurred
MANUAL recovery error =>to original agent
a SYSTEM error occurred
a RETRY error occurred
4
f
5
f

**Figure 131: Forward E-mail Processing Strategy**

## Summary of Flow

**Note:** The IRD objects that are described this section are keyed to the numbers in Figure 131 on page 115. The description excludes descriptions of some error processing objects.

1. A Forward E-Mail object requests E-mail Server to create an e-mail using the external address associated with the `Warranty Support E-mail Accounts` business attribute (see Figure 126 on page 111) and put the e-mail in a queue called `Step 4. Outbound sending.Outbound e-mails` (see Figure 132).



Email server:

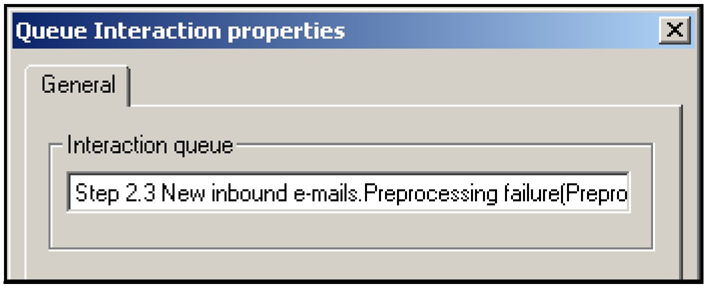
Output queue: Step 4. Outbound sending.Outbound e

Standard Warranty related

Address: Warranty support

Figure 132: Forward E-Mail Properties—Warranty Support Address

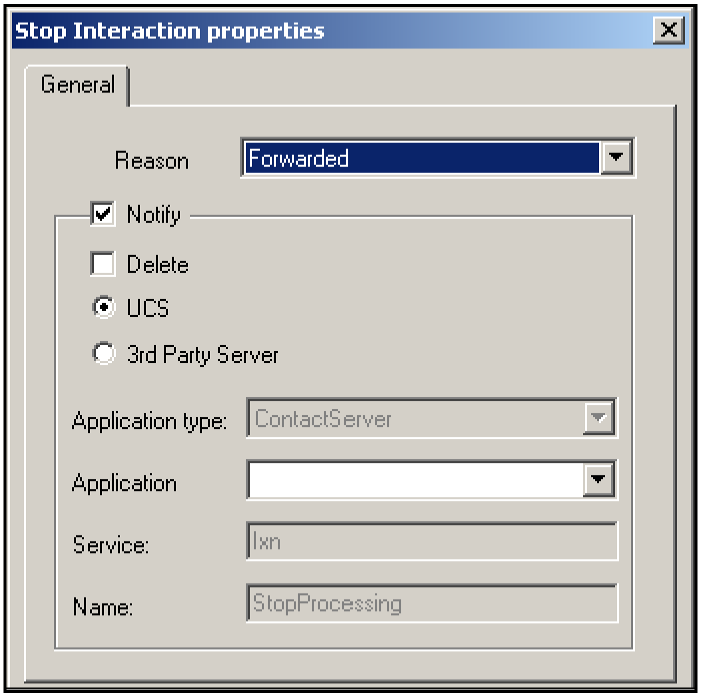
2. If E-mail Server cannot create the e-mail to be forwarded, the interaction goes out bottom red error port to a subroutine, which generates an error code. The name of the subroutine is specified in a Call Subroutine object (see Figure 116 on page 104).
3. A Generic Segmentation object directs interactions based on the error code returned by the subroutine (see Figure 117 on page 104).
4. If the returned error code equals `3,` the Delay function delays execution of the strategy and Interaction Server tries again to notify UCS (see Figure 118 on page 105).
5. If the returned error code equals `4`, indicating a manual recovery error, the e-mail is routed to a queue named `Step 2.3 New inbound e-mails.Preprocessing failure` (see Figure 133).



Interaction queue: Step 2.3 New inbound e-mails.Preprocessing failure(Prepro

Figure 133: Queue Interaction Properties—Preprocessing Failure

6. If E-mail Server creates an e-mail to be forwarded (Step 1), the interaction goes out the green port of the Forward E-Mail object to a Stop Interaction object, which is configured to notify UCS (through Interaction Server) that processing is finished and supply a reason code (see Figure 121).



Reason: Forwarded

☑ Notify

□ Delete

● UCS

○ 3rd Party Server

Application type: ContactServer

Application:

Service: Ixn

Name: StopProcessing

Figure 134: Stop Interaction Properties—Forwarded Reason Code

# Collaboration Reply Sending

The Business process `Step 2.2: Inbound Collaboration Reply` (see page 85) contains the strategy `Inbound collaboration reply processing` described in this section (see Figure 135).

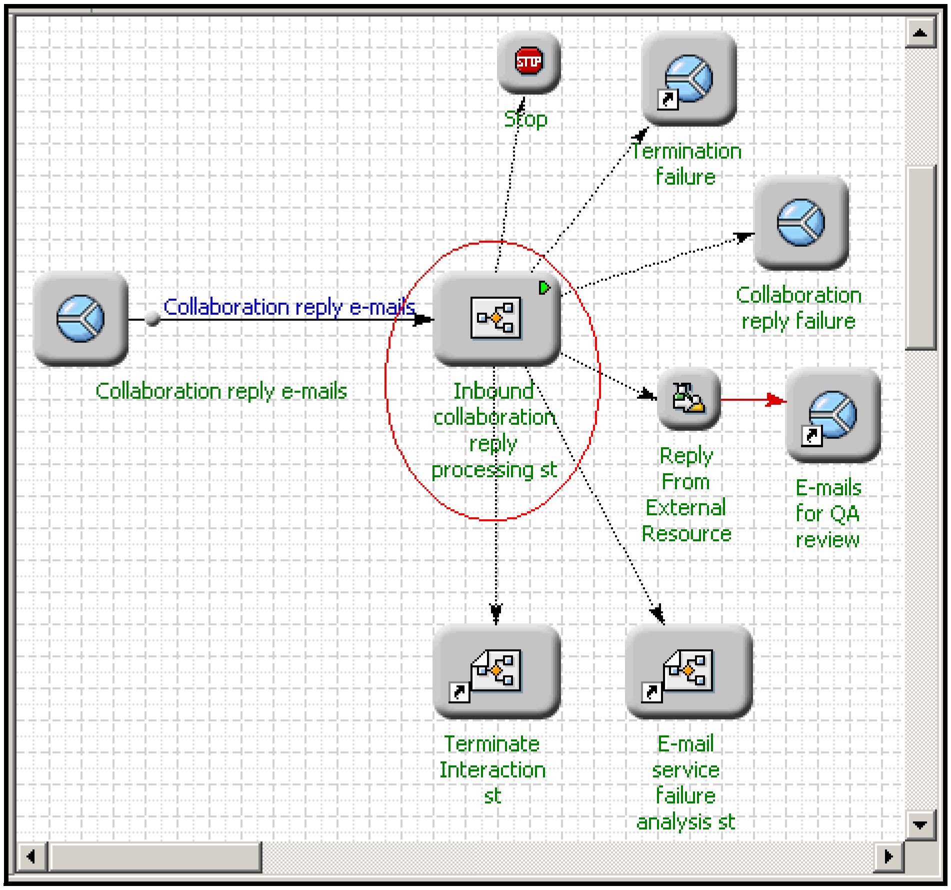

Stop
Termination failure
Collaboration reply failure
Collaboration reply e-mails
Collaboration reply e-mails
Inbound collaboration reply processing st
Reply From External Resource
E-mails for QA review
Terminate Interaction st
E-mail service failure analysis st

**Figure 135: Step 2.2: Inbound Collaboration Reply Business Process**

# Inbound Collaboration Reply Processing Strategy

The `Collaboration reply e-mails` queue in Figure 135 gets interactions from the `Inbound e-mail preprocessing` strategy shown in Figure 92 on page 86. A view attached to the `Collaboration reply e-mails` queue in Figure 135 extracts e-mails. A submitter object submits them to the `Inbound collaboration reply processing` strategy.

**Note:** For more information on Interaction Design queues, views, and submitters, see the *Universal Routing 8.1 Business Process User's Guide*.

## About Collaboration Replies

Within Genesys e-mail processing, collaboration replies are e-mails from *external resources*. An external resource is a name for any object outside the contact center. It may be an external agent or another contact center.

A collaboration reply inbound e-mail arrives as a result of the following actions:

- The parent e-mail is forwarded by an agent (via the Forward E-Mail object) requesting collaboration with another agent.
- The collaborating agent constructs a response.
- The collaborating agent sends the reply back to the requesting agent creating a collaboration reply inbound e-mail.

The Reply E-Mail From External Resource object in the `Inbound collaboration reply strategy` takes the collaboration reply as input, extracts the reply text from it, creates a customer reply outbound e-mail with it, and submits the e-mail to Interaction Server by putting the e-mail in an interaction queue.

Figure 136 shows the `Inbound collaboration reply processing` strategy.

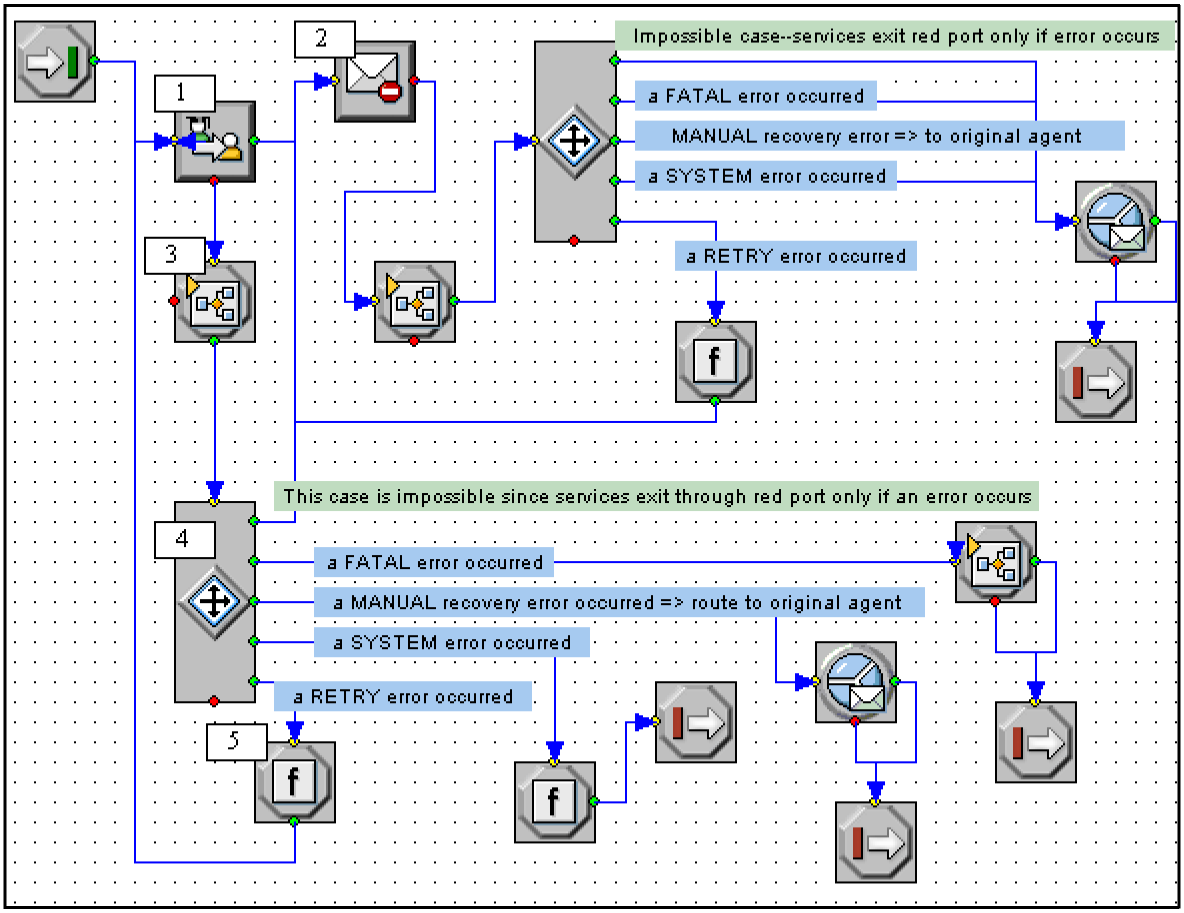
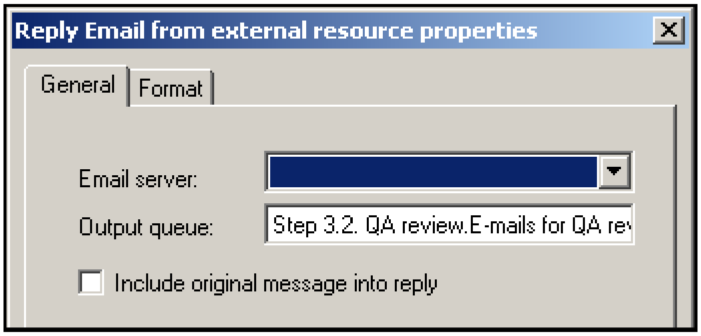

1
2
3
4
5
Impossible case--services exit red port only if error occurs
a FATAL error occurred
MANUAL recovery error => to original agent
a SYSTEM error occurred
a RETRY error occurred
f
This case is impossible since services exit through red port only if an error occurs
a FATAL error occurred
a MANUAL recovery error occurred => route to original agent
a SYSTEM error occurred
a RETRY error occurred
f
f

**Figure 136: Inbound Collaboration Reply Processing Strategy**

## Summary of Flow

**Note:** The IRD objects that are described this section are keyed to the numbers in Figure 136. The description excludes descriptions of some error processing objects.

1. As shown in Figure 137, a Reply E-Mail From External Resource object takes the e-mail reply from the collaborating agent and submits a request to E-mail Server to:
   - Create the customer reply outbound e-mail.
   - Place the customer reply outbound e-mail in a queue.



Email server:

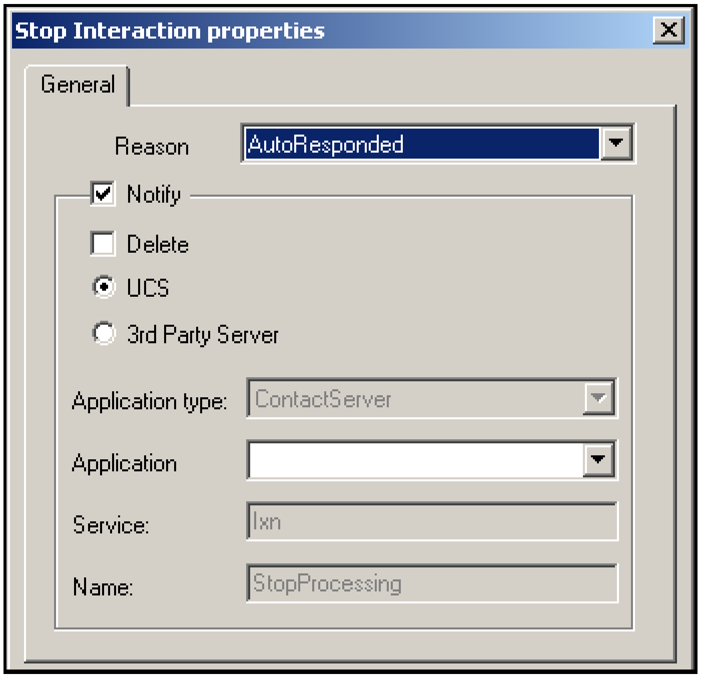
Output queue: Step 3.2. QA review.E-mails for QA rev

□ Include original message into reply

Figure 137: Reply E-Mail From External Resource Properties Dialog Box

**Note:** If the `Email server` field is left empty as in Figure 137, Interaction Server uses the first available E-mail Server in its Connections list.

2. If the customer reply e-mail can be created, a Stop Interaction object notifies UCS (through Interaction Server) that processing is finished and supplies a reason code (see Figure 138).



Reason: AutoResponded

☑ Notify

□ Delete

● UCS

○ 3rd Party Server

Application type: ContactServer

Application:

Service: Ixn

Name: StopProcessing

**Figure 138: Stop Interaction Properties—Auto-Responded Reason Code**

3. If the customer reply e-mail cannot be created, the e-mail goes out the red error port to a subroutine, which generates an error code. The name of the subroutine is specified in a Call Subroutine object (see Figure 116 on page 104).
4. A Generic Segmentation object directs interactions based on the error code returned by the subroutine (see Figure 117 on page 104).
5. If the returned error code equals `3,` the Delay function delays execution of the strategy and customer reply e-mail creation is tried again via the Reply E-Mail From External Resource object (see Figure 118 on page 105).

# Automatic Treatment With an Acknowledgement E-mail

The Interaction Workflow Samples (see page 80) contain the following routing strategy that demonstrates automatic treatment with an acknowledgement e-mail (Acknowledgement object):

- "Preliminary E-mail Screening Strategy" on page 100 (see Figure 122 on page 108 and Figure 124 on page 109).

# Autoresponse E-mail When Applicable

The Interaction Workflow Samples (see page 80) contain the following routing strategy that demonstrates sending an automatic response to the customer when applicable (Autoresponse object):

- "Preliminary E-mail Screening Strategy" on page 100 (see Figure 114 on page 102 and Figure 120 on page 106).

# Routing E-mails to Agents

The business process `Step 3.1: Processing By Agents` (see page 85) contains the strategy `E-mail distribution for processing` described in this section (see Figure 139).

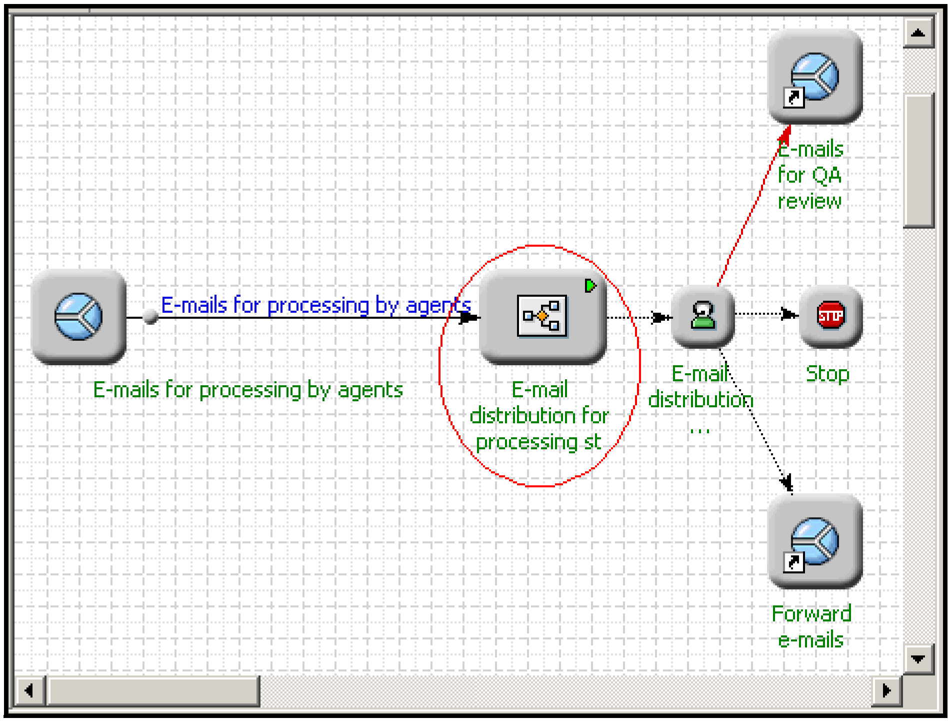

E-mails for processing by agents
E-mails for processing by agents
E-mail distribution for processing st
E-mail distribution ...
E-mails for QA review
STOP
Stop
Forward e-mails

**Figure 139: Step 3.1: Processing By Agents Business Process**

## E-mail Distribution for Processing Strategy

The `E-mails for processing by agents` queue in Figure 139 gets interactions from the `Preliminary e-mail screening` strategy (see Figure 140).

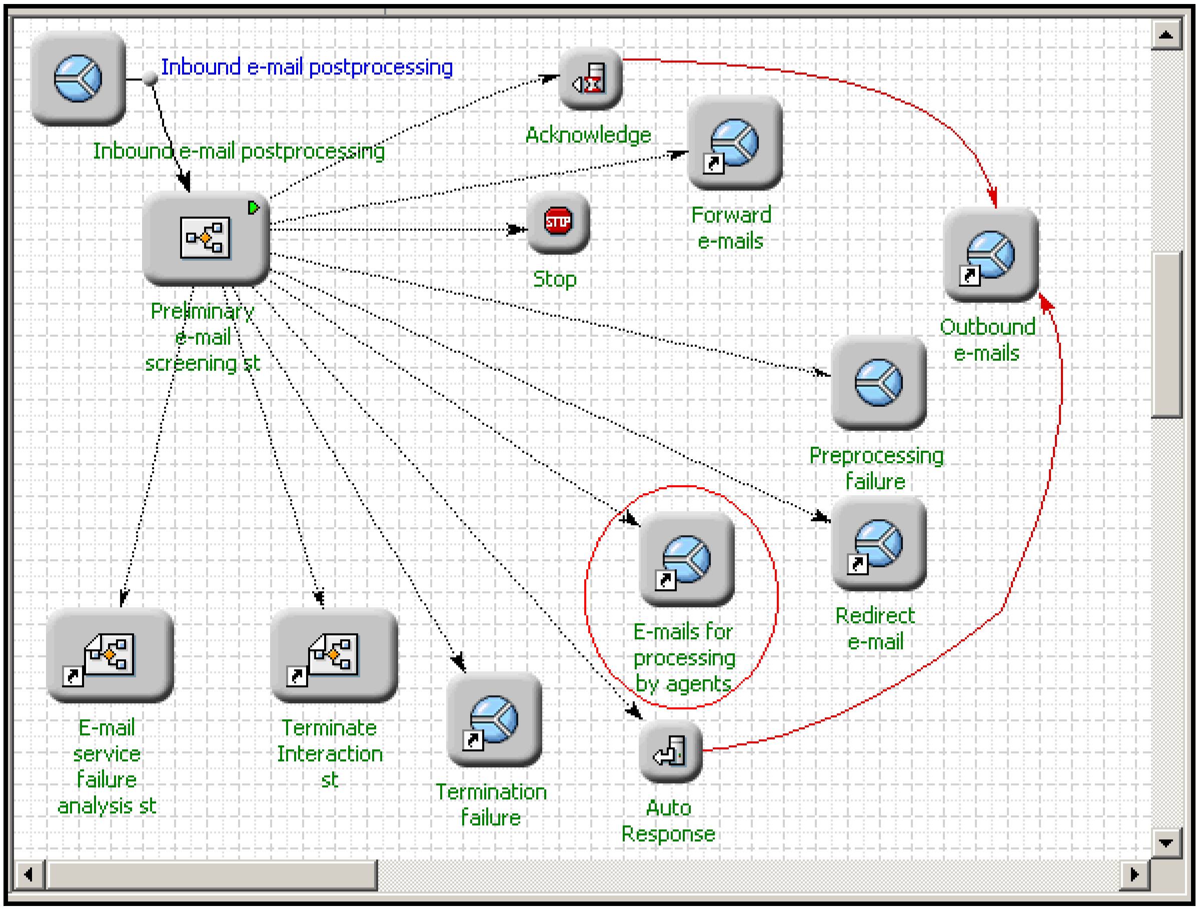

Inbound e-mail postprocessing
Inbound e-mail postprocessing
Acknowledge
Forward e-mails
Stop
Outbound e-mails
Preliminary e-mail screening st
Preprocessing failure
E-mails for processing by agents
Redirect e-mail
E-mail service failure analysis st
Terminate Interaction st
Termination failure
Auto Response

**Figure 140: Step 2.3: New Inbound E-mails Business Process**

As shown in Figure 139 on page 122, a view attached to the `E-mails for processing by agents` queue extracts e-mails. A submitter object submits them to the `E-mail distribution for processing` strategy.

This simple strategy routes to the agent group specified in the Route Interaction object. Figure 141 shows the `E-mail distribution for processing` strategy.

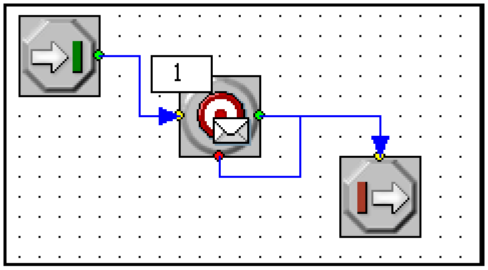
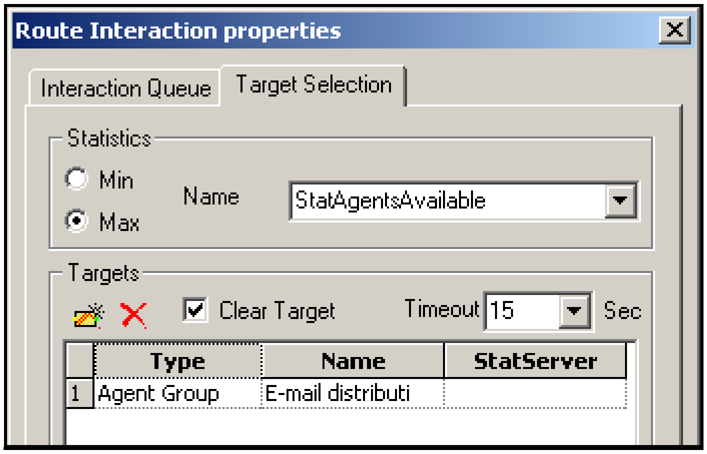


**Figure 141: E-mail Distribution For Processing Strategy**

## Summary of Flow

1. The Route Interaction object routes to an Agent Group target. The name of the `Agent Group` is `E-mail distribution for processing` (see Figure 142).



Statistics

○ Min
● Max

Name: StatAgentsAvailable

Targets

☑ Clear Target

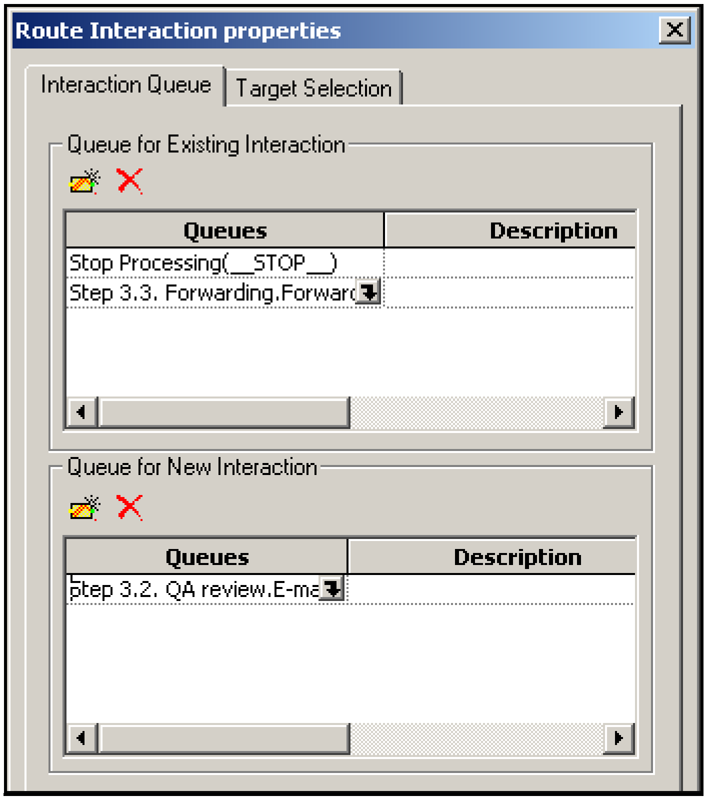
Timeout: 15 Sec

| | Type | Name | StatServer |
|---|---|---|---|
| 1 | Agent Group | E-mail distributi | |

**Figure 142: Route Interaction Properties—Target Selection Tab**

URS uses the `StatAgentsAvailable` statistic to select the target that has the maximum number of available agents if more than one target is available.

The Route Interaction object also has an Interaction Queue tab (see Figure 143).



Queue for Existing Interaction

| Queues | Description |
|---|---|
| Stop Processing(__STOP__) | |
| Step 3.3. Forwarding.Forwarc | |

Queue for New Interaction

| Queues | Description |
|---|---|
| Step 3.2. QA review.E-ma | |

**Figure 143: Route Interaction Properties—Interaction Queue Tab**

The optional Interaction Queue tab enables you to specify two types of queues:

- Queues for existing interactions (the queue where the agent should place the e-mail response when the response is part of an existing interaction).
- Queues for new interactions (the queue where the agent should place the interaction when a new e-mail is created).

  A `Description` (optional) appears as a hint on the agent desktop as to where to place the interaction.

**Note:** For more information on Interaction Design queues, views, and submitters, see the *Universal Routing 8.1 Business Process User's Guide*.

# Assigning Failure (Error) Codes to E-mails

The Interaction Workflow Samples (see page 80) contain the following routing strategy that demonstrates assigning failure codes to interactions:

- "Preliminary E-mail Screening Strategy" on page 100 (see Figure 116 on page 104 and Figure 117 on page 104).

# Promoting E-mails That Failed Pre-Routing to the Next Process

The Interaction Workflow Samples (see page 80) contain the following routing strategy that demonstrates promoting an interaction that failed in pre-routing to the next process:

- "Inbound E-mail Preprocessing Strategy" on page 86 (see Figure 101 on page 92).

# Routing E-mails for QA Review

The business process `Step 3.2: QA Review` (see page 85) contains the strategy `Outbound e-mail 65x QA` described in this section (see Figure 92).

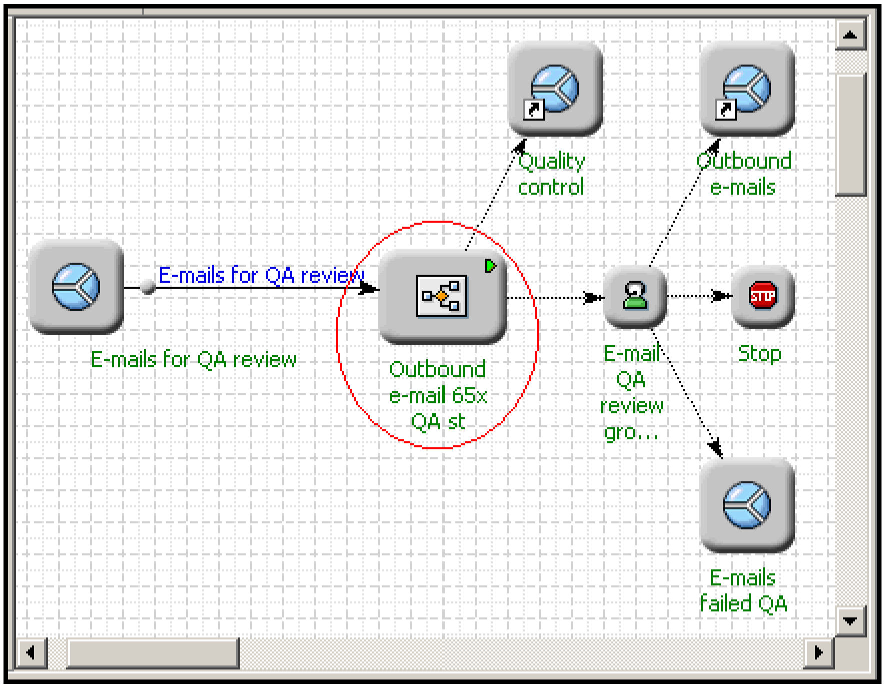


**Figure 144: Step 3:2: QA Review Business Process**

## Outbound E-mail 65X QA Strategy

The `E-mails for QA review` queue in Figure 144 gets interactions from the `E-mail distribution for processing` strategy (see Figure 139 on page 122).

A view attached to the `E-mails for QA review` queue in Figure 144 extracts e-mails. A submitter submits e-mails to the `Outbound e-mail 65x QA` strategy.

This strategy determines whether an agent’s e-mail response has undergone QA review. If the e-mail response has not undergone QA review, the strategy then determines whether QA review is necessary based on the skill level of the responding agent.

- If the responding agent does not have the default skill level or higher, the e-mail response is routed to QA for checking.
- If the responding agent has the default skill level or higher, QA review is not necessary so the e-mail goes to an outbound queue to be sent to the customer.

Figure 145 shows the `Outbound e-mail 65x QA` strategy.

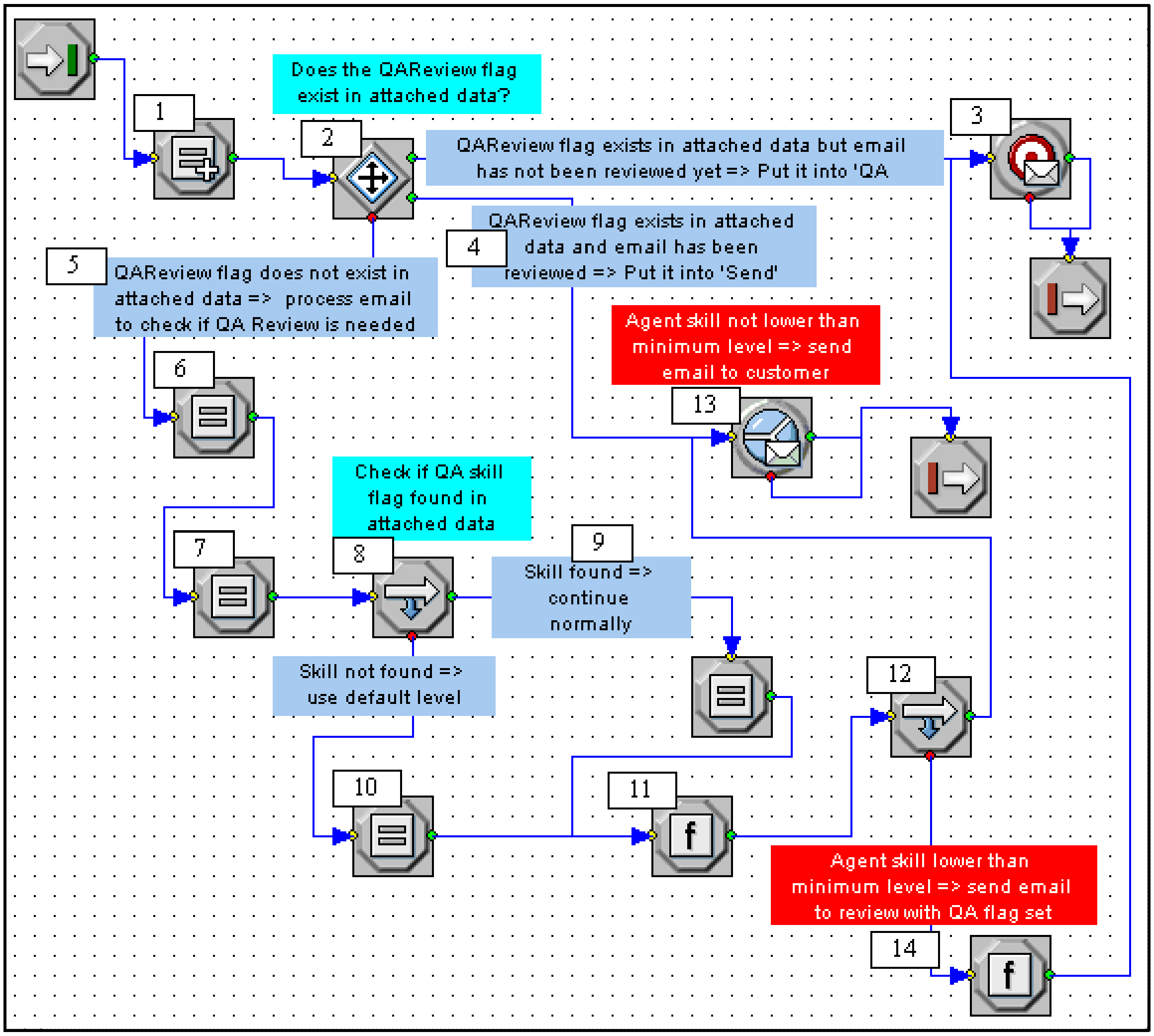

Does the QAReview flag exist in attached data?
1
2
3
QAReview flag exists in attached data but email has not been reviewed yet => Put it into 'QA
4
QAReview flag exists in attached data and email has been reviewed => Put it into 'Send'
5
QAReview flag does not exist in attached data => process email to check if QA Review is needed
Agent skill not lower than minimum level => send email to customer
13
6
Check if QA skill flag found in attached data
7
8
9
Skill found => continue normally
Skill not found => use default level
12
10
11
f
Agent skill lower than minimum level => send email to review with QA flag set
14
f

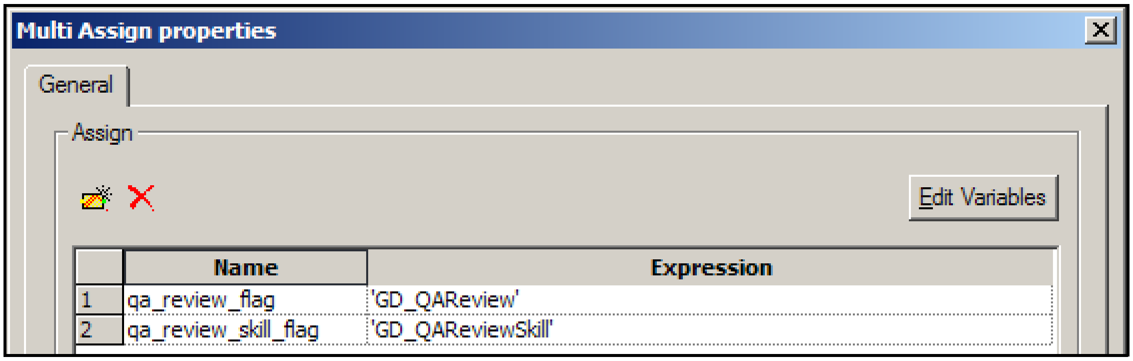
**Figure 145: Outbound E-mail 65x QA Strategy**

## Summary of Flow

**Note:** The IRD objects that are described this section are keyed to the numbers in Figure 145.

1. A MultiAssign IRD object assigns values to two predefined variables:
   - The variable `qa_review_flag` determines whether QA has reviewed the e-mail response.
   - The variable `qa_review_skill_flag` contains the skill level of the agent constructing the response.

   Figure 146 shows the MultiAssign Properties dialog box assigning interaction attached data values (`GD_QAReview` and `GD_QAReviewSkill`) to the two variables listed above.



|  | Name | Expression |
|---|---|---|
| 1 | qa_review_flag | 'GD_QAReview' |
| 2 | qa_review_skill_flag | 'GD_QAReviewSkill' |

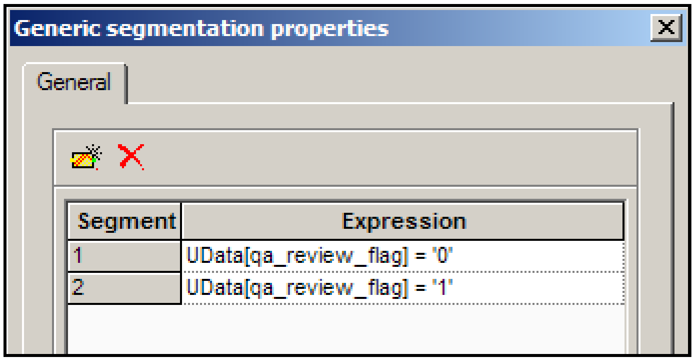
**Figure 146: MultiAssign Object Properties Dialog Box**

## Strategy Assumptions

This `Outbound e-mail 65x QA` strategy assumes the following:

- The value of the key `GD_QAReview` is already contained in the interaction attached data (added by the agent desktop). It determines what should be done with the current e-mail response.
  - If `GD_QAReview=1`, the e-mail response has already been reviewed and is ready to be sent to the customer.
  - If `GD_QAReview=0`, the e-mail response has already gone through the decision process and is ready to be reviewed by QA.
  - If `GD_QAReview` is undefined or empty, the e-mail response has to go through the decision process, which uses the value of the `GD_QAReviewSkill` attached data and sets the `GD_QAReview` with required value.
- The value of key `GD_QAReviewSkill` (already contained in the interaction attached data as added by the agent desktop) contains the skill level of the agent who constructed the e-mail response.
- A third piece of attached data is also assumed to be in the interaction: the employee ID of the QA person who reviewed the e-mail response. This is the `GD_ReviewerEmployeeId`, and is assumed to have been set by the agent desktop when the reviewer clicked the Send button.

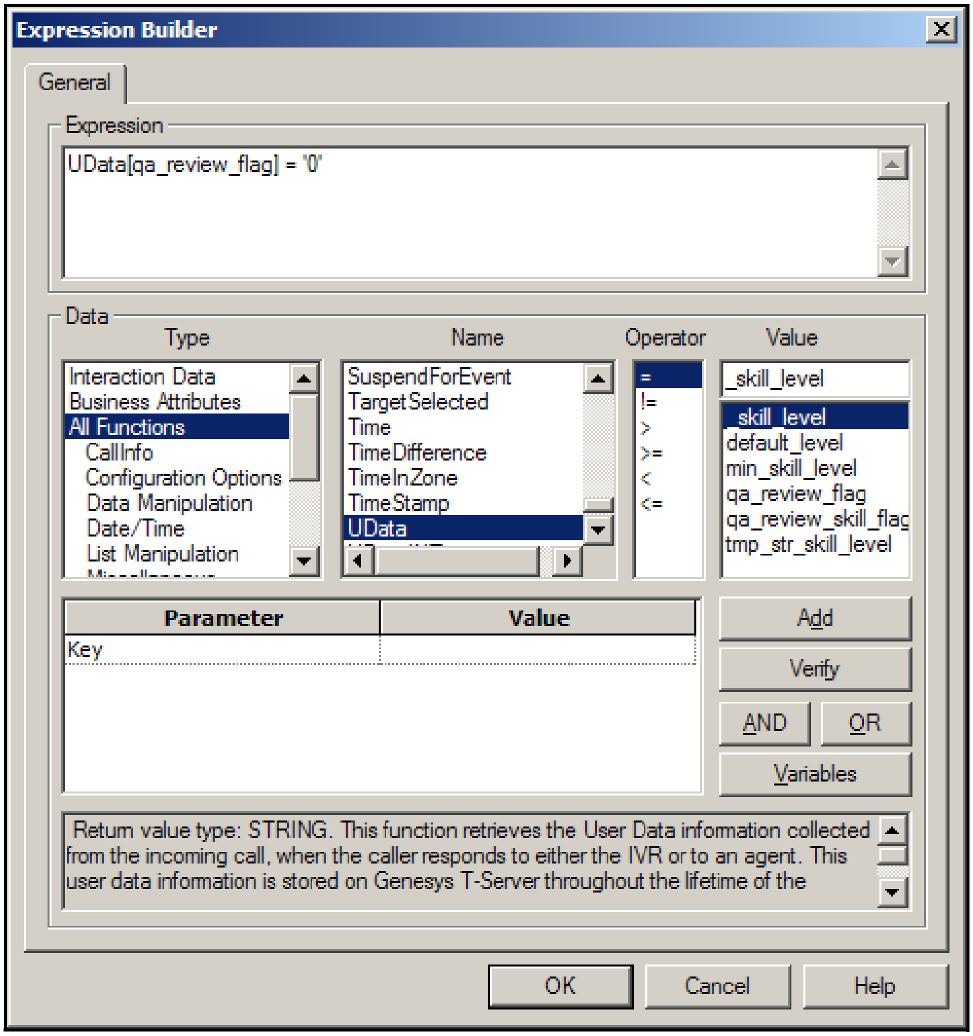
2. If the attached data is present in the variables in Figure 146, the interaction goes out the green port (see Figure 3 on page 18) of the MultiAssign object to a Generic Segmentation object, which has ports for each QA review flag value (see Figure 147).



| Segment | Expression |
|---|---|
| 1 | UData[qa_review_flag] = '0' |
| 2 | UData[qa_review_flag] = '1' |

**Figure 147: Generic Segmentation Object Properties Dialog Box**

Figure 148 shows the Expression Builder used to create Segment 1.



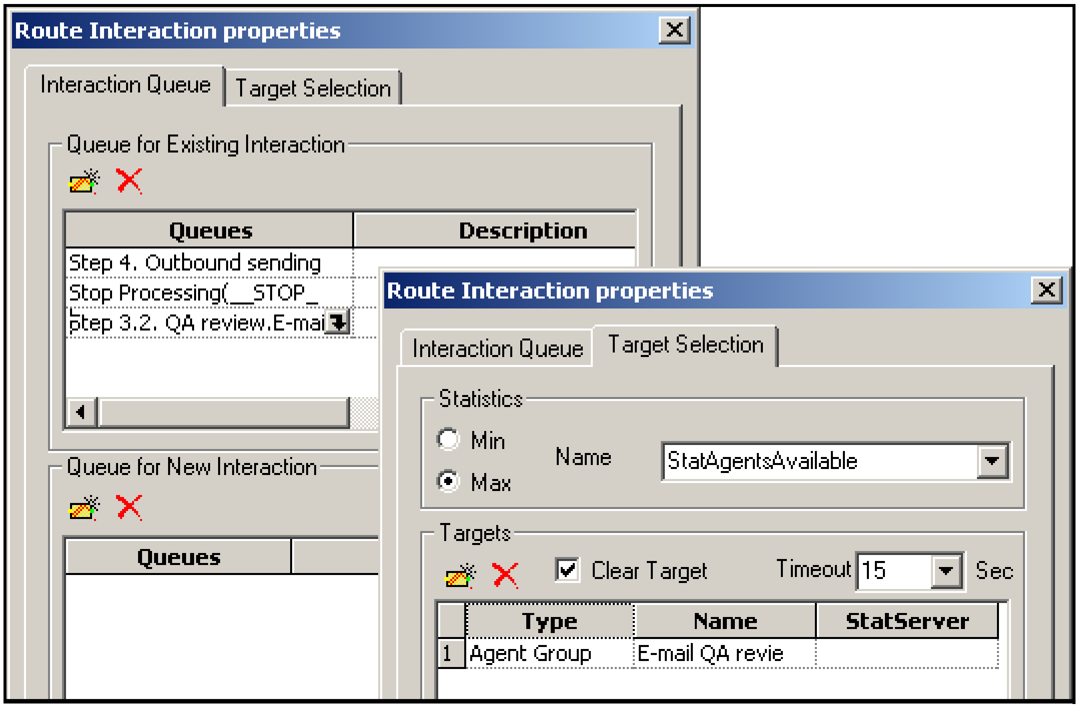
Expression: UData[qa_review_flag] = '0'

| Parameter | Value |
|---|---|
| Key | |

Return value type: STRING. This function retrieves the User Data information collected from the incoming call, when the caller responds to either the IVR or to an agent. This user data information is stored on Genesys T-Server throughout the lifetime of the

**Figure 148: Expression Builder, Segment 1**

3. Continuing with Figure 145 on page 127, if the value written to the `qa_review_flag` variable in Figure 147 equals `0`, then the e-mail response has already gone through the decision process and is ready to be reviewed by QA. The interaction goes out the green port of the Generic Segmentation object to a Route Interaction object (see Figure 149).



Route Interaction properties

Interaction Queue | Target Selection

Queue for Existing Interaction

| Queues | Description |
|---|---|
| Step 4. Outbound sending | |
| Stop Processing(__STOP_ | |
| Step 3.2. QA review.E-mai | |

Queue for New Interaction

| Queues | |
|---|---|

Route Interaction properties

Interaction Queue | Target Selection

Statistics

○ Min
● Max

Name: StatAgentsAvailable

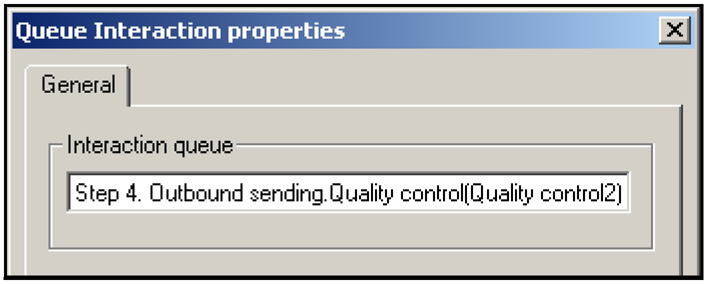
Targets

☑ Clear Target Timeout 15 Sec

| | Type | Name | StatServer |
|---|---|---|---|
| 1 | Agent Group | E-mail QA revie | |

Figure 149: Route Interaction Properties Dialog Box

4. If value contained in `qa_review_flag` in Figure 147 on page 129 equals `1`, QA has already reviewed the e-mail response and the e-mail is ready to be sent to the customer. In this case, the e-mail goes out the green port of the Generic Segmentation object to a Queue Interaction object (see Figure 150).



Queue Interaction properties

General

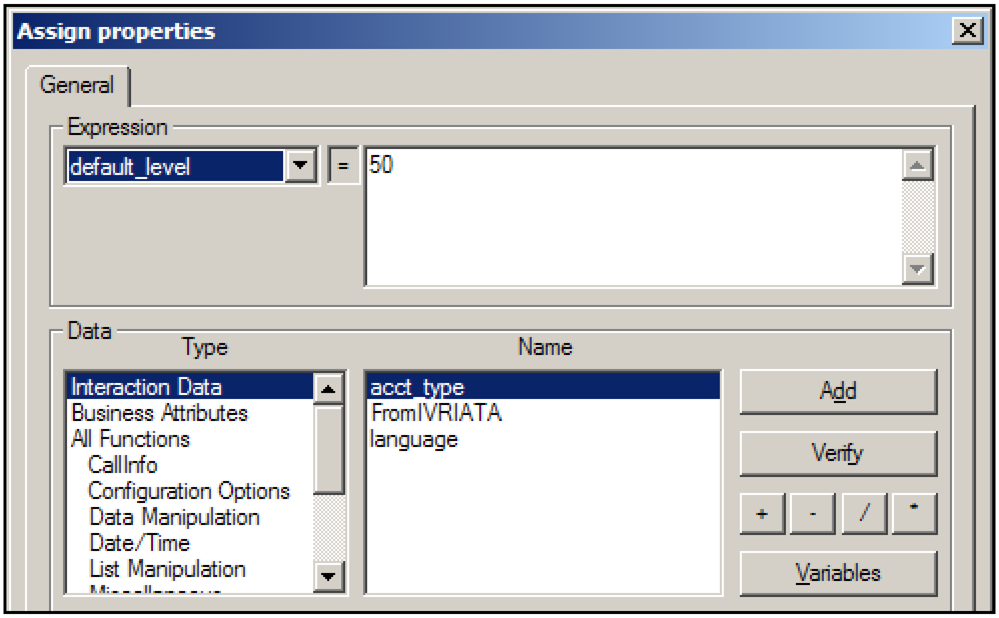
Interaction queue

Step 4. Outbound sending.Quality control(Quality control2)

Figure 150: Queue Interaction Properties Dialog Box

5. If the value contained in `qa_review_flag` Figure 147 on page 129 does not equal `0` or `1`, the e-mail response must go through the process that analyzes the skill level of the agent constructing the response. If the agent's skill level was below the minimum required, then QA must check the e-mail response before the e-mail can be sent to the customer.

6. The interaction goes out the red port of the Generic Segmentation object to an Assign object used set the default skill level (see Figure 151 on page 131).

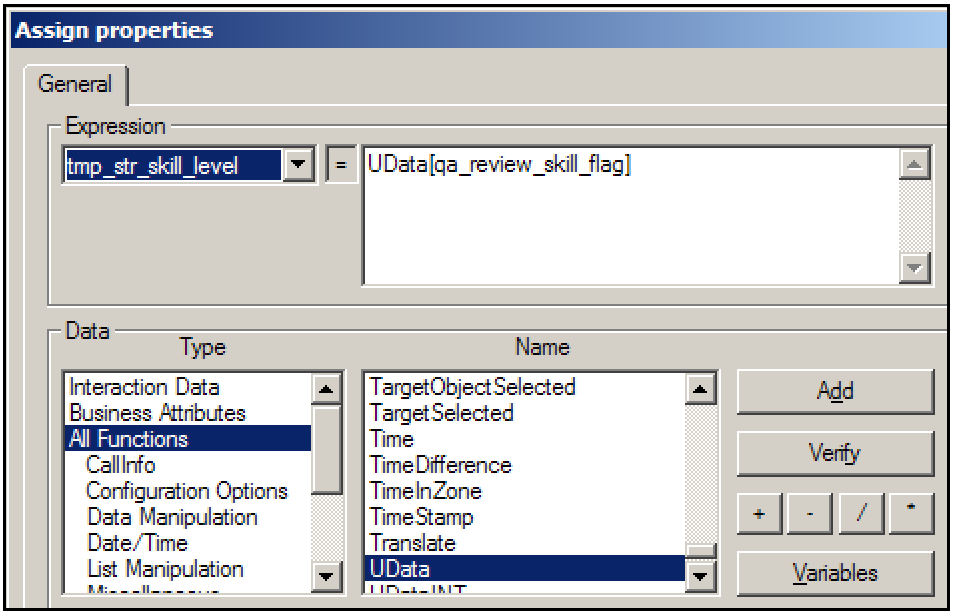

Assign properties

Expression

default_level = 50

**Figure 151: Assign Object Properties—Default Skill Level**

7. A second Assign object assigns the skill level (already contained in the interaction attached data as described in “Strategy Assumptions” on page 128)) for the agent creating the response e-mail to a variable, `tmp_str_skill_level` (see Figure 152).

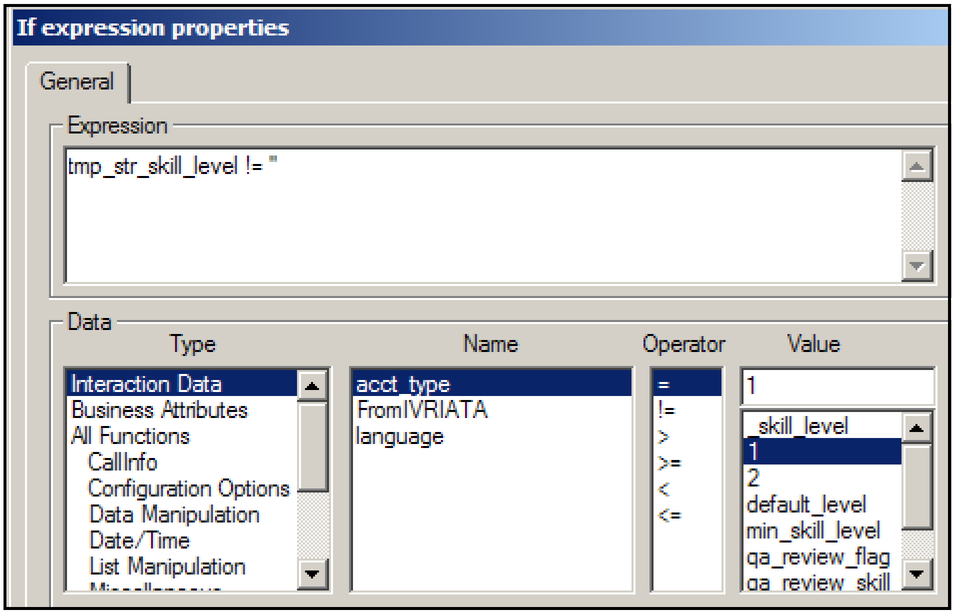

Assign properties

Expression

tmp_str_skill_level = UData[qa_review_skill_flag]

**Figure 152: Assign Object Properties—QA Review Skill Flag**

8. The interaction goes out the green port of the Assign object to an If object. It checks whether the `tmp_str_skill_level` variable contains the agent's required skill level (see Figure 153).



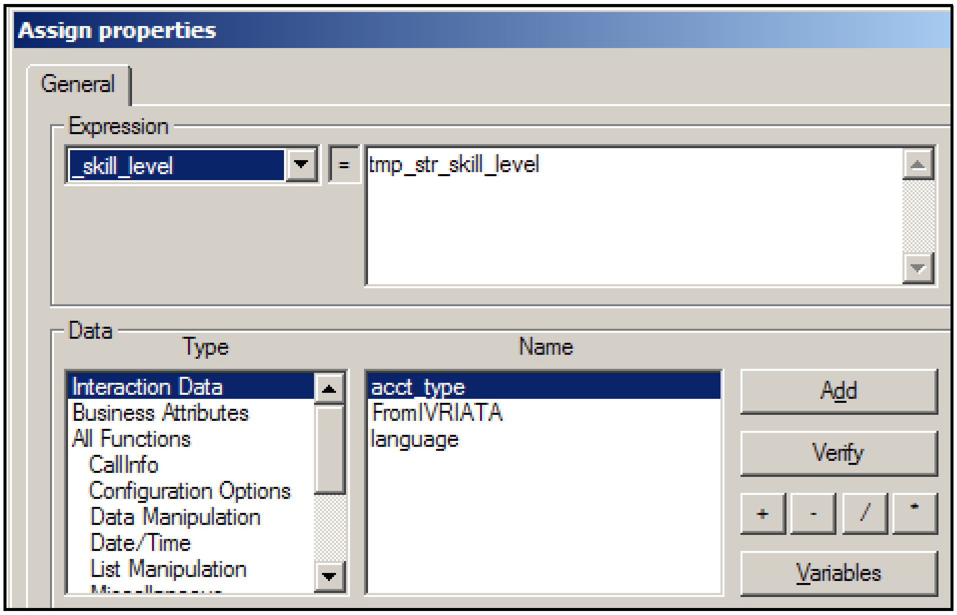
If expression properties

Expression

tmp_str_skill_level != ''

Figure 153: If Object Checking Properties—Skill Level Check

9. If the `tmp_str_skill_level` variable contains the agent's required skill level, the e-mail goes out the green port to another Assign object, which assigns the value to the `_skill_level` variable (see Figure 154).



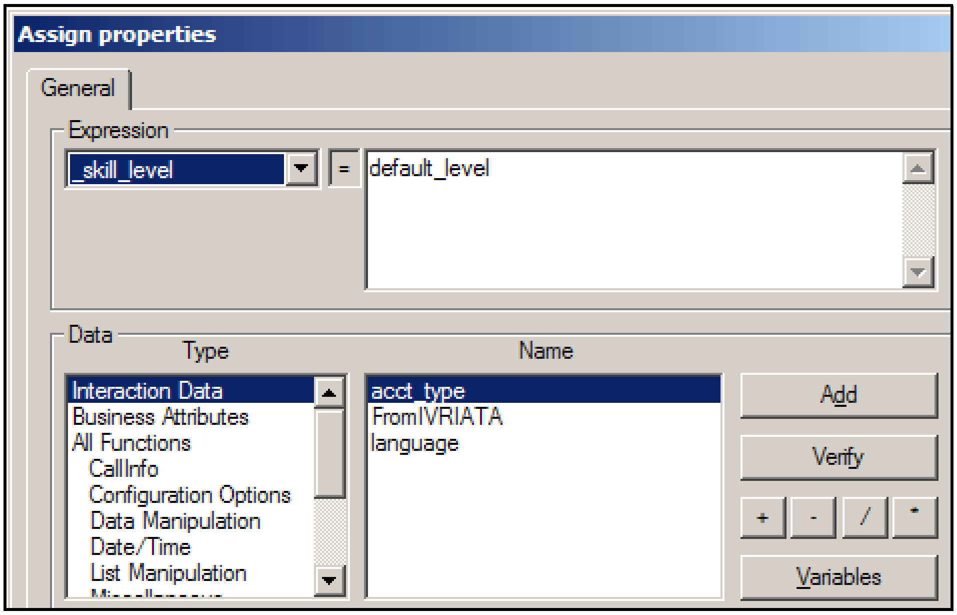
Assign properties

Expression

_skill_level = tmp_str_skill_level

Figure 154: Assign Object Properties—Skill Level Assignment

10. If the `tmp_str_skill_level` variable does not contain the agent's skill level, another Assign object assigns the default skill level (see Figure 155).

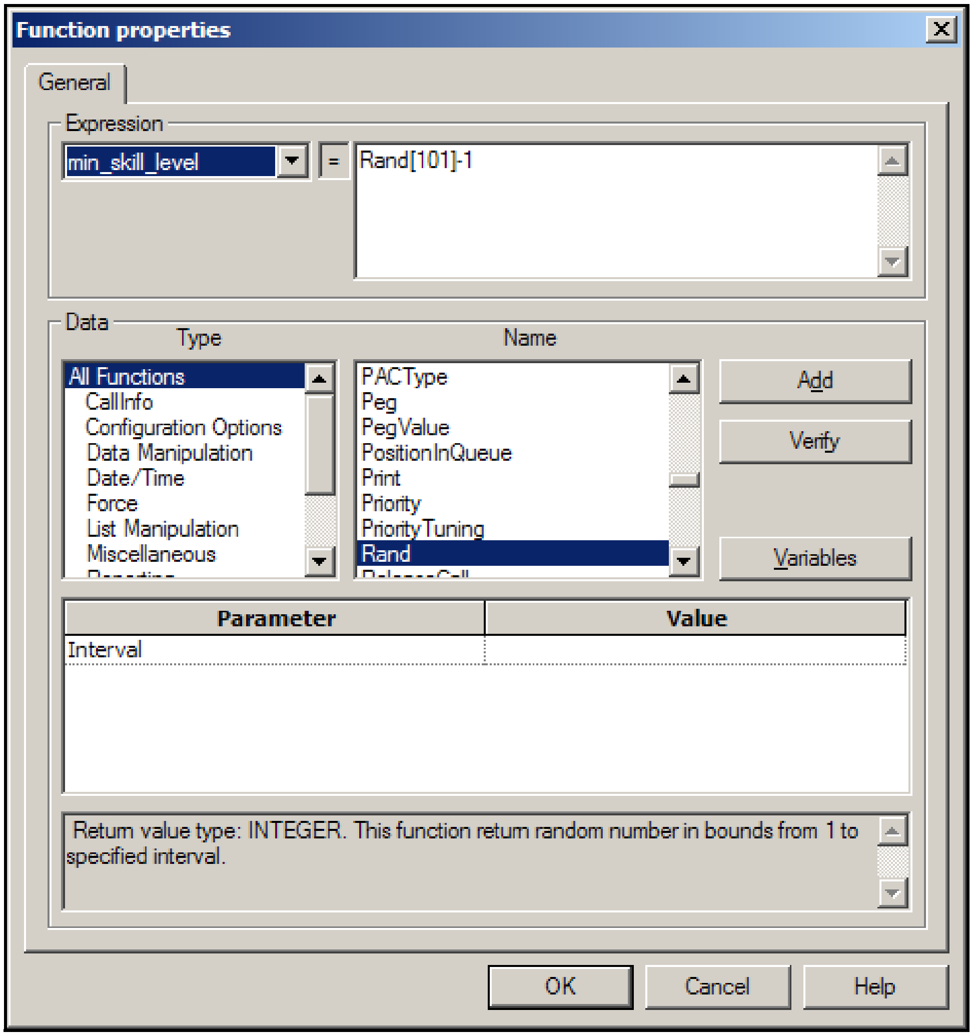


Expression

_skill_level = default_level

**Figure 155: Assign Object Properties—Default Skill Level**

11. A Function object uses the Rand function to compute the minimum skill level for the agent creating the e-mail response (see Figure 156).



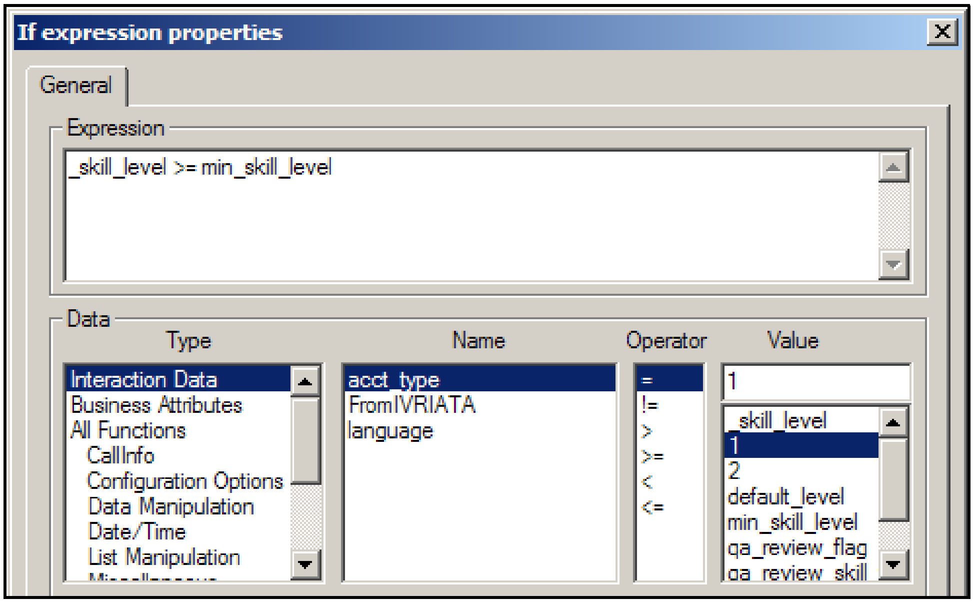
min_skill_level = Rand[101]-1

| Parameter | Value |
|---|---|
| Interval | |

Return value type: INTEGER. This function return random number in bounds from 1 to specified interval.

**Figure 156: Function Properties—Rand Function**

12. An If object reviews the responding agent's skill level to determine whether the agent's response needs to be checked or can be sent to the customer (see Figure 157).

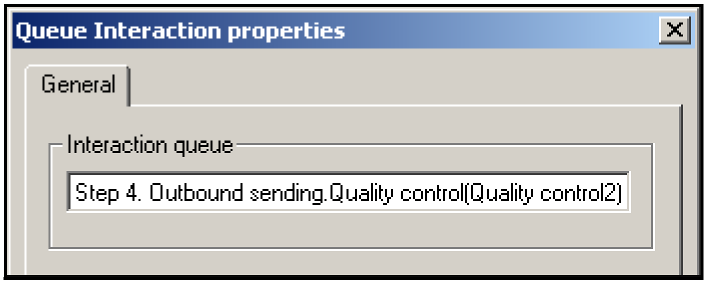


Expression: _skill_level >= min_skill_level

Value: 1

**Figure 157: If Properties—Checking for Minimum Skill Level**

13. If the expression is true, the interaction goes out through the green port of the If object to a Queue Interaction object for outbound sending quality control (see Figure 158).

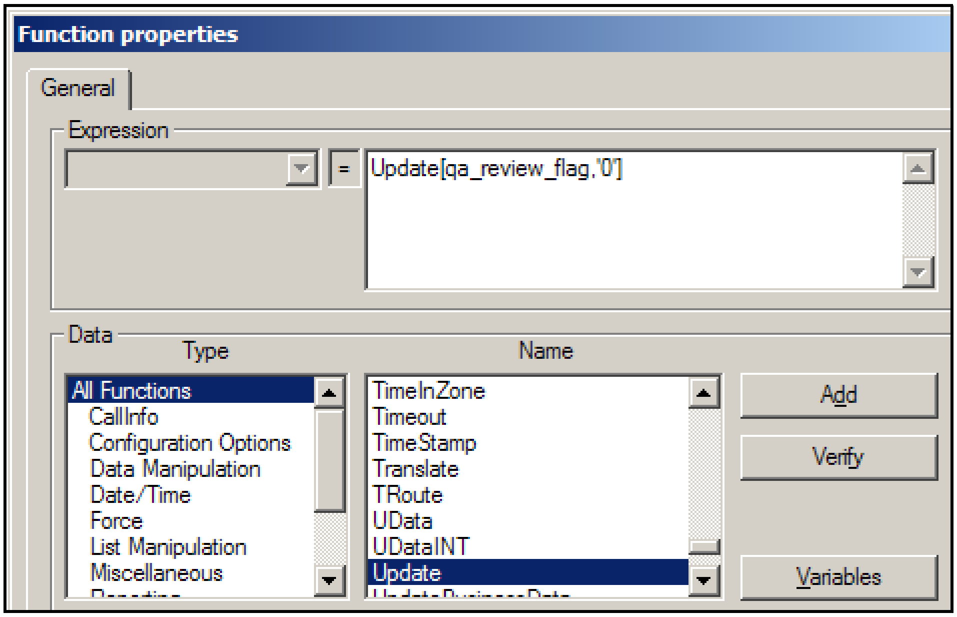


Interaction queue: Step 4. Outbound sending.Quality control(Quality control2)

**Figure 158: Queue Interaction Properties—Quality Control Queue**

14. If the expression in Figure 157 is false, the e-mail has already gone through the decision process. The responding agent's skill level is less than the minimum required so the e-mail response must be checked by QA. The interaction goes out the red port of to a Function object, which updates the QA review flag to `0` (see "Strategy Assumptions" on page 128).

    Figure 159 shows the Function Properties dialog box.



Expression: Update[qa_review_flag,'0']

Figure 159: Function Properties—Update Function

15. After updating the QA review flag to `0,` the interaction goes out through the green port of the Function object to same Route Interaction object previously used in Step 4 (see Figure 149 on page 130) to an agent group named `E-mail QA review group`.

# Skill-Based Review of Agent Response

The Interaction Workflow Samples (see page 80) contain the following routing strategy that demonstrates QA review of an e-mail response based on the responding agent's skill level:

- "Outbound E-mail 65X QA Strategy" on page 126 (see Figure 157 on page 135).

# Re-Processing E-mails That Failed QA Review

The Interaction Workflow Samples (see page 80) contain the following routing strategy that demonstrates re-processing interactions that failed QA review:

- "Outbound E-mail 65X QA Strategy" on page 126.

# Quality Control for Outbound E-mails Based On Screening

The business process `Step 4: Outbound Sending` (see page 85) contains the `Quality control` strategy, which is described in this section (see Figure 160).

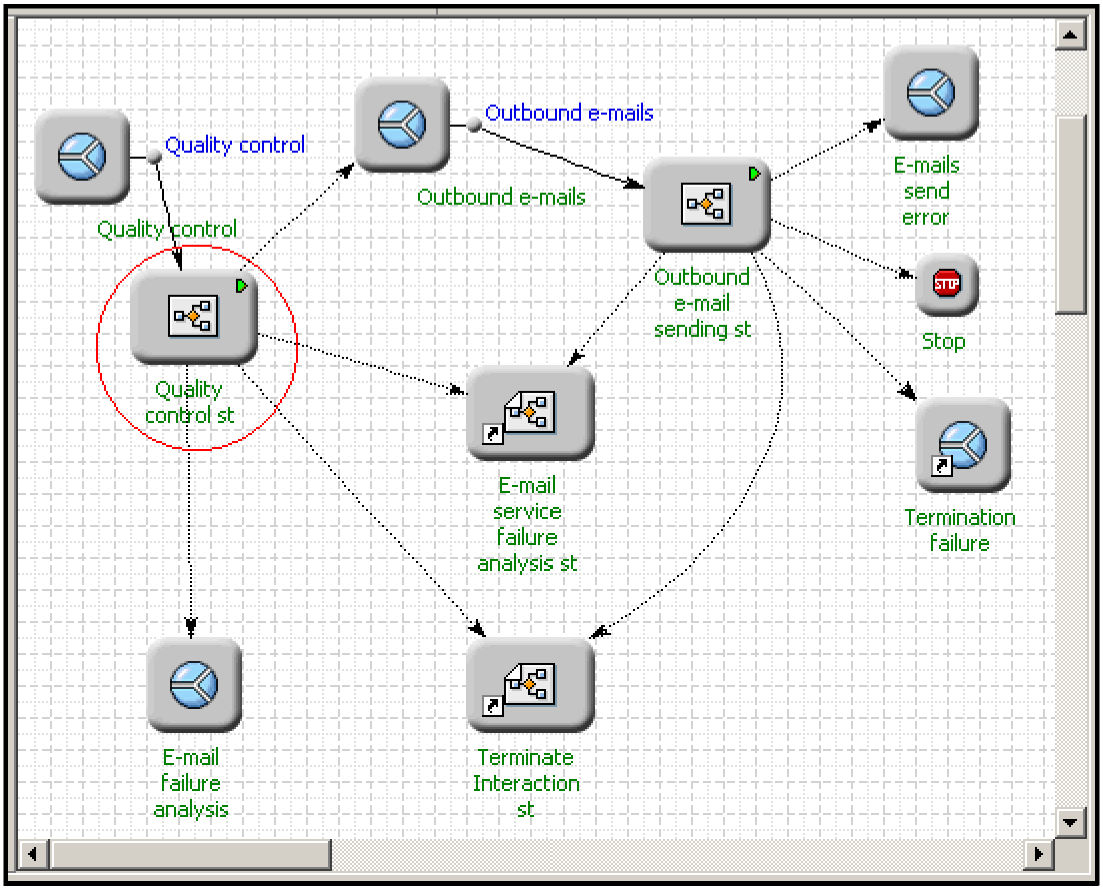


**Figure 160: Step 4: Outbound Sending Business Process**

The `Quality control` queue in Figure 160 gets interactions from the `Outbound e-mail 65x QA` strategy (see Figure 144 on page 126). A view attached to the `Quality control` queue extracts e-mails. A submitter object submits them to the `Quality control` routing strategy.

## Quality Control Strategy

This strategy screens an agent's e-mail response to determine whether it needs failure analysis because the `Outbound e-mail 65x QA strategy` placed the interaction in a queue for quality control checking (see Figure 158 on page 135).

Figure 161 shows the `Quality control` strategy.

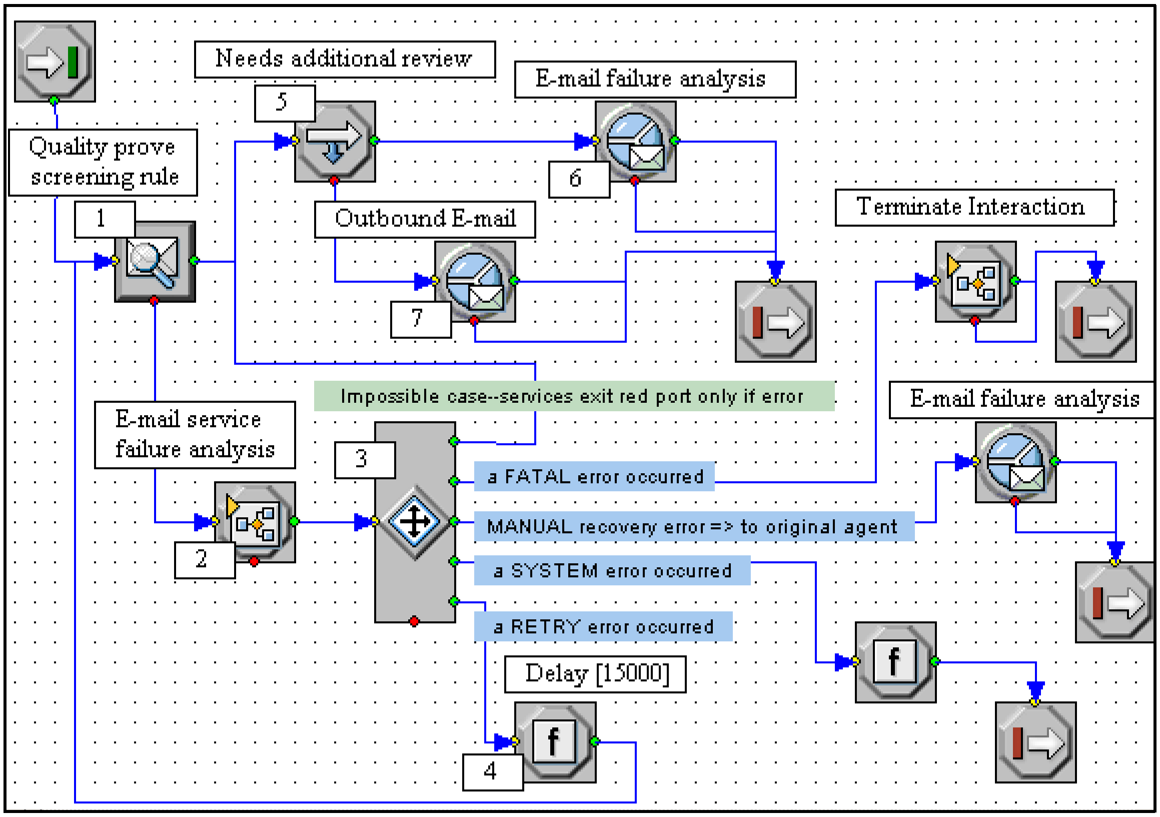
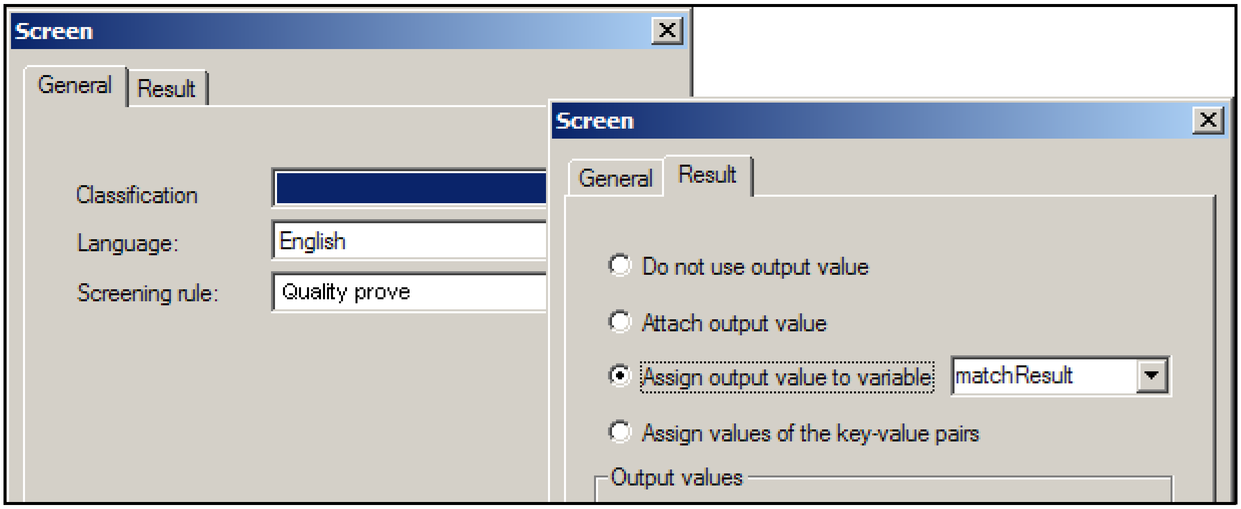


Figure 161: Quality Control Strategy

## Summary of Flow

**Note:** The IRD objects that are described this section are keyed to the numbers in Figure 161.

1. In Figure 161, a Screen object uses the `Quality prove` screening rule to analyze the e-mail content in order to determine whether failure analysis is required and writes the screening results to a variable, `matchResult` (see Figure 162).



Classification
Language: English
Screening rule: Quality prove

○ Do not use output value
○ Attach output value
● Assign output value to variable matchResult
○ Assign values of the key-value pairs

Figure 162: Screen Object Properties—Quality Prove Screening Rule

### Screening Rule Output Example

If the `matchResult` variable in Figure 162 is equal to:

```
ScreenRuleName:""|Id:""|ScreenRuleMatch:false|return:ok
```

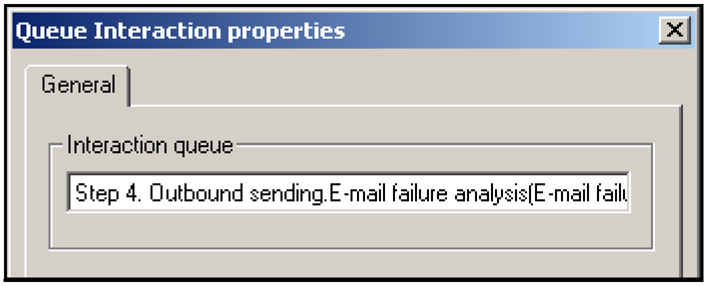
then the e-mail did *not* match the rule.

If the `matchResult` variable is equal to:

```
00001a05F5U900QW:AutoReponseAvailable|ScreenRuleName:Quality
prove|Id:00001a05F5U900QW|ScreenRuleMatch:true|return:ok
```

then the e-mail matched the rule.

2. If the e-mail did not match the rule, the interaction goes out the red error port to a subroutine, which generates an error code. The name of the subroutine is specified in a Call Subroutine object (see Figure 116 on page 104).
3. A Generic Segmentation object directs interactions based on the error code returned by the subroutine (see Figure 117 on page 104).
4. If the returned error code equals `3`, the Delay function delays execution of the strategy and the UCS notification is tried again (see Figure 118 on page 105).
5. If the e-mail matched the rule, the interaction goes out the green port to an IRD If object. Screening results contained in the `matchResult` variable in Figure 162 are used in an If expression (see Figure 119 on page 105).
6. If the expression is true, the e-mail needs failure analysis so it goes to a Queue Interaction object specifying a queue for failure analysis (see Figure 163).

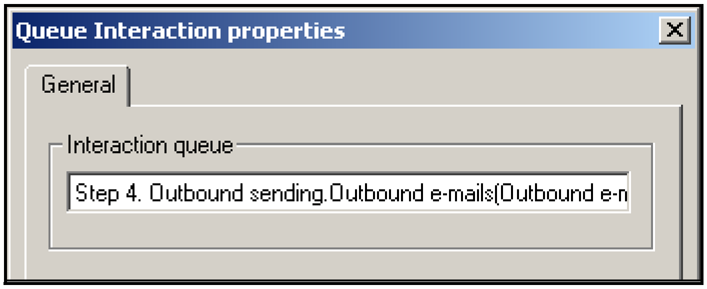


Interaction queue

Step 4. Outbound sending.E-mail failure analysis(E-mail failu

**Figure 163: Queue Interaction Object, Failure Analysis**

7. If the expression in Step 5 is false, the e-mail does not need failure analysis so it goes to a Queue Interaction object specifying an queue for outbound e-mails (see Figure 164).



Interaction queue

Step 4. Outbound sending.Outbound e-mails(Outbound e-n

**Figure 164: Queue Interaction Object, Outbound E-mails**

# Re-Processing E-mails That Failed Quality Control

The Interaction Workflow Samples (see page 80) contain the following routing strategy that demonstrates re-processing e-mails that failed quality control:

- “Outbound E-mail 65X QA Strategy” on page 126.

# Sending E-mail Responses to Customers

The business process `Step 4: Outbound Sending` (see page 85) contains the `Outbound e-mail sending` strategy described in this section (see Figure 165).

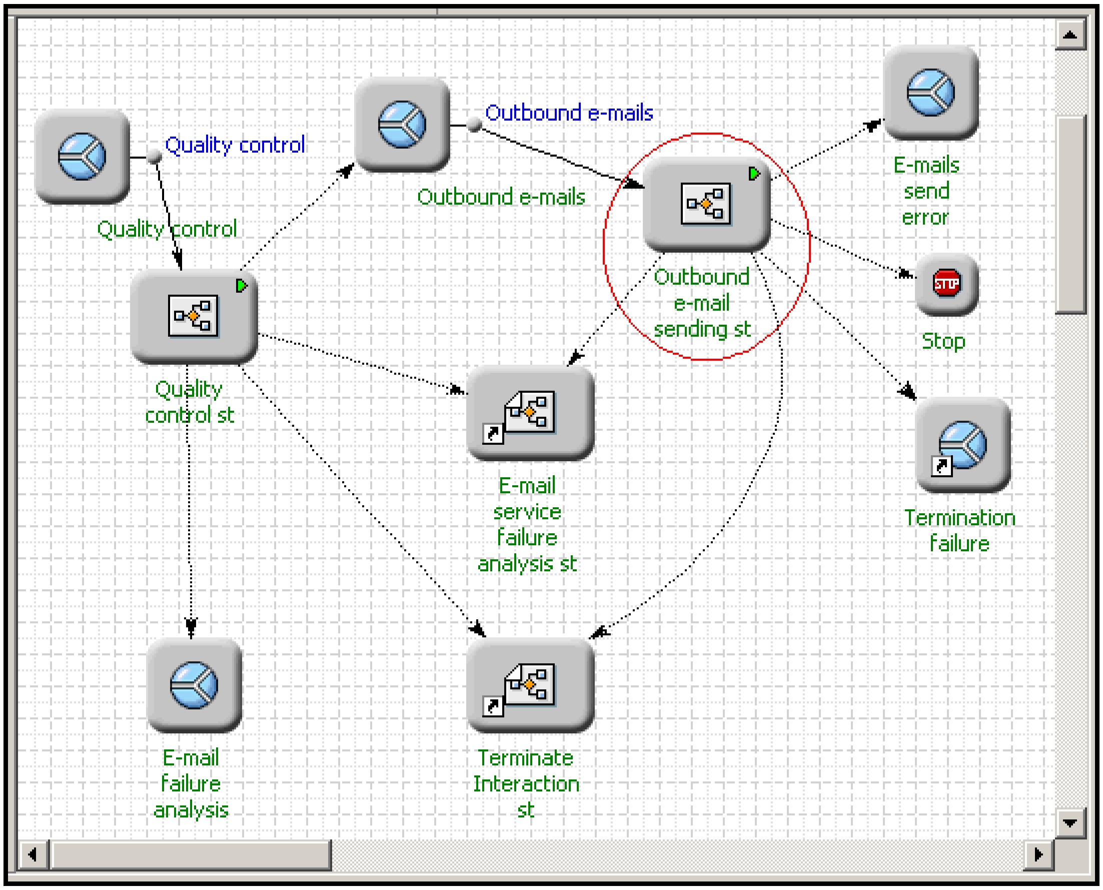


Figure 165: Step 4: Outbound Sending Business Process—Outbound E-mail Strategy

## Outbound E-mail Sending Strategy

The `Outbound e-mails` queue in Figure 165 gets interactions from the `Quality control` strategy.

A view attached to the `Outbound e-mails` queue in Figure 165 extracts e-mails. A submitter submits them to the `Outbound e-mail sending` strategy.

The purpose of the `Outbound e-mail sending` strategy is to send e-mails that have gone through quality control to E-mail Server for sending to the customer. If the e-mail cannot be sent, the strategy performs error processing.

Figure 166 shows the `Outbound e-mail sending` strategy.

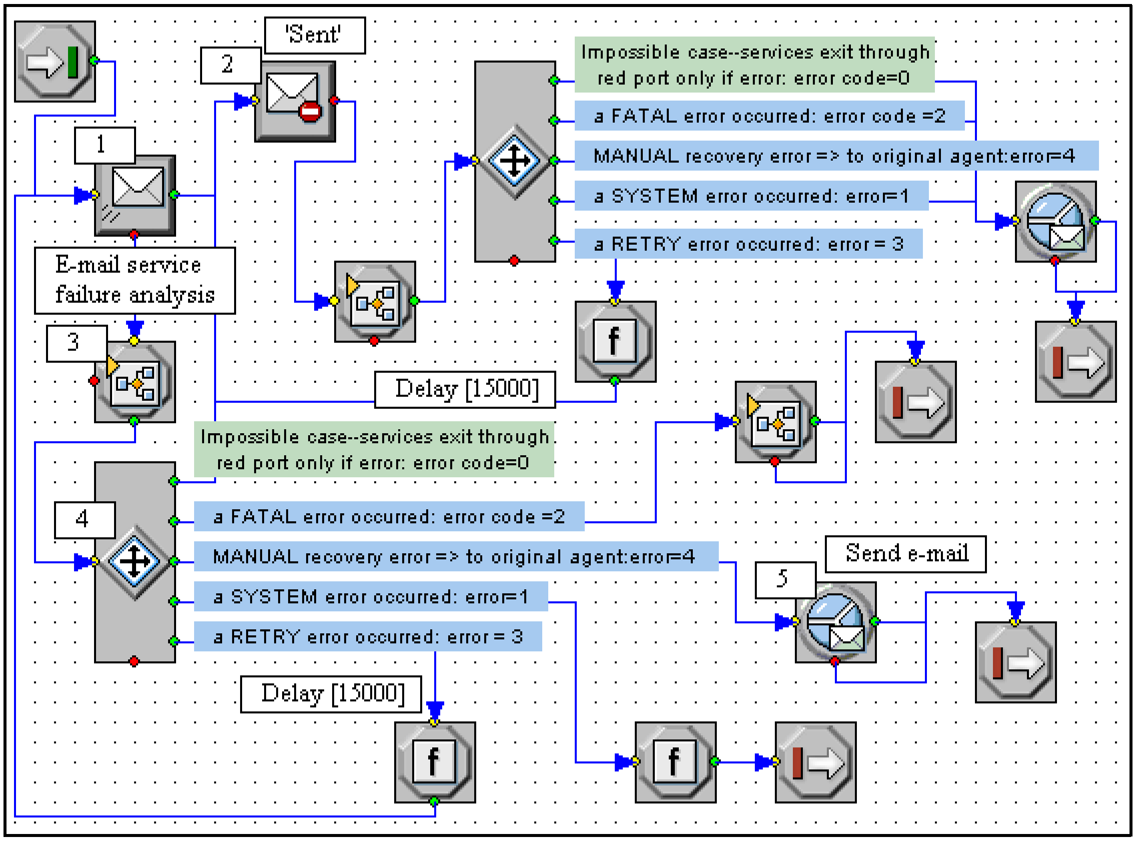

'Sent'
2
1
E-mail service failure analysis
3
Impossible case--services exit through red port only if error: error code=0
a FATAL error occurred: error code =2
MANUAL recovery error => to original agent:error=4
a SYSTEM error occurred: error=1
a RETRY error occurred: error = 3
Delay [15000]
4
Impossible case--services exit through red port only if error: error code=0
a FATAL error occurred: error code =2
MANUAL recovery error => to original agent:error=4
a SYSTEM error occurred: error=1
a RETRY error occurred: error = 3
Delay [15000]
Send e-mail
5
f

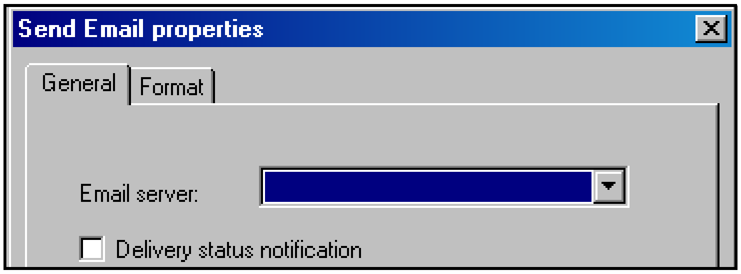
**Figure 166: Outbound E-mail Sending Strategy**

**Note:** The intent of this chapter is to demonstrate the correct use of the IRD E-mail objects. The "Summary of Flow" section is limited to the start of the strategy, which uses the Send E-Mail and Stop Interaction objects.

## Summary of Flow

**Note:** The IRD objects that are described this section are keyed to the numbers in Figure 166.

1. The first object after the Entry object in Figure 166 is a Send E-Mail object, which names the E-mail Server that will send the e-mail to the customer (see Figure 167).

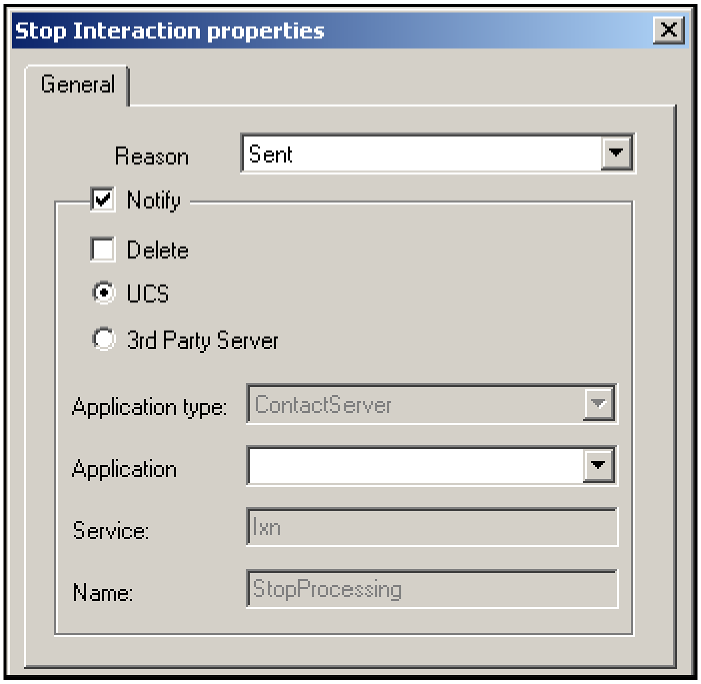


Email server:

☐ Delivery status notification

Figure 167: Send E-Mail Properties Dialog Box

2. If the e-mail is successfully sent, a Stop Interaction object halts processing and, optionally, notifies UCS (via Interaction Server) that e-mail processing is finished and supplies a reason code (see Figure 168).



Reason: Sent

☑ Notify

☐ Delete

● UCS

○ 3rd Party Server

Application type: ContactServer

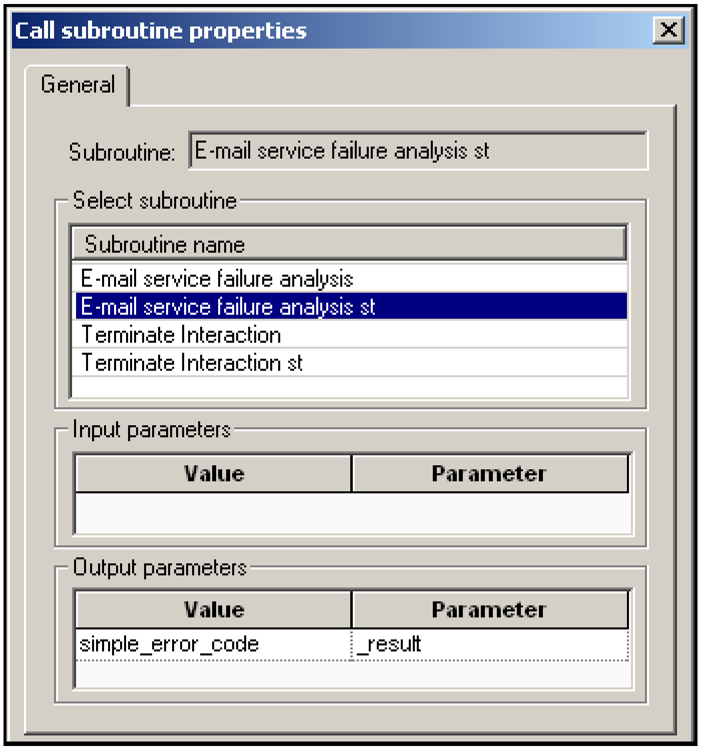
Application:

Service: Ixn

Name: StopProcessing

Figure 168: Stop Interaction Properties—Sent Reason Code

3. If the e-mail is not successfully sent, the interaction goes out the red error port to a subroutine which generates an error code. The name of the subroutine is specified in a Call Subroutine object (see Figure 169).



Subroutine: E-mail service failure analysis st

Select subroutine

| Subroutine name |
|---|
| E-mail service failure analysis |
| E-mail service failure analysis st |
| Terminate Interaction |
| Terminate Interaction st |

Input parameters

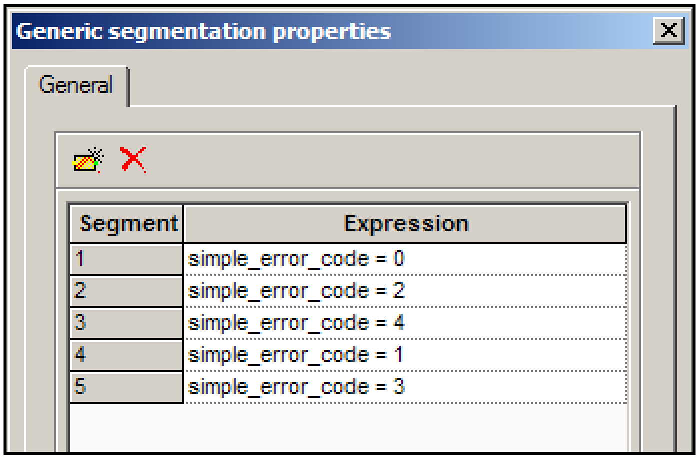
| Value | Parameter |
|---|---|

Output parameters

| Value | Parameter |
|---|---|
| simple_error_code | _result |

Figure 169: Call Subroutine Properties—E-mail Service Failure Analysis

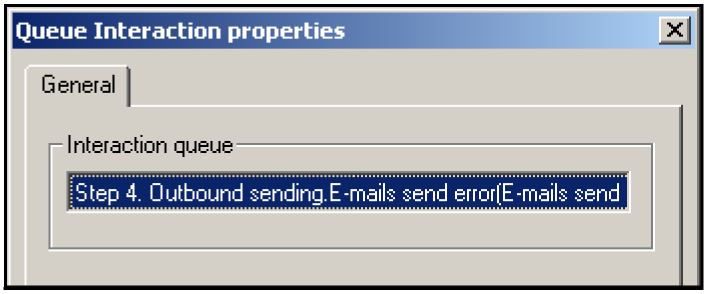
4. The interaction goes out the green port of the Call Subroutine object to a Generic Segmentation object that directs interactions based on the error code returned by the subroutine (see Figure 170).



| Segment | Expression |
|---|---|
| 1 | simple_error_code = 0 |
| 2 | simple_error_code = 2 |
| 3 | simple_error_code = 4 |
| 4 | simple_error_code = 1 |
| 5 | simple_error_code = 3 |

Figure 170: Generic Segmentation Properties Dialog Box

5. If the returned error code equals `4,` a manual recovery error occurred. The interaction goes to a Queue Interaction object that specifies a queue for e-mail send errors (see Figure 171).



Step 4. Outbound sending.E-mails send error(E-mails send

Figure 171: Queue Interaction Properties—E-mails Send Error Queue

# Re-Processing E-mails That Failed During Sending

The Interaction Workflow Samples (see page 80) contain the following routing strategy, which demonstrates reprocessing of e-mails that failed quality control:

- "Outbound E-mail Sending Strategy" on page 141. See Figure 171.

# Chat Processing

The `ABC Simple Chat BP` business process contains the following strategies:

1. `Chat inbound` strategy, which routes interactions to the target agent group.
2. `Chat strategy - create chat transcript e-mail`, which uses the IRD Chat Transcript object to request E-mail Server to create an e-mail with the attached chat transcript.
3. `Chat strategy - send transcript e-mail`, which sends the e-mail with the attached chat transcript to the customer.

Figure 172 shows the `ABC Simple Chat BP` business process, and the three strategies included in it, as it appears in the Interaction Design window.

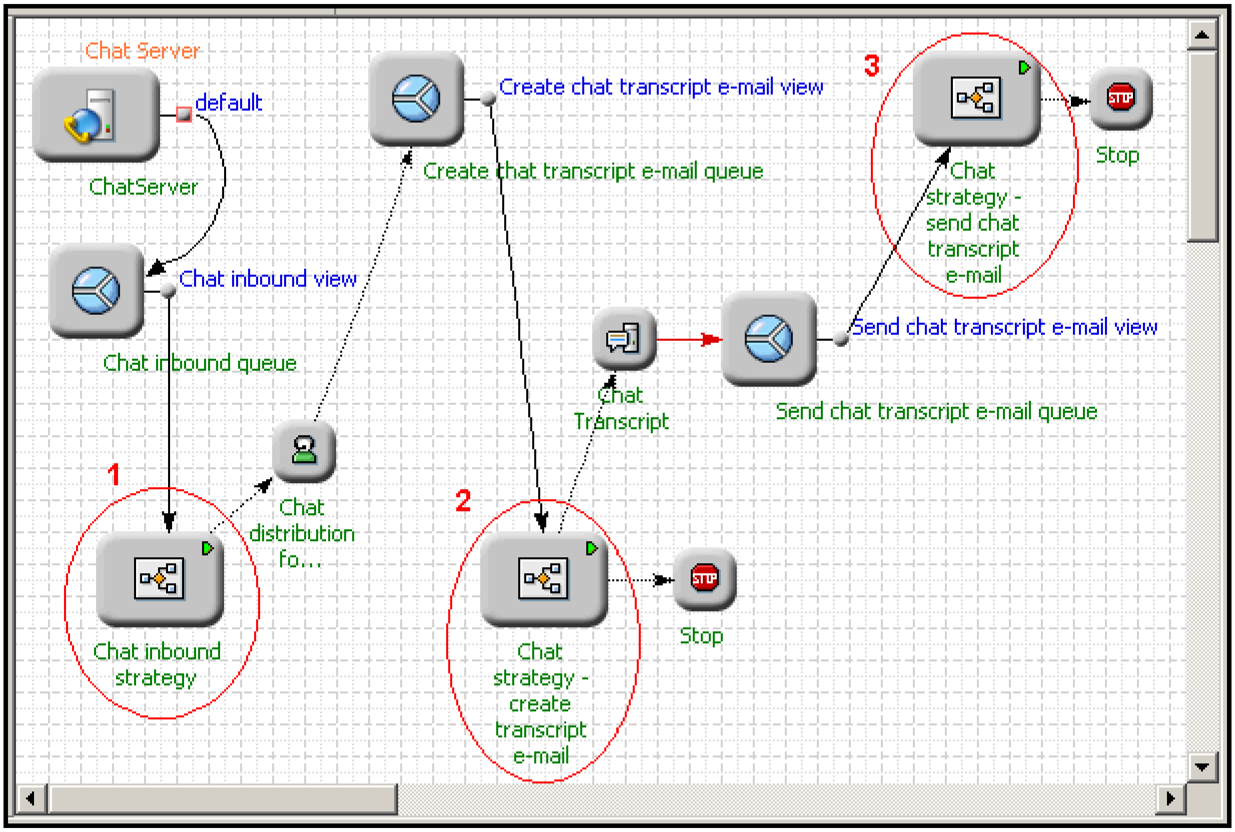

Chat Server
default
ChatServer
Create chat transcript e-mail view
Create chat transcript e-mail queue
3
Chat strategy - send chat transcript e-mail
Stop
Chat inbound view
Chat inbound queue
Send chat transcript e-mail view
Send chat transcript e-mail queue
Chat Transcript
1
Chat distribution fo...
2
Chat inbound strategy
Chat strategy - create transcript e-mail
Stop

Figure 172: ABC Simple Chat BP Business Process

## Chat Inbound Strategy

A view attached to `Chat inbound` queue in Figure 172 extracts chat interactions. A submitter object submits interactions to the `Chat inbound` strategy. The purpose of the strategy is to route interactions to the target agent group. Figure 173 shows the `Chat inbound strategy`.

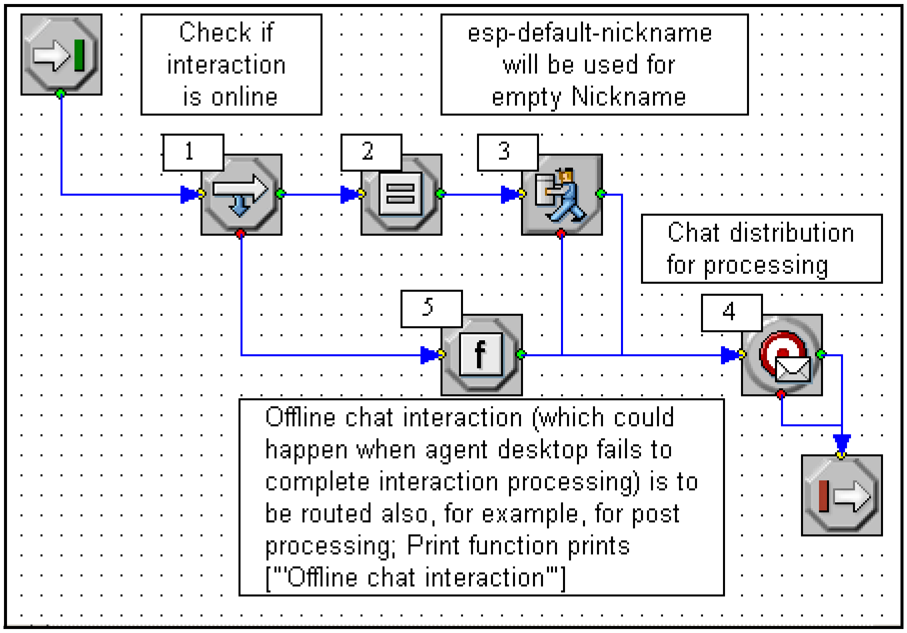
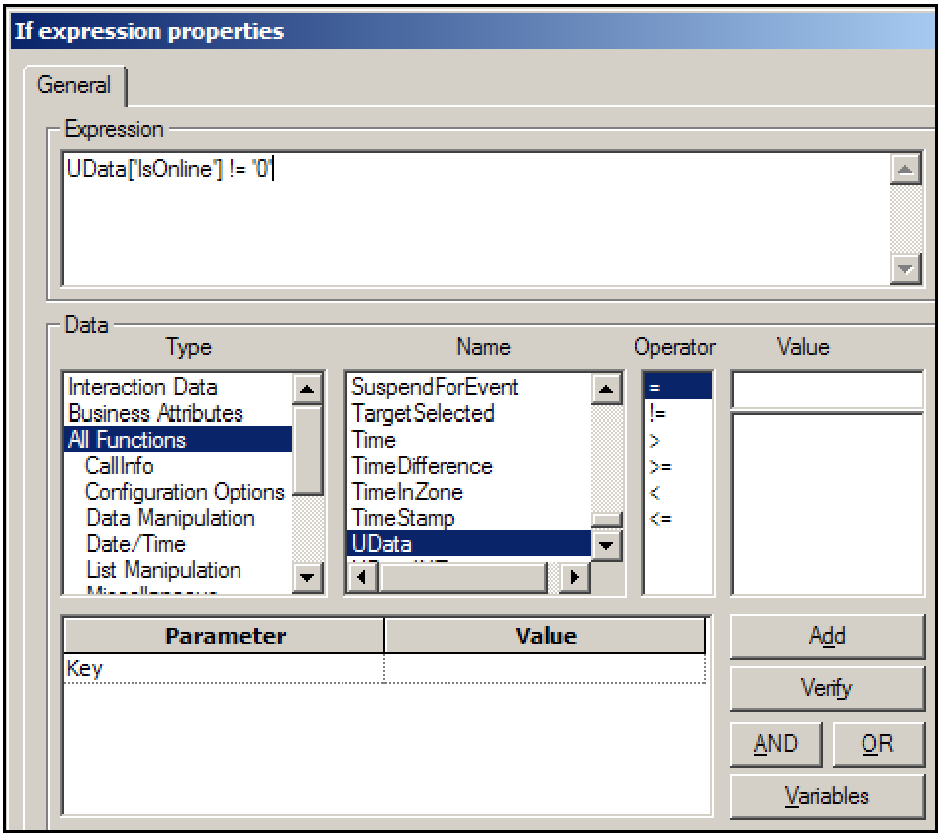

Check if interaction is online
esp-default-nickname will be used for empty Nickname
1
2
3
Chat distribution for processing
5
4
Offline chat interaction (which could happen when agent desktop fails to complete interaction processing) is to be routed also, for example, for post processing; Print function prints ["'Offline chat interaction'"]

**Figure 173: Chat Inbound Strategy**

## Summary of Flow

**Note:** The IRD objects that are described this section are keyed to the numbers in Figure 173.

1. An If object creates an expression used to determine whether an interaction has an `OnLine` attribute, which indicates an ongoing chat session that is still alive. The placement of an If object in this strategy is only to demonstrate the presence of such an attribute (see Figure 174).



If expression properties

General

Expression

UData['IsOnline'] != '0'

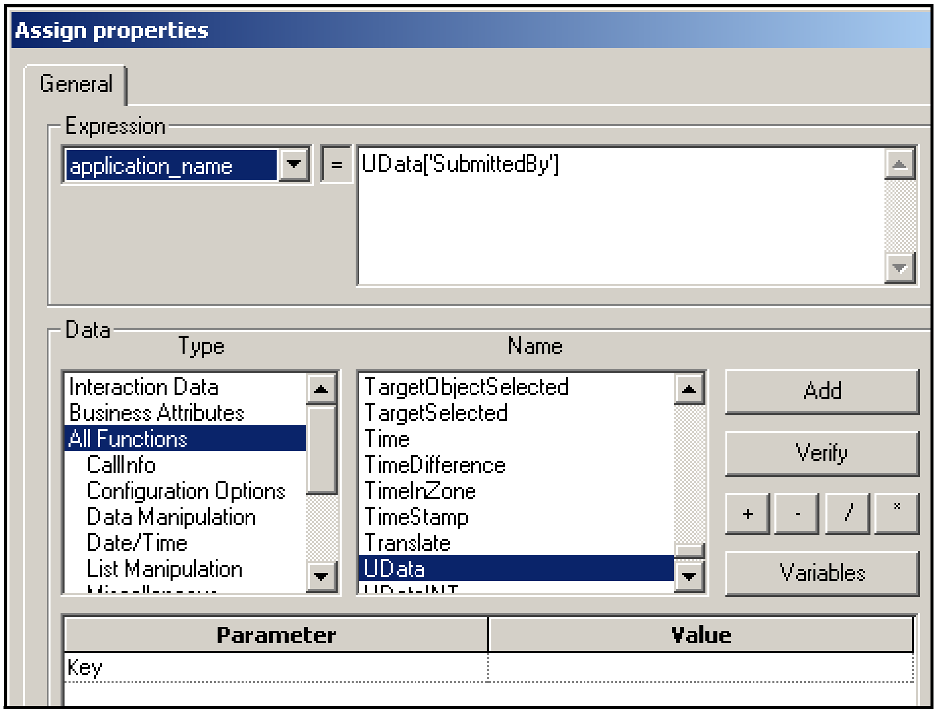
Data

| Parameter | Value |
|---|---|
| Key | |

Figure 174: If Object Function Properties—UData Function

**Note:** The UData function extracts a value from the `IsOnline` variable (see “Example EventRouteRequest Message” on page 88).

2. If the interaction is online, it proceeds to an Assign object, where information about the customer who submitted the chat request is attached to the interaction (see Figure 175).

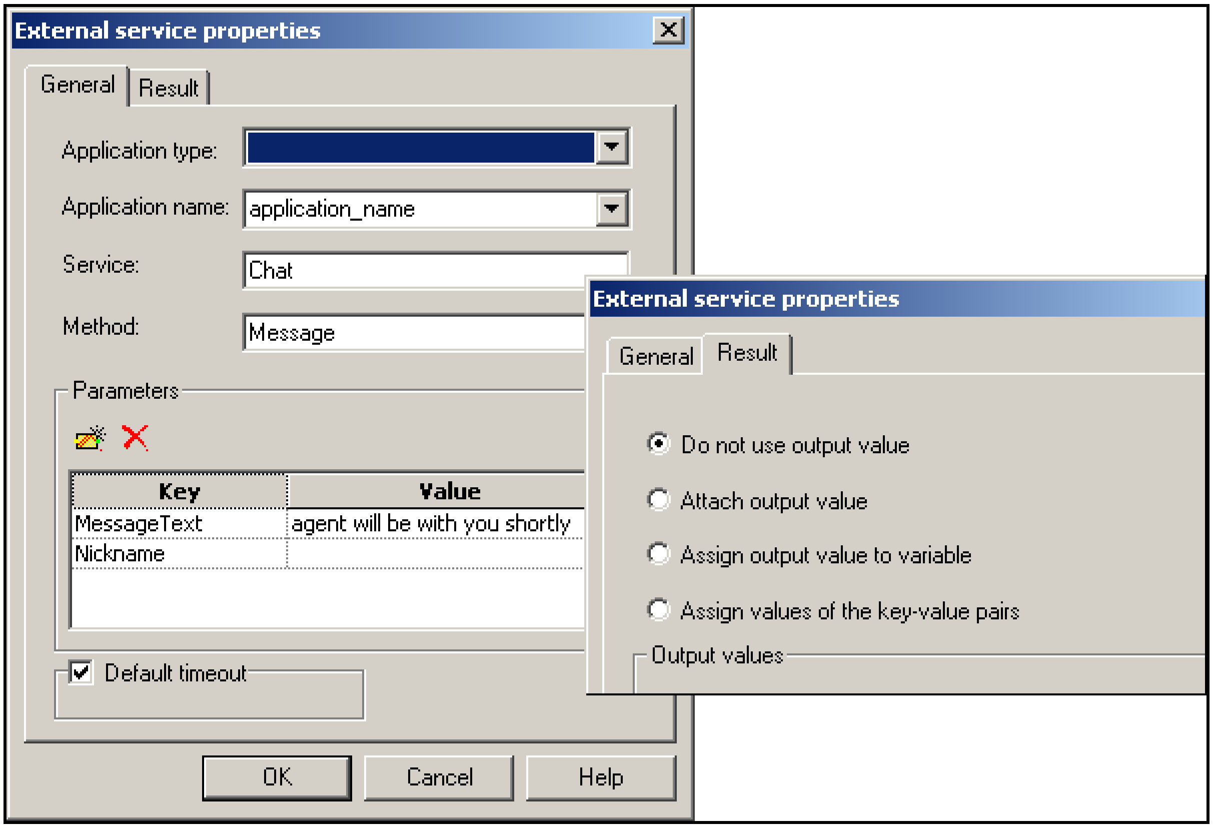


Expression: application_name = UData['SubmittedBy']

| Parameter | Value |
|---|---|
| Key | |

Figure 175: Assign Object Attaches User Data on Chat Request Submitter

3. Next, an External Service object sends a message to the customer who requested the chat session, letting her know that an agent will join the session shortly (see Figure 176).



Application type:

Application name: application_name

Service: Chat

Method: Message

Parameters

| Key | Value |
|---|---|
| MessageText | agent will be with you shortly |
| Nickname | |

☑ Default timeout

● Do not use output value

○ Attach output value

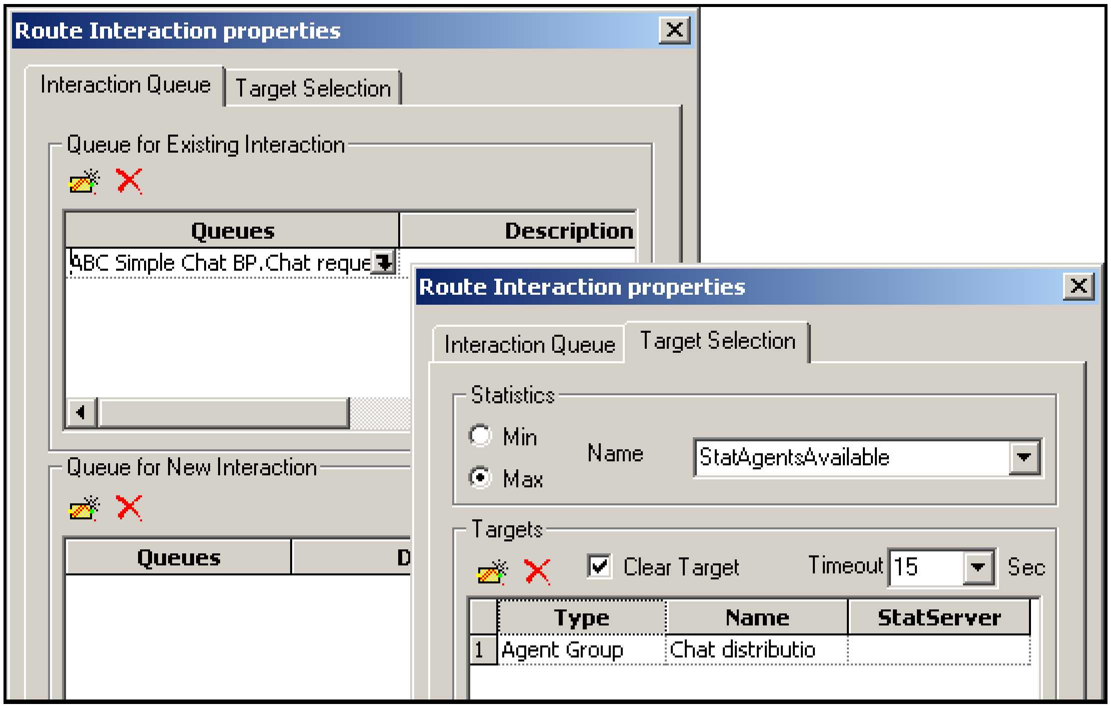
○ Assign output value to variable

○ Assign values of the key-value pairs

Output values

**Figure 176: External Service Object Sending Message to Chat Customer**

4. Whether the expression in Figure 174 on page 148 is true or false, the interaction is routed to the `Chat Distribution for processing` agent group using a Route Interaction object. When the agent is done with the interaction, the agent places it, along with the chat transcript, in the `Create chat transcript e-mail` queue (see Figure 177 and also Figure 172 on page 146).



Route Interaction properties

Interaction Queue | Target Selection

Queue for Existing Interaction

| Queues | Description |
|---|---|
| ABC Simple Chat BP.Chat reque | |

Queue for New Interaction

| Queues | D |
|---|---|

Route Interaction properties

Interaction Queue | Target Selection

Statistics

○ Min ● Max Name StatAgentsAvailable

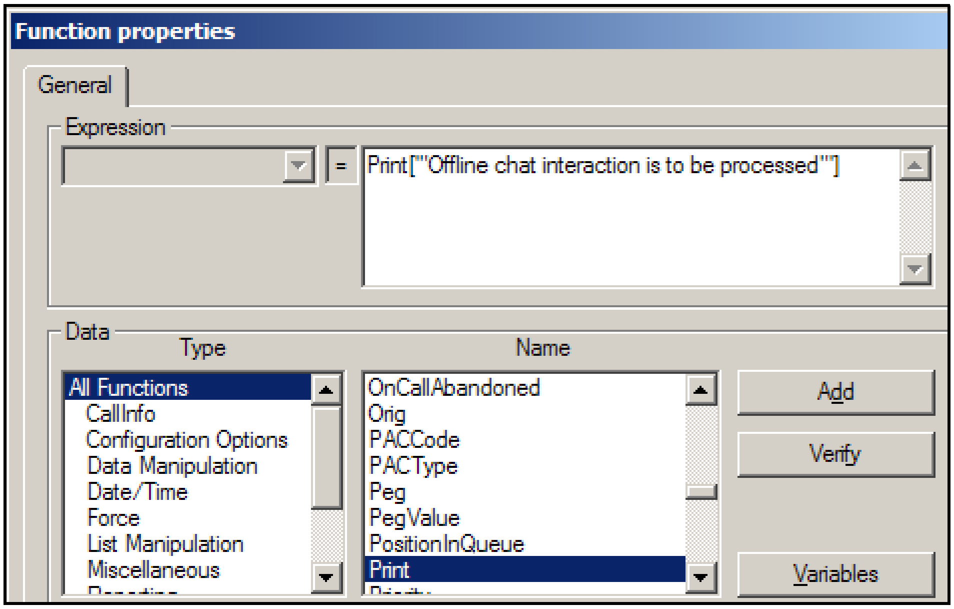
Targets

☑ Clear Target Timeout 15 Sec

| | Type | Name | StatServer |
|---|---|---|---|
| 1 | Agent Group | Chat distributio | |

**Figure 177: Route Interaction Properties Dialog Box**

5. If an interaction is determined to be offline, a Function object (see Figure 178) prints an informational message stating that the interaction is offline. The interaction is then routed to the `Chat Distribution` agent group for processing (see Figure 177).



Function properties

General

Expression

= Print["'Offline chat interaction is to be processed'"]

Data

| Type | Name |
|---|---|
| All Functions | OnCallAbandoned |
| CallInfo | Orig |
| Configuration Options | PACCode |
| Data Manipulation | PACType |
| Date/Time | Peg |
| Force | PegValue |
| List Manipulation | PositionInQueue |
| Miscellaneous | Print |

**Figure 178: Function Properties Dialog Box, Print Function**

# Chat Strategy - Create Transcript E-mail

After the agent completes the chat session and places the chat transcript in the `Create chat transcript e-mail` queue (see Figure 172 on page 146), the next strategy, `Chat Strategy - Create Transcript E-mail`, continues the processing.

The purpose of the `Chat Strategy - Create Transcript E-mail` strategy is to request E-mail Server to create an e-mail with the attached chat transcript and place it in a queue for sending to the customer. If this action cannot be completed, the strategy prints a message to that effect in the error log.

A view (called `Create chat transcript e-mail view`) attached to the `Create chat transcript e-mail` queue shown in Figure 172 on page 146 takes interactions from the queue. A submitter object submits them to `Chat Strategy - Create Transcript E-mail`.

Figure 179 shows `Chat Strategy - Create Transcript E-mail` in the Routing Design window.

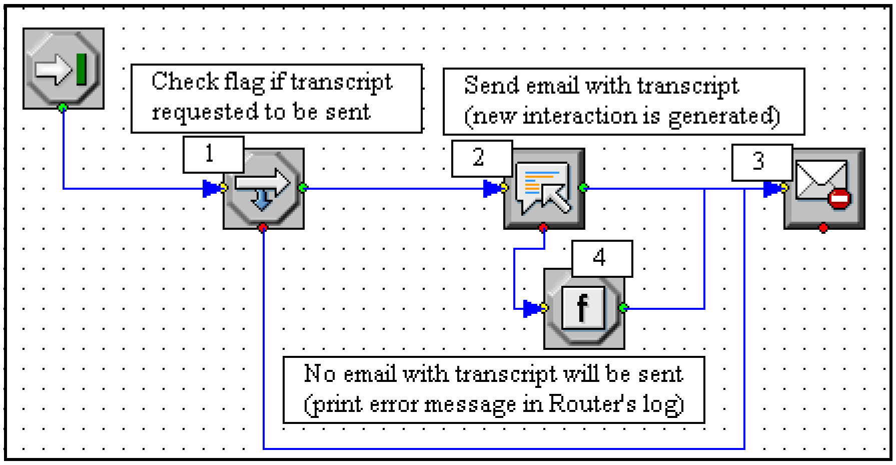

Check flag if transcript requested to be sent
Send email with transcript (new interaction is generated)
1
2
3
4
No email with transcript will be sent (print error message in Router's log)

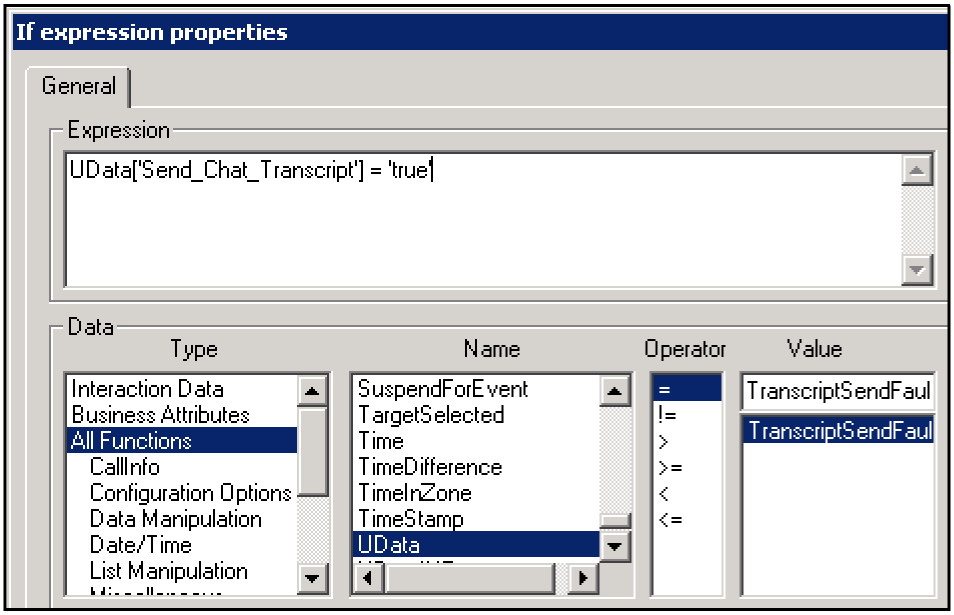
**Figure 179: Chat Strategy - Create Transcript E-mail**

## Summary of Flow

**Note:** The IRD objects that are described this section are keyed to the numbers in Figure 179.

1. An If object verifies whether the attached data `Send_Chat_Transcript` is true.

**Note:** This key/value pair can be modified at the agent desktop by selecting or clearing the `Send Chat Transcript` check box (see Figure 180).



If expression properties

General

Expression

UData['Send_Chat_Transcript'] = 'true'

Data

Type: Interaction Data, Business Attributes, All Functions (selected), CallInfo, Configuration Options, Data Manipulation, Date/Time, List Manipulation

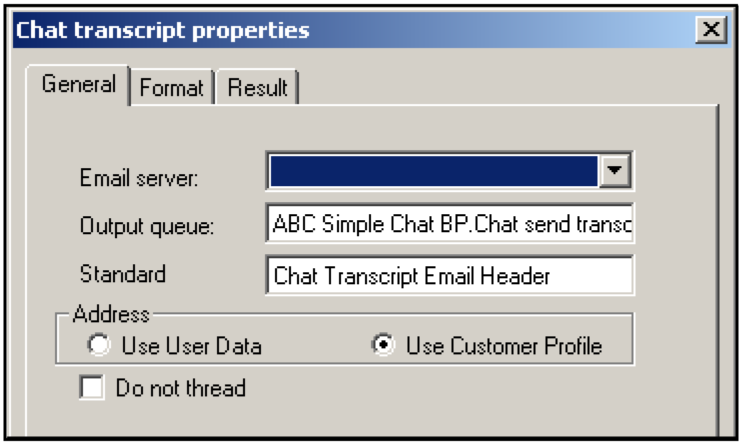
Name: SuspendForEvent, TargetSelected, Time, TimeDifference, TimeInZone, TimeStamp, UData (selected)

Operator: = (selected), !=, >, >=, <, <=

Value: TranscriptSendFaul, TranscriptSendFaul (selected)

**Figure 180: If Object For Sending Chat Transcript**

2. If true, a Chat Transcript object requests E-mail Server to create an e-mail (using text from the Knowledge Manager standard response library) indicating that the customer’s chat transcript is attached. The object also requests E-mail Server to place the e-mail in an output queue (see Figure 181).



Chat transcript properties

General | Format | Result

Email server:

Output queue: ABC Simple Chat BP.Chat send transc

Standard: Chat Transcript Email Header

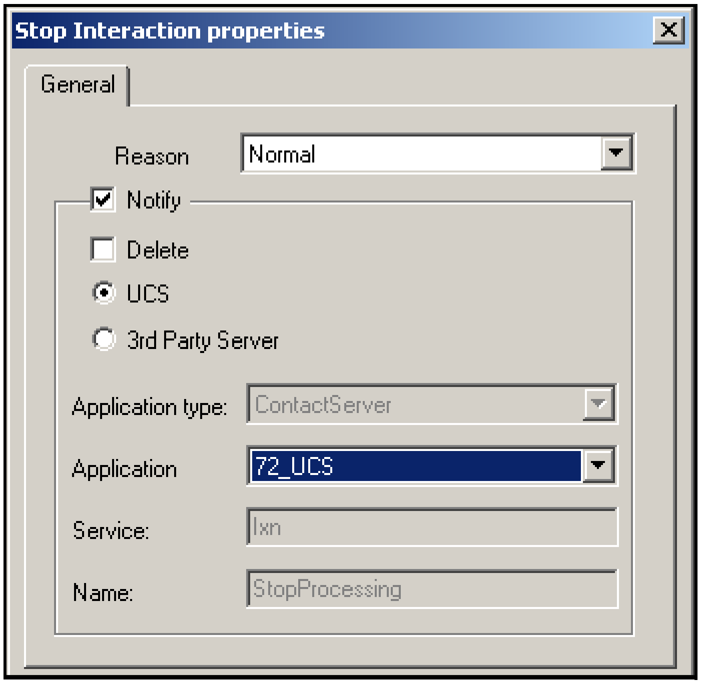
Address: ○ Use User Data ● Use Customer Profile

□ Do not thread

**Figure 181: Chat Transcript Object**

In Figure 181, the `Use Customer Profile` button is selected, indicating that E-mail Server should use the customer’s profile in the UCS Database to supply the customer’s e-mail address. `Do not thread` is cleared, indicating that E-mail Server should not thread this e-mail to another e-mail.

3. If the e-mail is successfully created, the interaction goes out the green port to a Stop Interaction object. This object stops interaction processing and notifies UCS through Interaction Server that chat processing is finished. It also supplies a reason code indicating why processing stopped (see Figure 182).



Reason Normal

☑ Notify

□ Delete

● UCS

○ 3rd Party Server

Application type: ContactServer

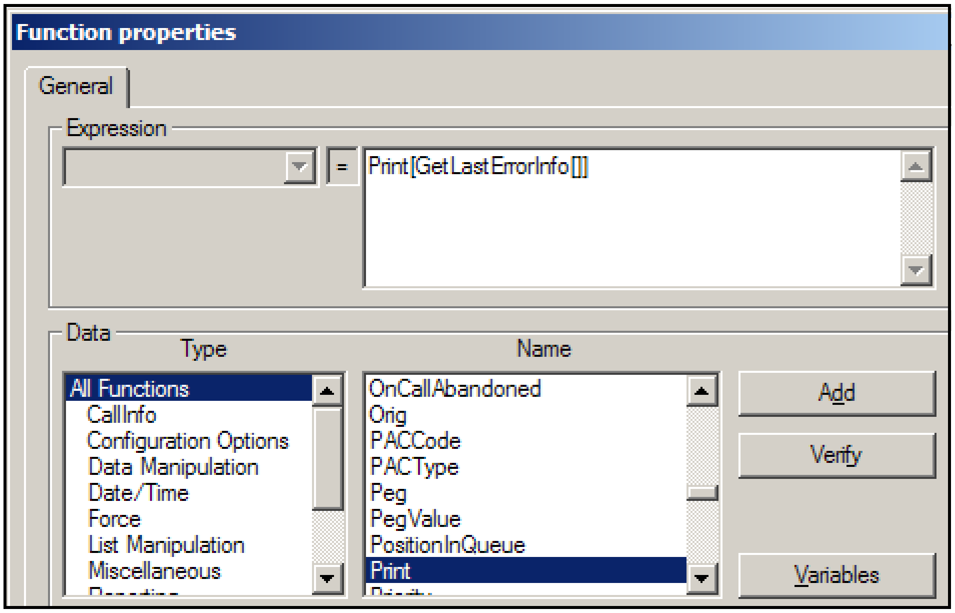
Application 72_UCS

Service: Ixn

Name: StopProcessing

**Figure 182: Stop Interaction Properties—Normal Reason Code**

4. If the e-mail is not successfully created, the interaction goes out the red error port to a Function object that prints a message in the error log (see Figure 183).



Function properties

Expression: = Print[GetLastErrorInfo[]]

Data Type: All Functions

Name: Print

**Figure 183: Function Object, Print Function**

## Chat Strategy - Send Chat Transcript E-mail

Once E-mail Server creates the e-mail with the attached chat transcript and places it in a queue, the e-mail must be sent to the customer. This is the purpose of the Chat Strategy - send chat transcript e-mail strategy in the ABC Simple Chat BP business process (shown in Figure 172 on page 146).

A view attached to the Send chat transcript e-mail queue takes interactions from the queue. A submitter object submits interactions to the Send chat transcript e-mail strategy.

Figure 184 shows Chat Strategy - Send chat transcript e-mail.

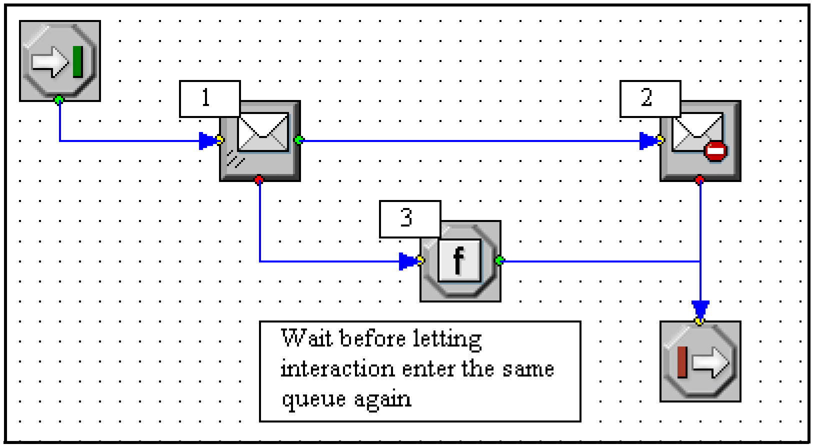
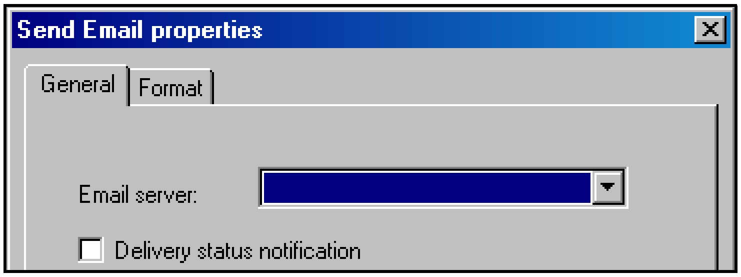


**Figure 184: Chat Strategy - Send Chat Transcript E-mail**

## Summary of Flow

**Note:** The IRD objects that are described this section are keyed to the numbers in Figure 184.

1. A Send E-Mail object requests an E-mail Server to send the e-mail with the attached chat transcript to the customer (see Figure 185).

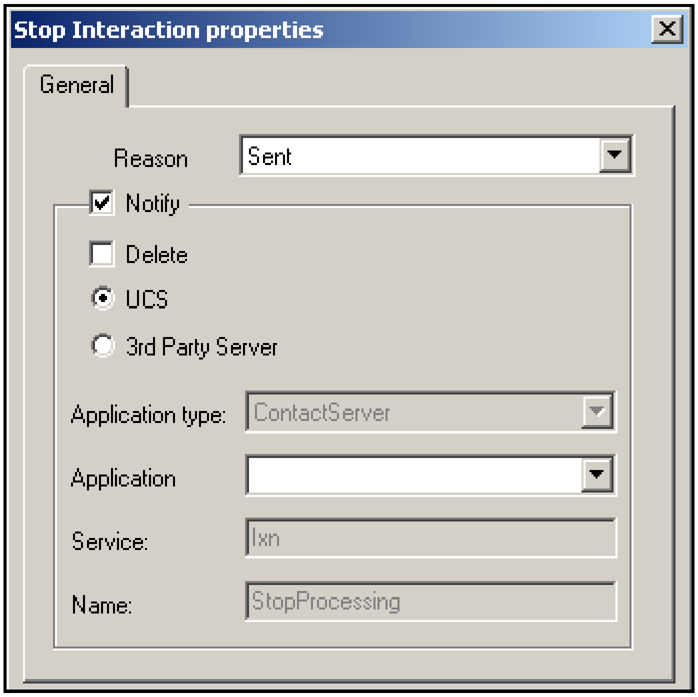


Email server:

□ Delivery status notification

**Figure 185: Send E-mail Properties Dialog Box**

2. If the e-mail is successfully sent, a Stop Interaction object stops processing and notifies UCS (via Interaction Server) that e-mail processing is finished. It also supplies a reason code (see Figure 186).



Reason: Sent

☑ Notify

□ Delete

● UCS

○ 3rd Party Server

Application type: ContactServer

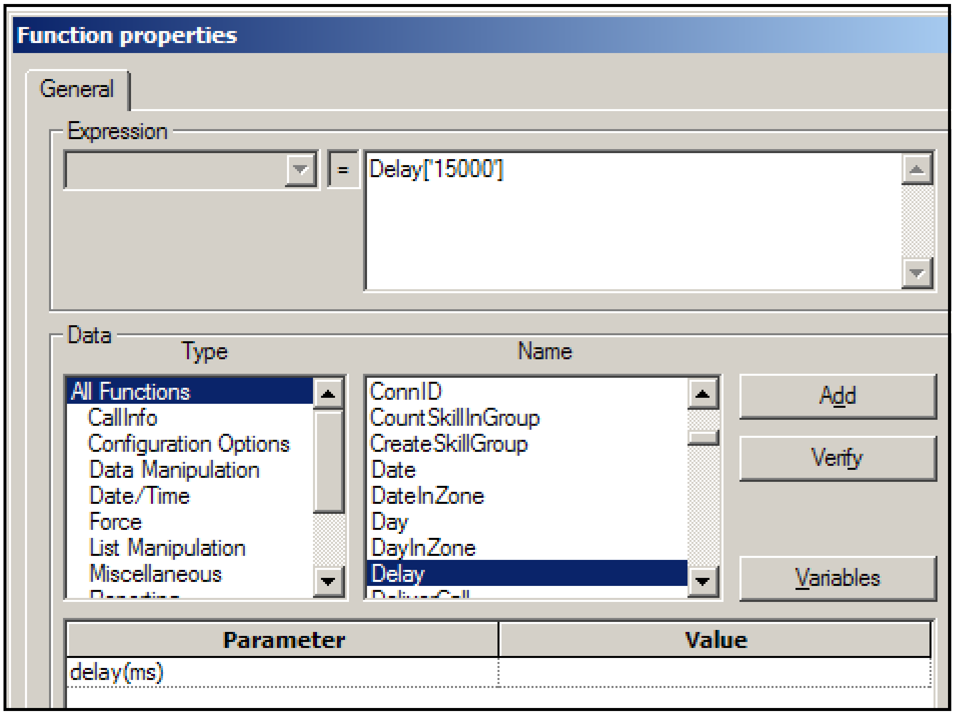
Application

Service: Ixn

Name: StopProcessing

**Figure 186: Stop Interaction Properties—Sent Reason Code**

3. If the e-mail is not successfully sent, the interaction goes out the red error port to a Function object that uses the Delay function to have a specified number of seconds elapse before the interaction re-enters the same send queue (see Figure 187).



Expression: Delay['15000']

| Parameter | Value |
|---|---|
| delay(ms) | |

**Figure 187: Function Object, Delay Function**

# MMS Processing

The `ABC Simple MMS` business process contains one strategy: `MMS Inbound Processing`. This strategy routes the MMS interaction to the appropriate agent. The `ABC Simple MMS` business process contains one queue: `MMS Inbound`.

Figure 188 shows the ABC Simple MMS business process and the strategy that is included in it, as it appears in the Interaction Design window.

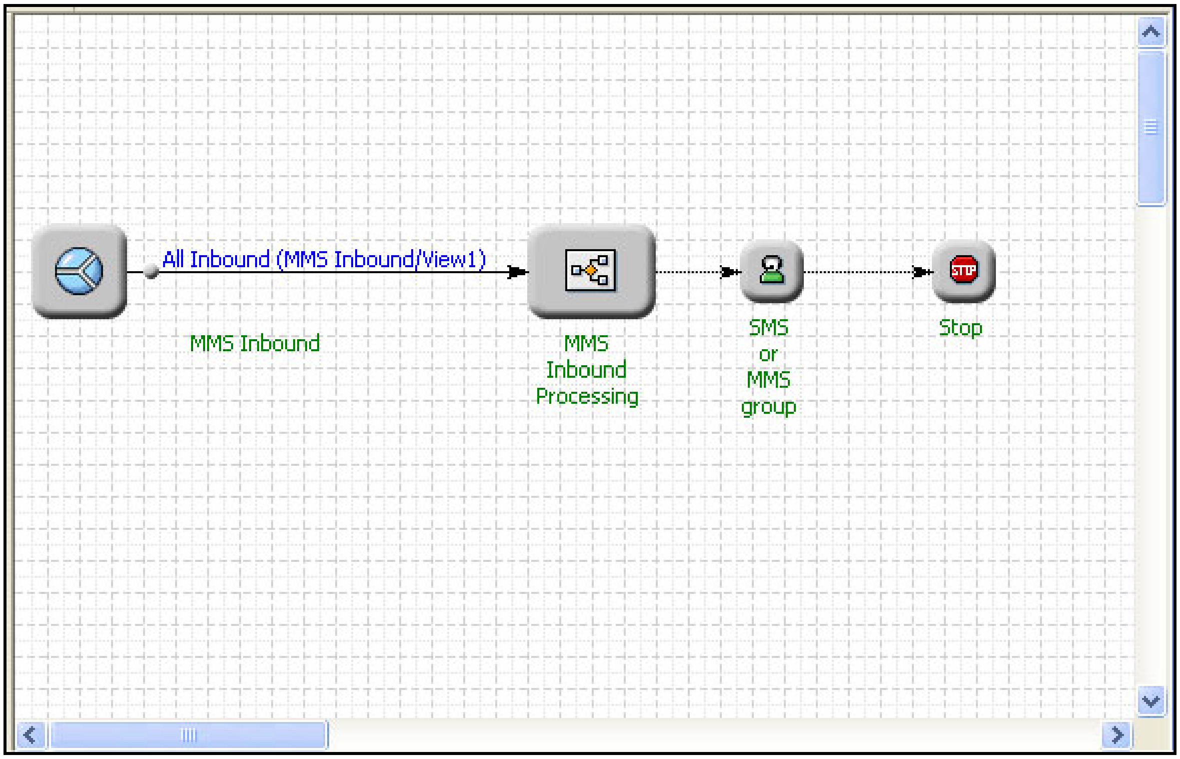


**Figure 188: ABC Simple MMS Business Process**

## MMS Inbound Processing Strategy

The purpose of this strategy is to route inbound MMS interactions to an appropriate agent. Figure 189 shows the MMS Inbound Processing strategy.

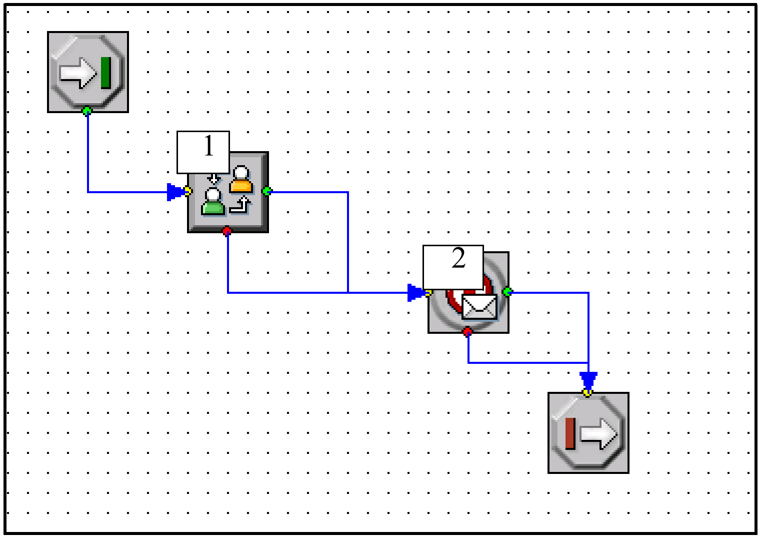
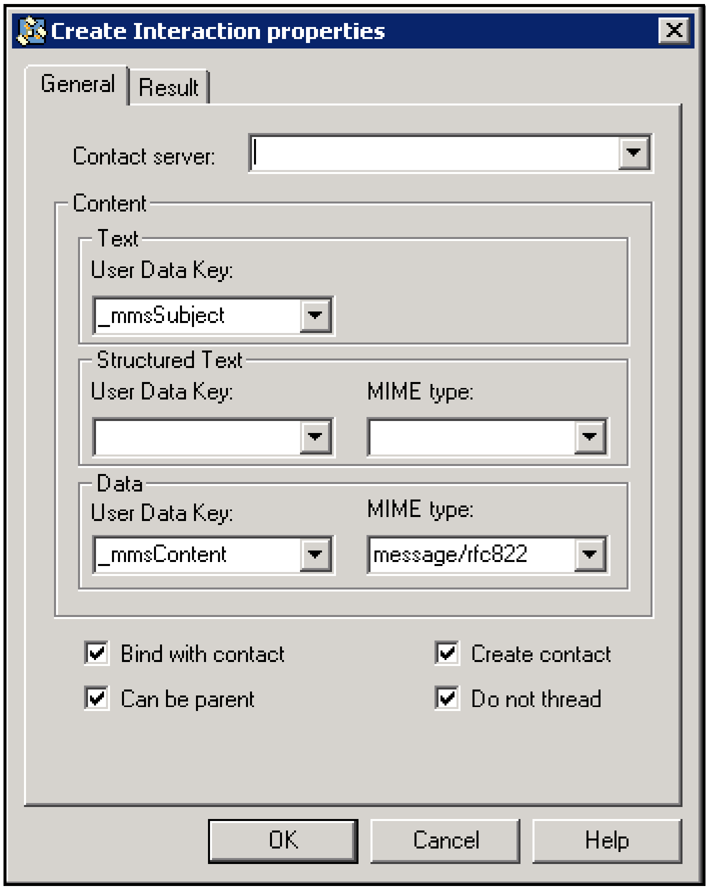


**Figure 189: MMS Inbound Processing Strategy**

## Summary of Flow

**Note:** The IRD objects that are described this section are keyed to the numbers in Figure 189.

1. The Create Interaction object (see Figure 190) searches for the interaction's contact in the UCS database. If the contact is found, the interaction is associated with it. If the contact is not found, a new contact record is created in the UCS database and the interaction is associated with the new contact record. The information that is entered in the various fields—such as first and last name, account number, email address, and so on—are taken from the user data that is attached to the interaction.



Contact server:

Content

Text
User Data Key: _mmsSubject

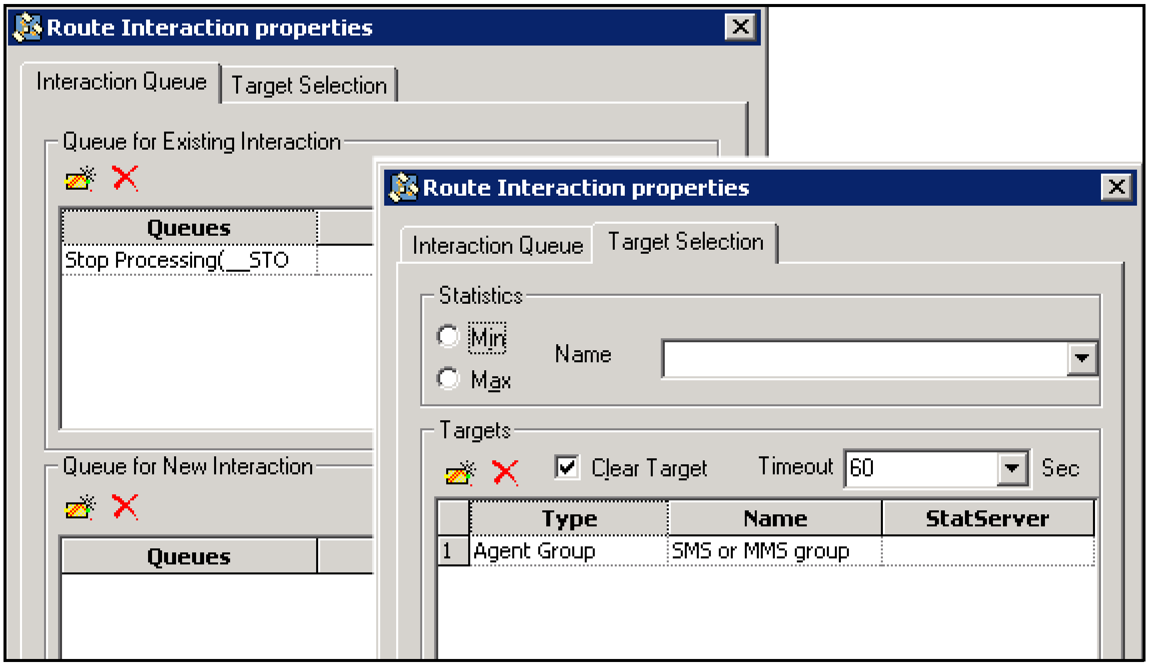
Structured Text
User Data Key:
MIME type:

Data
User Data Key: _mmsContent
MIME type: message/rfc822

☑ Bind with contact ☑ Create contact
☑ Can be parent ☑ Do not thread

Figure 190: Create Interaction Object

2. A Route Interaction object attempts to route the interaction to an SMS or MMS agent group. See Figure 191.



☑ Clear Target

Timeout 60 Sec

| | Type | Name | StatServer |
|---|---|---|---|
| 1 | Agent Group | SMS or MMS group | |

Figure 191: Route Interaction Object

# SMS Processing

The ABC Simple SMS Paging business process contains the following strategies:

1. SMS Inbound strategy
2. SMS Outbound strategy

The ABC Simple SMS Paging business process contains the following queues:

1. SMS Paging Inbound
2. SMS Paging Outbound
3. SMS Paging Default

Figure 192 shows the ABC Simple SMS Paging business process and the two strategies that are included in it, as it appears in the Interaction Design window.

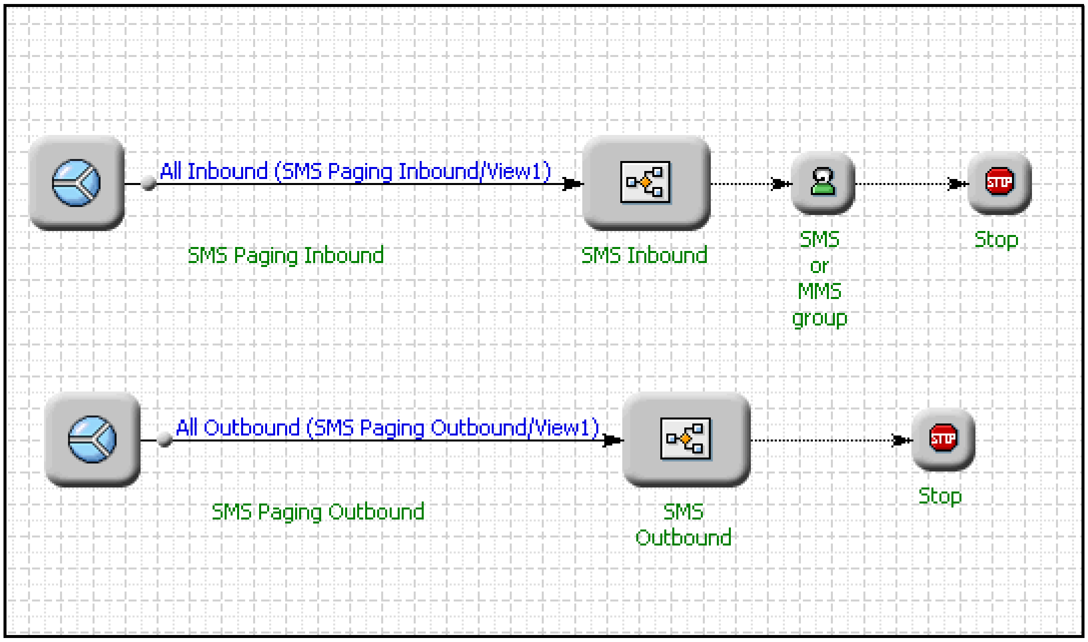


**Figure 192: ABC Simple SMS Paging Business Process**

## SMS Inbound Strategy

The purpose of this strategy is to route an inbound SMS message to an appropriate agent. Figure 193 shows the `SMS Inbound` strategy.

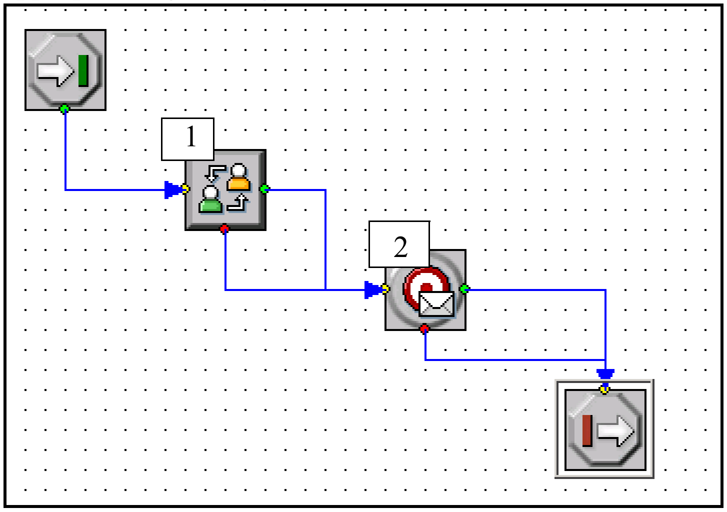
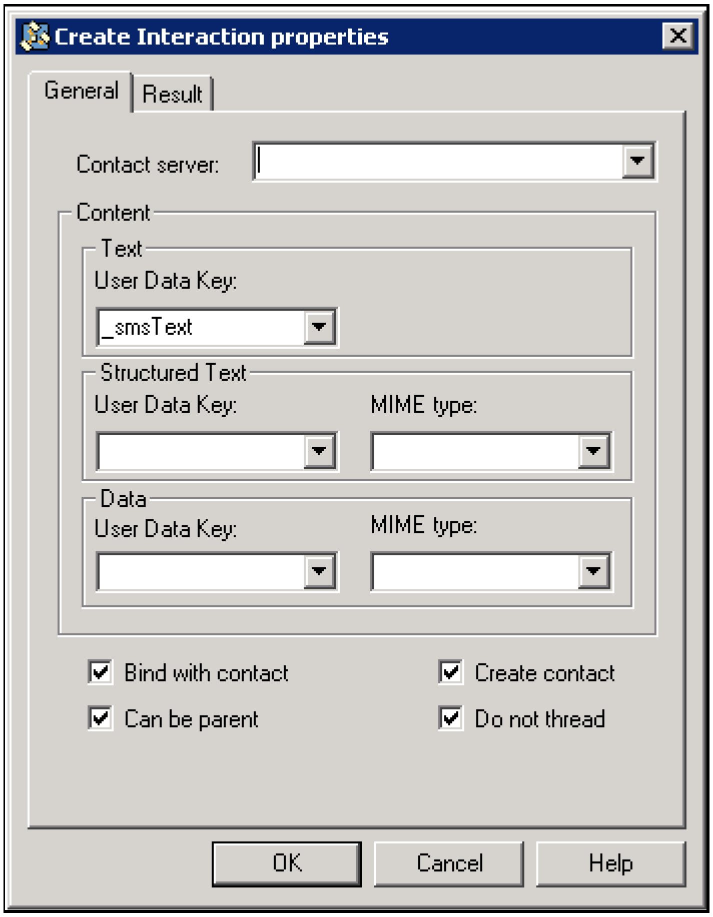


**Figure 193: SMS Inbound Strategy**

## Summary of Flow

**Note:** The IRD objects that are described this section are keyed to the numbers in Figure 193.

1. The Create Interaction object (see Figure 194) searches for the interaction's contact in the UCS database. If the contact is found, the interaction is associated with it. If the contact is not found, a new contact record is created in the UCS database and the interaction is associated with the new contact record. The information that is entered in the various fields—such as first and last name, account number, email address, and so on—are taken from the user data that is attached to the interaction.



Create Interaction properties

General | Result

Contact server:

Content

Text

User Data Key: _smsText

Structured Text

User Data Key:

MIME type:

Data

User Data Key:

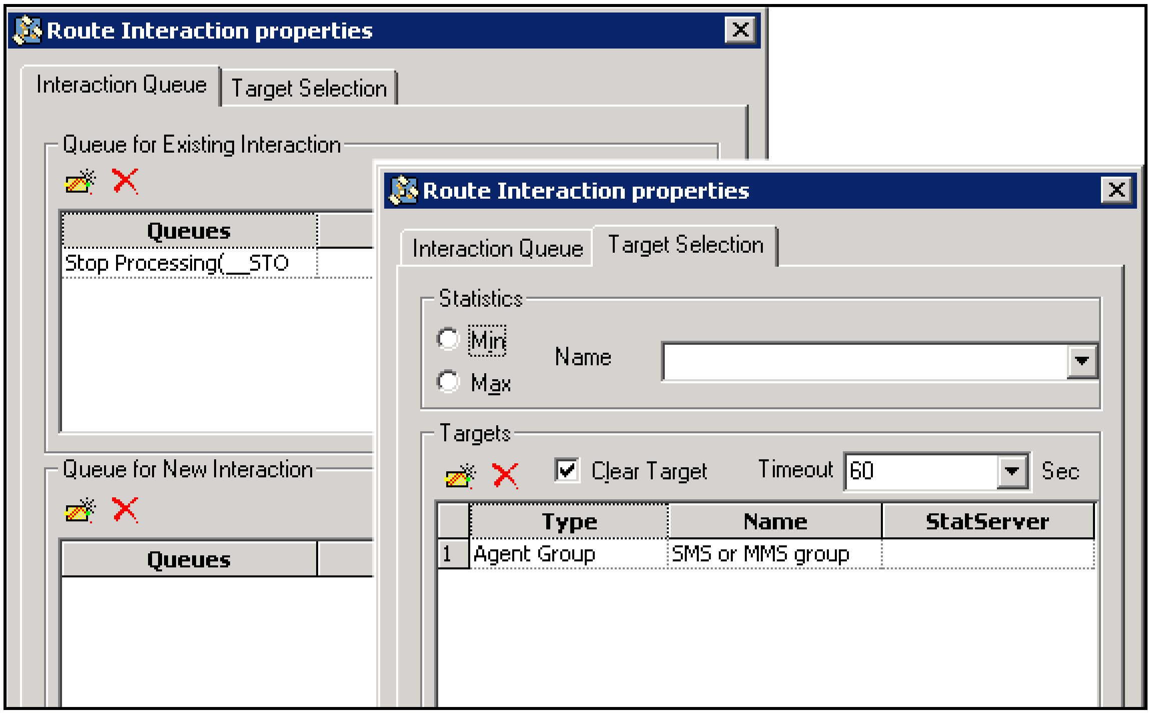
MIME type:

☑ Bind with contact ☑ Create contact

☑ Can be parent ☑ Do not thread

**Figure 194: Create Interaction Object**

2. A Route Interaction object attempts to route the interaction to an SMS or MMS agent group. See Figure 195.


Route Interaction properties
Interaction Queue
Target Selection
Statistics
Min
Max
Name
Targets
Queue for Existing Interaction
Queue for New Interaction

Route Interaction properties — Interaction Queue

Queue for Existing Interaction

| Queues |
|---|
| Stop Processing(__STO |

Queue for New Interaction

| Queues |
|---|

Route Interaction properties — Target Selection

Statistics: ○ Min ○ Max Name:

Targets: ☑ Clear Target Timeout 60 Sec

|  | Type | Name | StatServer |
|---|---|---|---|
| 1 | Agent Group | SMS or MMS group |  |

Figure 195: Route Interaction Object

## SMS Outbound Strategy

The purpose of this strategy is to send a reply SMS message to a contact. Figure 196 shows the SMS Outbound strategy.

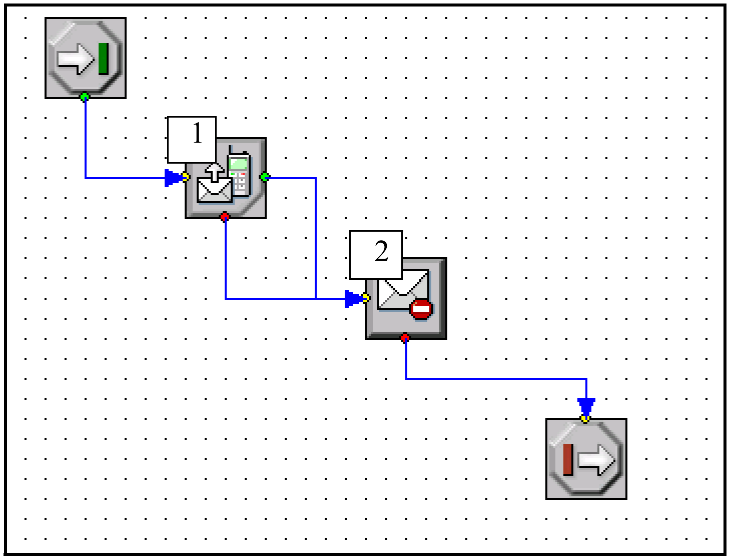


Figure 196: SMS Outbound Strategy

## Summary of Flow

**Note:** The IRD objects that are described this section are keyed to the numbers in Figure 193.

1. The Send SMS Out object (see Figure 197) creates a `Send SMS Out` ESP request to SMS Server. The request contains all of the data that is needed by SMS Server to form and send the outbound SMS message. (The data, which SMS Server needs to form and send the outbound SMS message, is coded in the interaction's attributes).

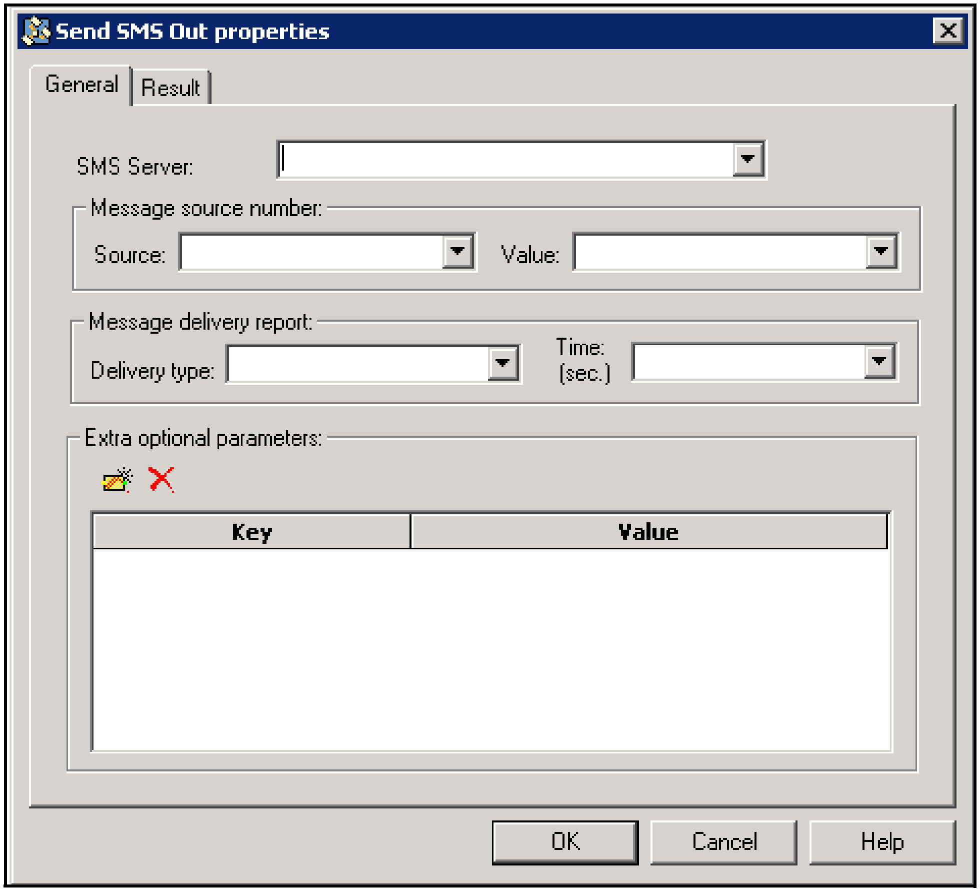
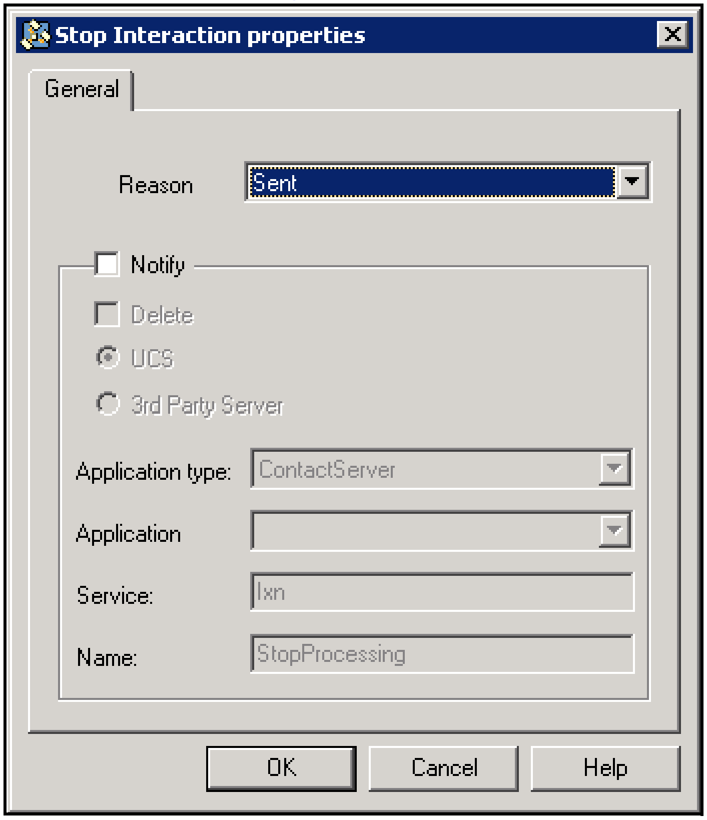


Figure 197: Send SMS Out Object

2. A Stop Interaction object stops processing and notifies UCS (via Interaction Server) that SMS processing is finished. It also supplies a reason code (see Figure 198).



Reason: Sent

☐ Notify

☐ Delete

● UCS

○ 3rd Party Server

Application type: ContactServer

Application:

Service: Ixn

Name: StopProcessing

**Figure 198: Stop Interaction Object**

# Web Callback Processing

The `WebCallback` business process contains the following strategies:

1. `Delivering` strategy
2. `Expired Conference Callbacks` strategy
3. `Expired Transfer Callbacks` strategy
4. `Outbound notification email sending` strategy
5. `Preprocessing` strategy
6. `Rescheduled by Agent` strategy
7. `Rescheduled by Customer` strategy
8. `Stop By Agent` strategy
9. `Stop By Customer` strategy

The `WebCallback` business process contains the following subroutines:

1. `Check Customer Session State`

2. `Check Interaction`
3. `Check Maximum Attempts`
4. `Check Maximum Waiting Time`
5. `Increment Number of Attempts`
6. `Schedule Web Callback`
7. `Send Email Notification`
8. `Stop Web Callback`

Figure 199 shows the `WebCallback` business process and the strategies included in it, as it appears in the Interaction Design window.

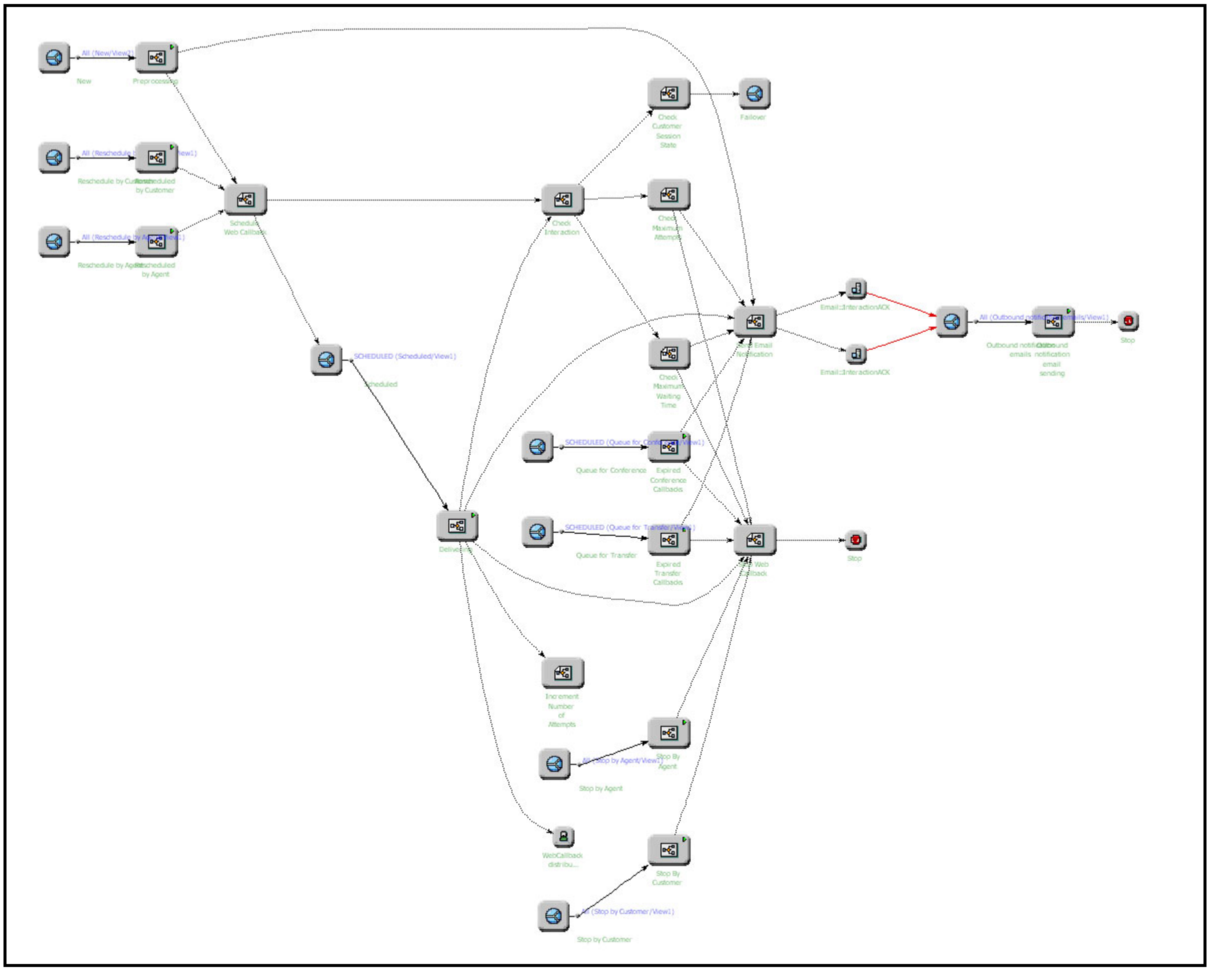

All (New/View2)
New
Preprocessing
All (Reschedule by Customer/View1)
Reschedule by Customer
Rescheduled by Customer
All (Reschedule by Agent/View1)
Reschedule by Agent
Rescheduled by Agent
Schedule Web Callback
SCHEDULED (Scheduled/View1)
Scheduled
Check Interaction
Check Customer Session State
Failover
Check Maximum Attempts
Check Maximum Waiting Time
Send Email Notification
Email::InteractionACK
Email::InteractionACK
All (Outbound notification emails/View1)
Outbound notification emails
Outbound notification email sending
Stop
SCHEDULED (Queue for Conference/View1)
Queue for Conference
Expired Conference Callbacks
SCHEDULED (Queue for Transfer/View1)
Queue for Transfer
Expired Transfer Callbacks
Delivering
Stop Web Callback
Stop
Increment Number of Attempts
All (Stop by Agent/View1)
Stop By Agent
Stop by Agent
WebCallback distribu...
Stop By Customer
All (Stop by Customer/View1)
Stop by Customer

**Figure 199: WebCallback Business Process**

## Preprocessing Strategy

The purpose of this strategy is to create a new incoming web callback interaction in UCS, send an e-mail notification to the customer and schedule a web callback for delivery to an agent.

Figure 200 shows the `Preprocessing` strategy.

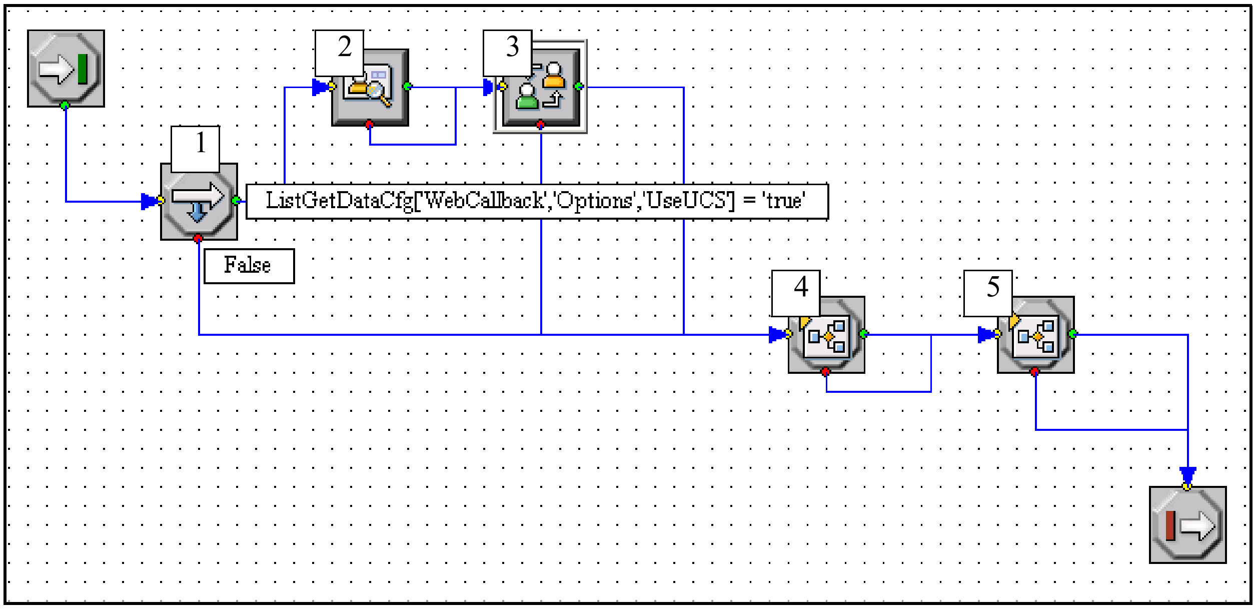


**Figure 200: Preprocessing Strategy**

## Summary of Flow

**Note:** The IRD objects that are described this section are keyed to the numbers in Figure 200.

1. The `UseUCS` configuration option is read from the configuration options. If `UseUCS = true`, processing continues at Step 2. Otherwise, processing continues at Step 4.
2. The contact is created in UCS.
3. The interaction is created in UCS.
4. The interaction goes to a subroutine that sends an e-mail notification to the customer. See “Send Email Notification Subroutine” on page 181 for a description of the subroutine.
5. The interaction goes to a subroutine that schedules a web callback. See “Schedule Web Callback Subroutine” on page 180 for a description of the subroutine.

# Delivering Strategy

The purpose of this strategy is to route an interaction to an agent. Figure 201 shows the Delivering strategy.

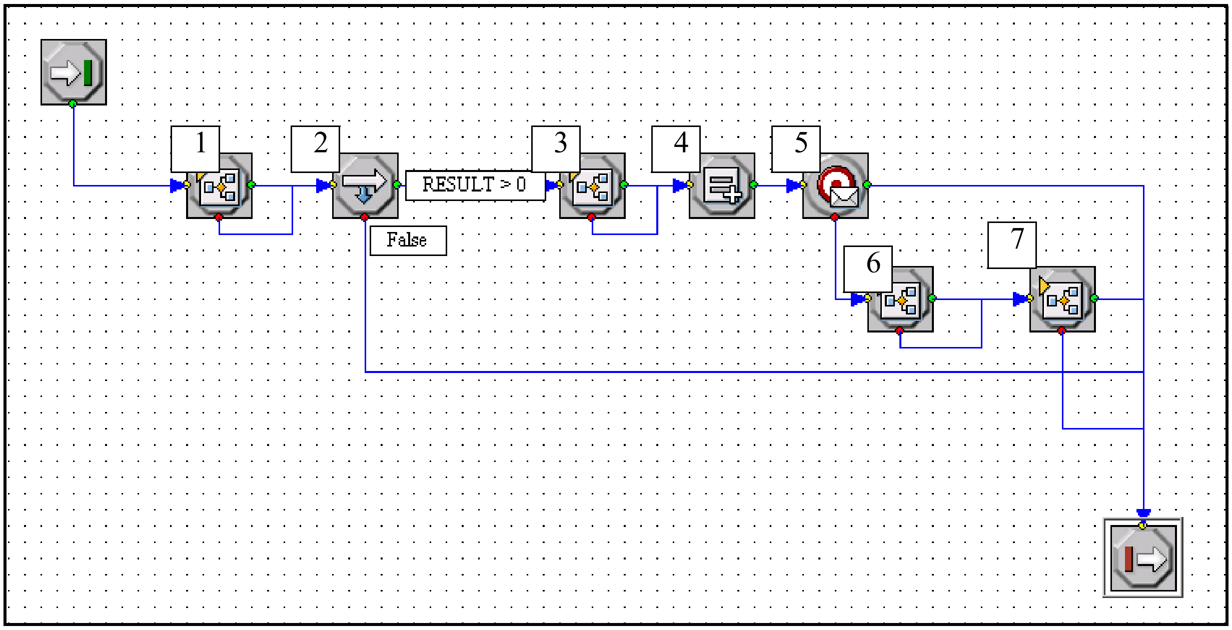


**Figure 201: Delivering Strategy**

## Summary of Flow

**Note:** The IRD objects that are described this section are keyed to the numbers in Figure 201.

1. The Check Interaction subroutine is called.
2. If the output parameter RESULT equals 0, the strategy exits.
3. The Increment Number of Attempts subroutine is called. See “Increment Number of Attempts Subroutine” on page 179 for a description of the subroutine.
4. The maximum time for an interaction to wait for an available agent is calculated.
5. If an available agent is found, the interaction is routed to the agent and the strategy exits.
6. If no available agent is found, the Send Email Notifications subroutine is called to notify the customer.
7. The Stop Web Callback subroutine is called to stop the web callback interaction. See “Stop Web Callback Subroutine” on page 182 for a description of the subroutine.

# Rescheduled by Agent Strategy

The purpose of this strategy is to reschedule web callback interactions that were placed in the `Reschedule by Agent` queue. Figure 202 shows the `Rescheduled by Agent` strategy.

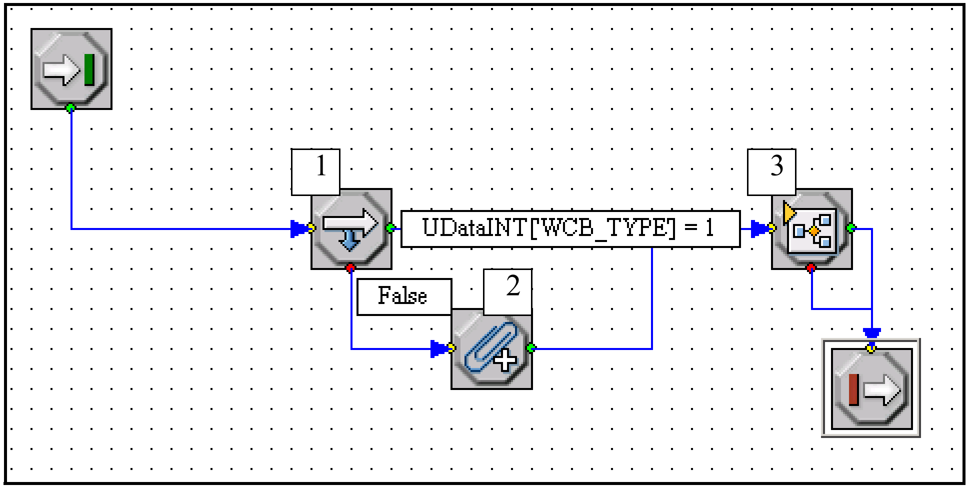


**Figure 202: Rescheduled by Agent Strategy**

## Summary of Flow

**Note:** The IRD objects that are described this section are keyed to the numbers in Figure 202.

1. A check is made whether the web callback interaction type is equal to `1` (scheduled).
2. If the web callback interaction type is not equal to `1`, the interaction type is set to `1` (scheduled).
3. The `Schedule Web Callback` subroutine is called. See “Schedule Web Callback Subroutine” on page 180 for a description of the subroutine.

# Rescheduled by Customer Strategy

The purpose of this strategy is to reschedule web callback interactions that were placed in the `Reschedule by Customer` queue. Figure 203 shows the `Rescheduled by Customer` strategy.

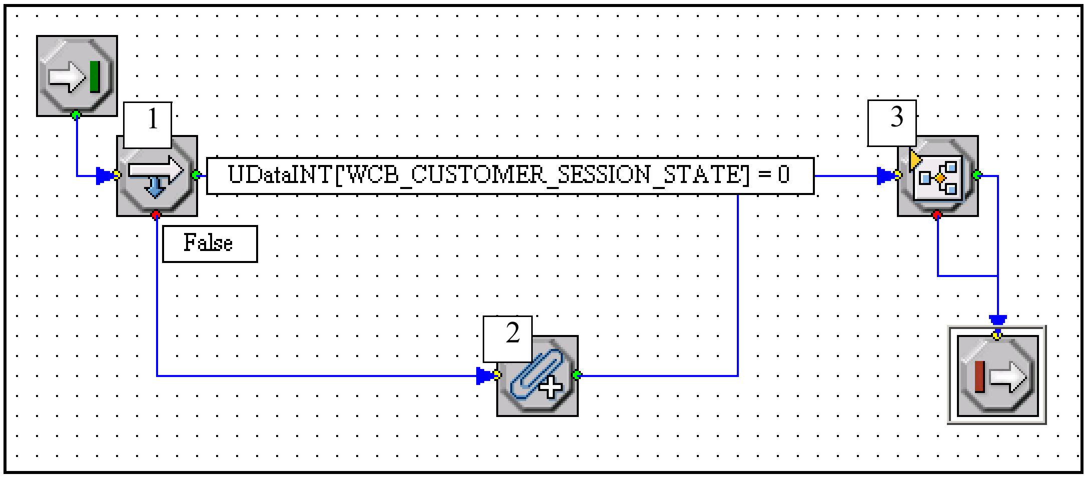


Figure 203: Rescheduled by Customer Strategy

## Summary of Flow

**Note:** The IRD objects that are described this section are keyed to the numbers in Figure 203.

1. A check is made as to whether the customer session state is equal to `0` (Not Active).
2. If he customer session state is not equal to `0`, the customer session state is set to `0` (Not Active).
3. The `Schedule Web Callback` subroutine is called. See “Schedule Web Callback Subroutine” on page 180 for a description of the subroutine.

# Expired Conference Callbacks Strategy

The purpose of this strategy is to handle interactions that where not pulled back from the Queue for Conference queue by an agent after the end of the conference. Figure 204 shows the Expired Conference Callbacks strategy.

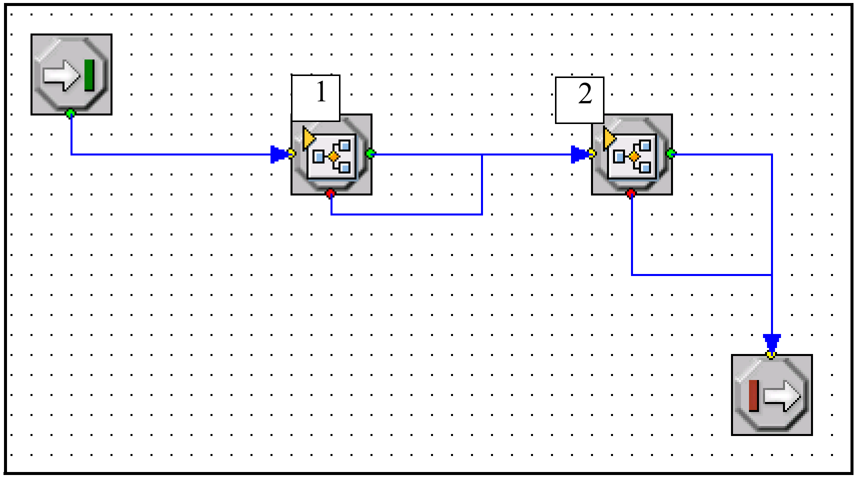


**Figure 204: Expired Conference Callbacks Strategy**

## Summary of Flow

**Note:** The IRD objects that are described this section are keyed to the numbers in Figure 204.

1. The Send Email Notification subroutine is called. See "Send Email Notification Subroutine" on page 181 for a description of the subroutine.
2. The Stop Web Callback subroutine is called. See "Stop Web Callback Subroutine" on page 182 for a description of the subroutine.

# Expired Transfer Callbacks Strategy

The purpose of this strategy is to handle interactions that were not pulled back from the `Queue for Transfer` queue by an agent after the end of the transfer. Figure 205 shows the `Expired Transfer Callbacks` strategy.

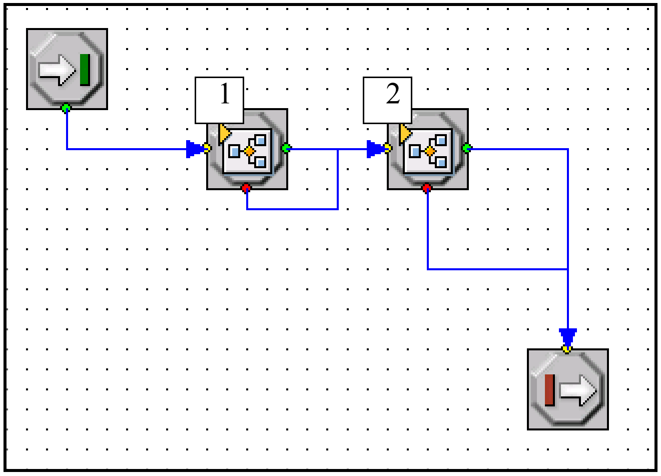


**Figure 205: Expired Transfer Callbacks Strategy**

## Summary of Flow

**Note:** The IRD objects that are described this section are keyed to the numbers in Figure 205.

3. The `Send Email Notification` subroutine is called. See “Send Email Notification Subroutine” on page 181 for a description of the subroutine.
4. The `Stop Web Callback` subroutine is called. See “Stop Web Callback Subroutine” on page 182 for a description of the subroutine.

# Outbound notification email sending Strategy

The purpose of this strategy is to send e-mails. Figure 206 shows the `Outbound notification email sending` strategy.

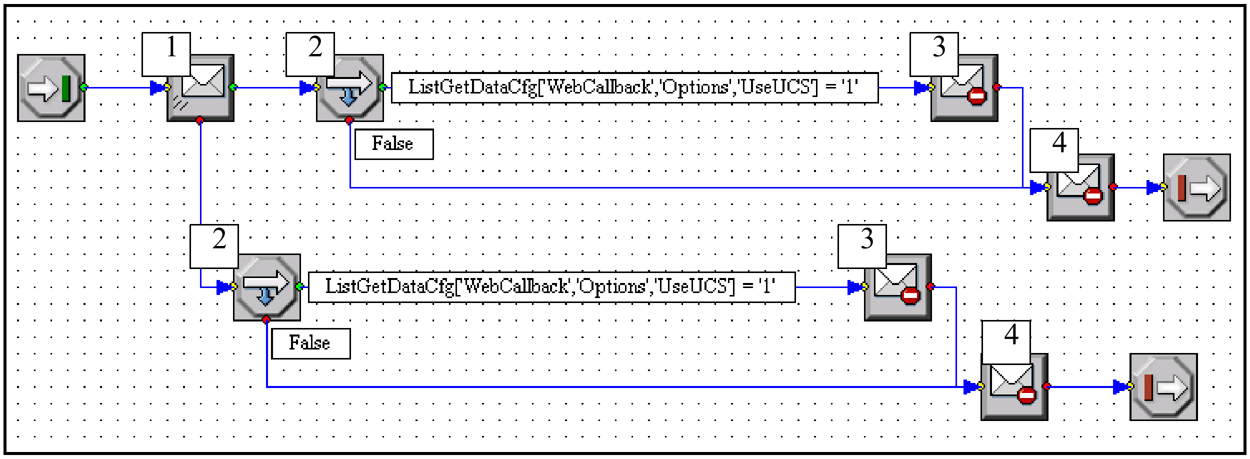

1
2
ListGetDataCfg['WebCallback','Options','UseUCS'] = '1'
False
3
4
2
ListGetDataCfg['WebCallback','Options','UseUCS'] = '1'
False
3
4

**Figure 206: Outbound notification email sending Strategy**

## Summary of Flow

1. An e-mail is sent.
2. A check is made as to whether the `UseUCS` configuration option is set to `1` (true).
3. If `UseUCS` is set to `1`, the interaction is stopped and deleted from UCS.
4. If `UseUCS` is not set to `1`, the e-mail interaction is stopped.

# Stop By Agent Strategy

The purpose of this strategy is to stop interactions that an agent has placed in the `Stop by Agent` queue. Figure 207 shows the `Stop By Agent` strategy.

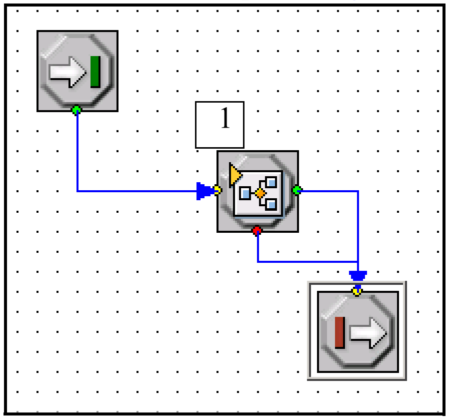


**Figure 207: Stop By Agent Strategy**

### Summary of Flow

**Note:** The IRD objects that are described this section are keyed to the numbers in Figure 207.

- The `Stop Web Callback` subroutine is called. See “Stop Web Callback Subroutine” on page 182 for a description of the subroutine.

## Stop By Customer Strategy

The purpose of this strategy is to stop interactions that were placed in the `Stop by Customer` queue. Figure 208 shows the `Stop By Customer` strategy.

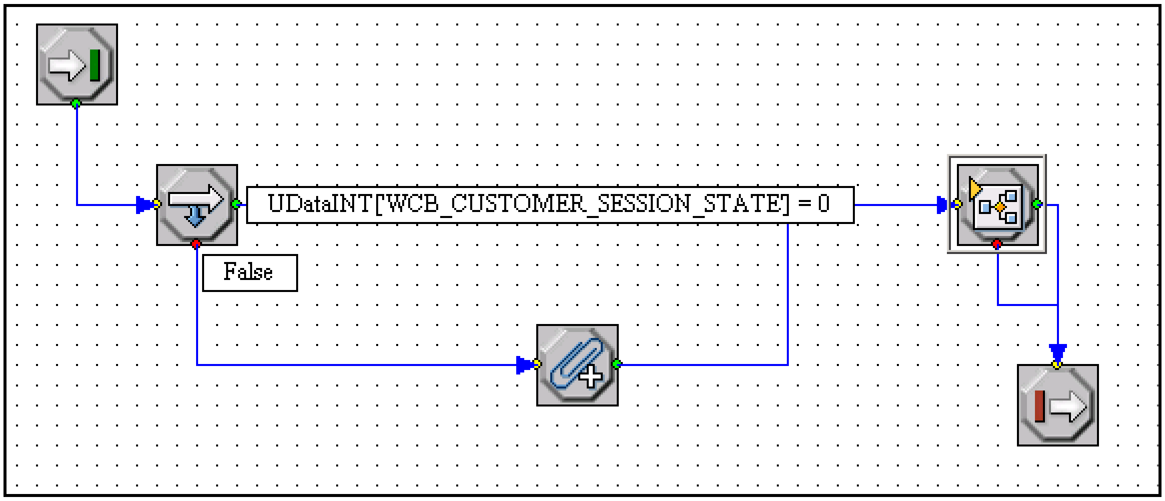


**Figure 208: Stop by Customer Strategy**

### Summary of Flow

**Note:** The IRD objects that are described this section are keyed to the numbers in Figure 208.

1. A check is made as to whether the customer session state is equal to `0` (Not Active).
2. If the customer session state is not equal to `0`, the customer session state is set to `0` (Not Active).
3. The `Stop Web Callback` subroutine is called. See “Stop Web Callback Subroutine” on page 182 for a description of the subroutine.

# Check Customer Session State Subroutine

The purpose of this subroutine is to check the customer session state and update the result string. Figure 209 shows the `Check Customer Session State` subroutine.

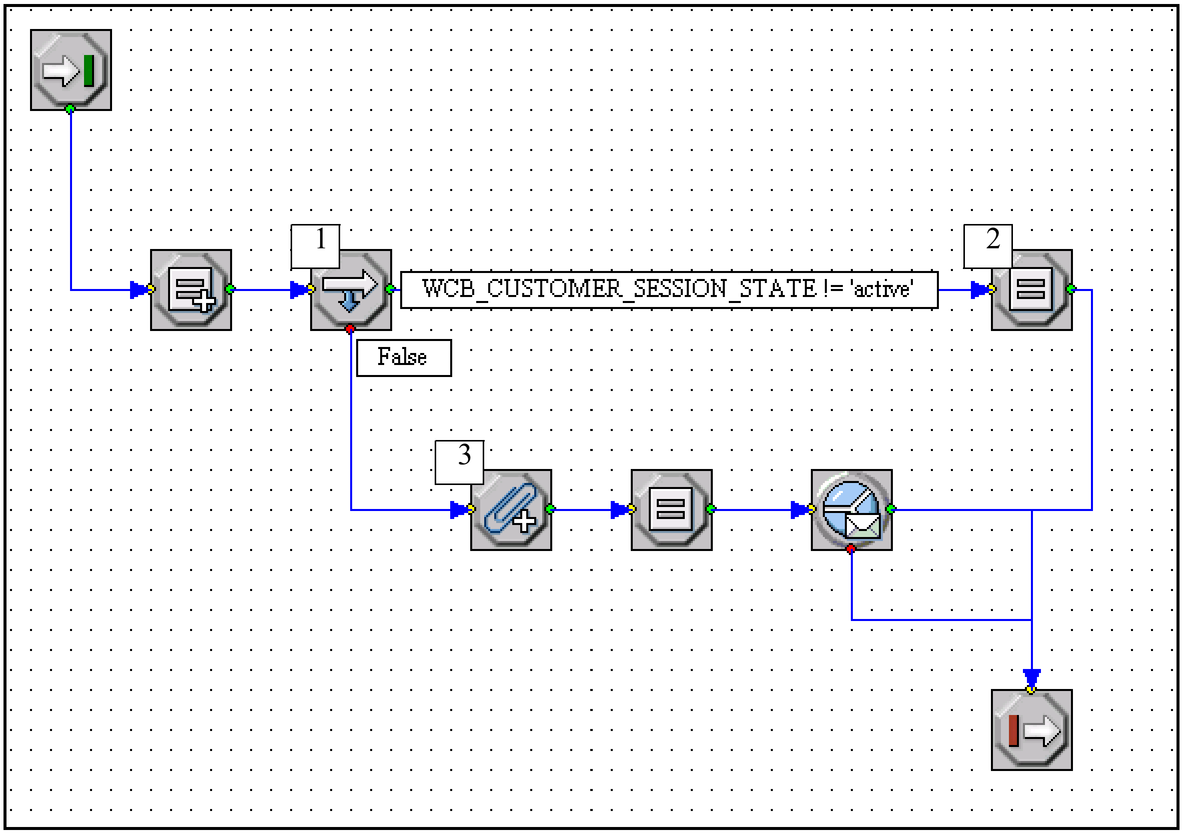


**Figure 209: Check Customer Session State Subroutine**

## Summary of Flow

**Note:** The IRD objects that are described this section are keyed to the numbers in Figure 209.

1. A check is made as to whether the customer session state is active.
2. If the customer session state is not active, the result is set to `1`, and the interaction exits the subroutine.
3. If the customer session state is active, the result is set to `0`, the result string is updated, and the interaction is placed in the `Failover` queue and exits the subroutine.

# Check Interaction Subroutine

The purpose of this subroutine is to perform checks on the interaction. The customer session state, maximum attempts, and maximum waiting time are all checked in this subroutine. Figure 210 shows the `Check Interaction` subroutine.

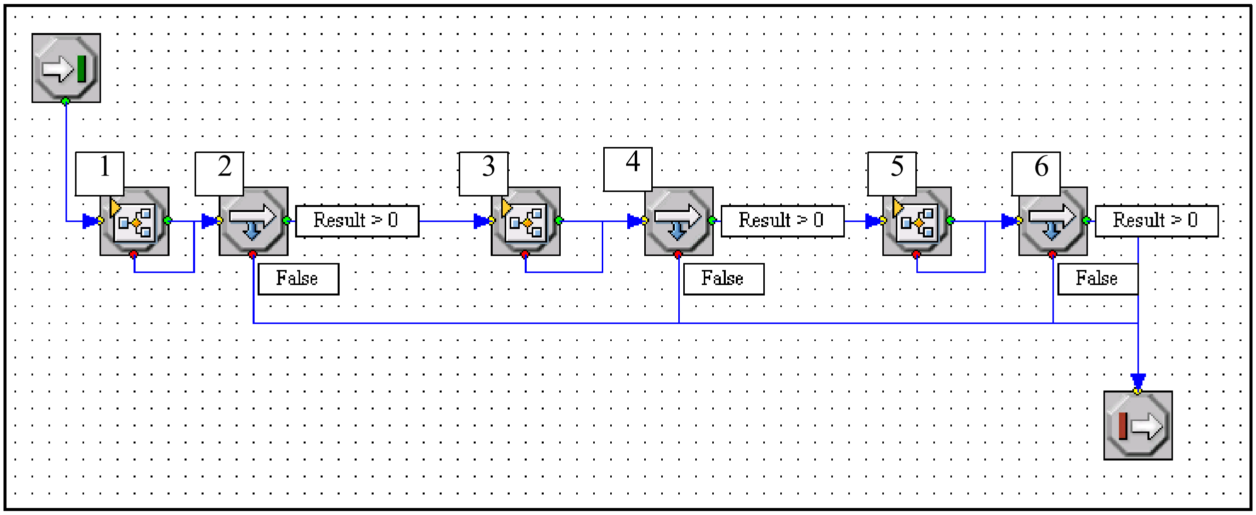


**Figure 210: Check Interaction Subroutine**

## Summary of Flow

**Note:** The IRD objects that are described this section are keyed to the numbers in Figure 210.

1. The `Check Customer Session State` subroutine is called.
2. If the result string is `0`, the interaction exits the subroutine.
3. The `Check Maximum Attempts` subroutine is called.
4. If the result string is `0`, the interaction exits the subroutine.
5. The `Check Maximum Waiting Time` subroutine is called.
6. The result is checked and the interaction exits the subroutine.

# Check Maximum Attempts Subroutine

The purpose of this subroutine is to check the maximum attempts for the interaction. Figure 211 shows the `Check Maximum Attempts` subroutine.

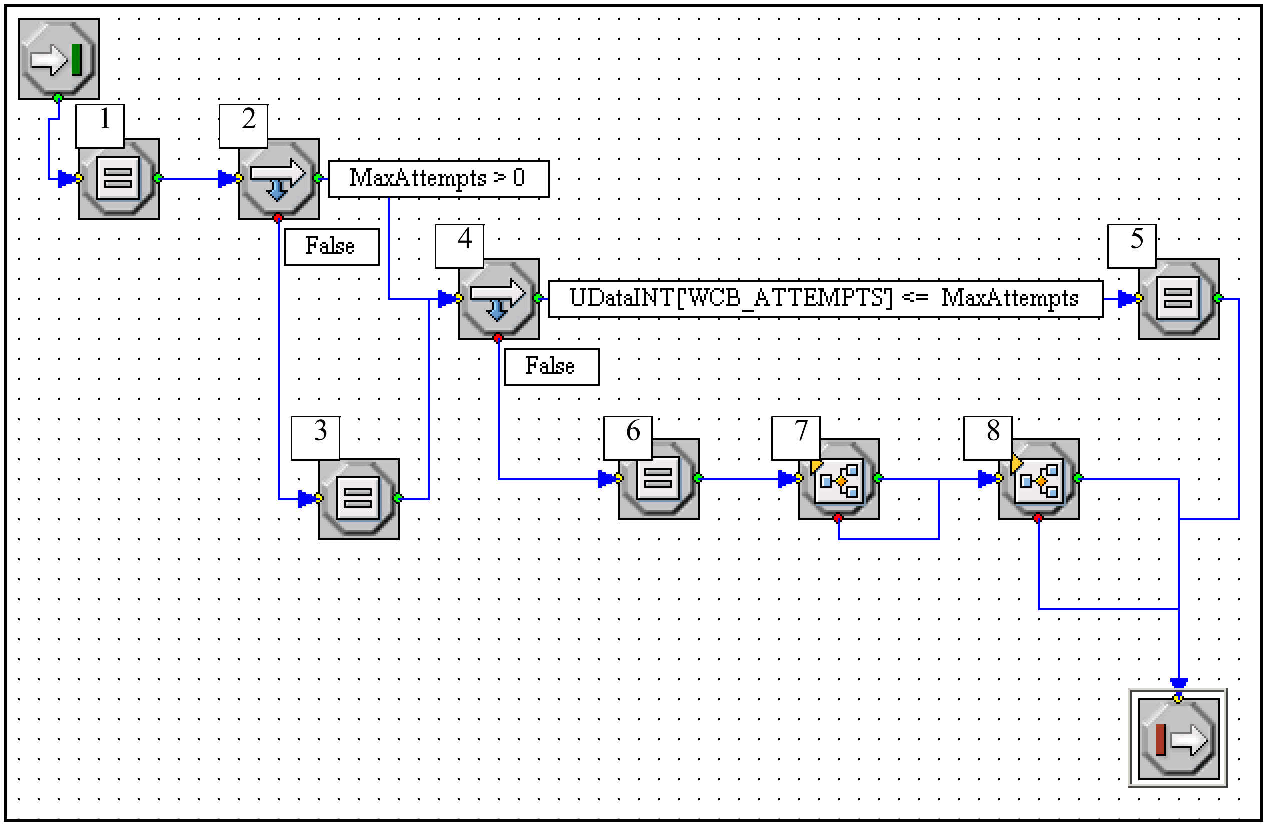


**Figure 211: Check Maximum Attempts Subroutine**

## Summary of Flow

**Note:** The IRD objects that are described this section are keyed to the numbers in Figure 211.

1. The value of the `MaxAttempts` configuration option is retrieved.
2. A check is made as to whether the value of `MaxAttempts` is greater than `0`.
3. If `MaxAttempts` is equal to or less than `0`, the default value is assigned.
4. A check is made as to whether `WCB_ATTEMPTS` in user data is less than or equal to `MaxAttempts`.
5. If `WCB_ATTEMPTS` in user data is less than or equal to `MaxAttempts`, the result is set to `1` and the interaction exits the subroutine.
6. If `WCB_ATTEMPTS` in user data is greater than `MaxAttempts`, the result is set to `0`.

7. The Send Email Notification subroutine is called.
8. The Stop Web Callback subroutine is called, and the interaction exits the subroutine.

# Check Maximum Waiting Time Subroutine

The purpose of this subroutine is to check the maximum waiting time for the interaction. Figure 212 shows the Check Maximum Waiting Time subroutine.

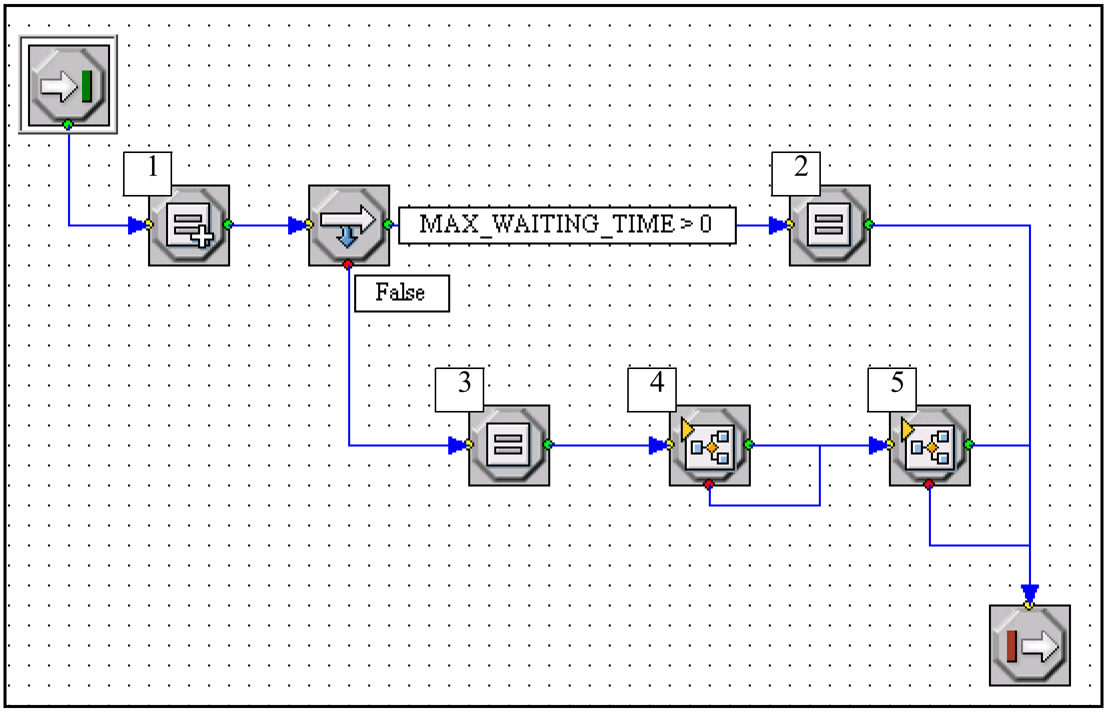


**Figure 212: Check Maximum Waiting Time Subroutine**

## Summary of Flow

**Note:** The IRD objects that are described this section are keyed to the numbers in Figure 212.

1. The MAX_WAITING_TIME variable is set. The value for MAX_WAITING_TIME is calculated by subtracting UTC from the WCB_END_TIME in user data.
2. If MAX_WAITING_TIME is greater than or equal to 0, the result is set to 1 and the interaction exits the subroutine.
3. If the MAX_WAITING_TIME is less than 0, the result is set to 0.
4. The Send Email Notification subroutine is called.
5. The Stop Web Callback subroutine is called, and the interaction exits the subroutine.

# Increment Number of Attempts Subroutine

The purpose of this subroutine is to increment the number of attempts for the web callback interaction. Figure 213 shows the `Increment Number of Attempts` subroutine.

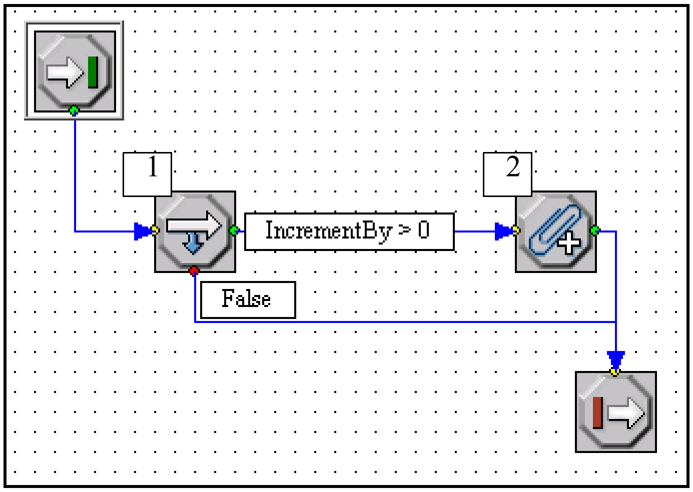


**Figure 213: Increment Number of Attempts Subroutine**

## Summary of Flow

**Note:** The IRD objects that are described this section are keyed to the numbers in Figure 213.

1. If the `IncrementBy` input variable is not greater than `0`, the interaction exits the subroutine.
2. If the `IncrementBy` input variable is greater than `0`, `WCB_ATTEMPTS` in user data is incremented by the `IncrementBy` input variable, and the interaction exits the subroutine.

# Schedule Web Callback Subroutine

The purpose of this subroutine is to schedule the web callback interaction. Figure 214 shows the `Schedule Web Callback` subroutine.

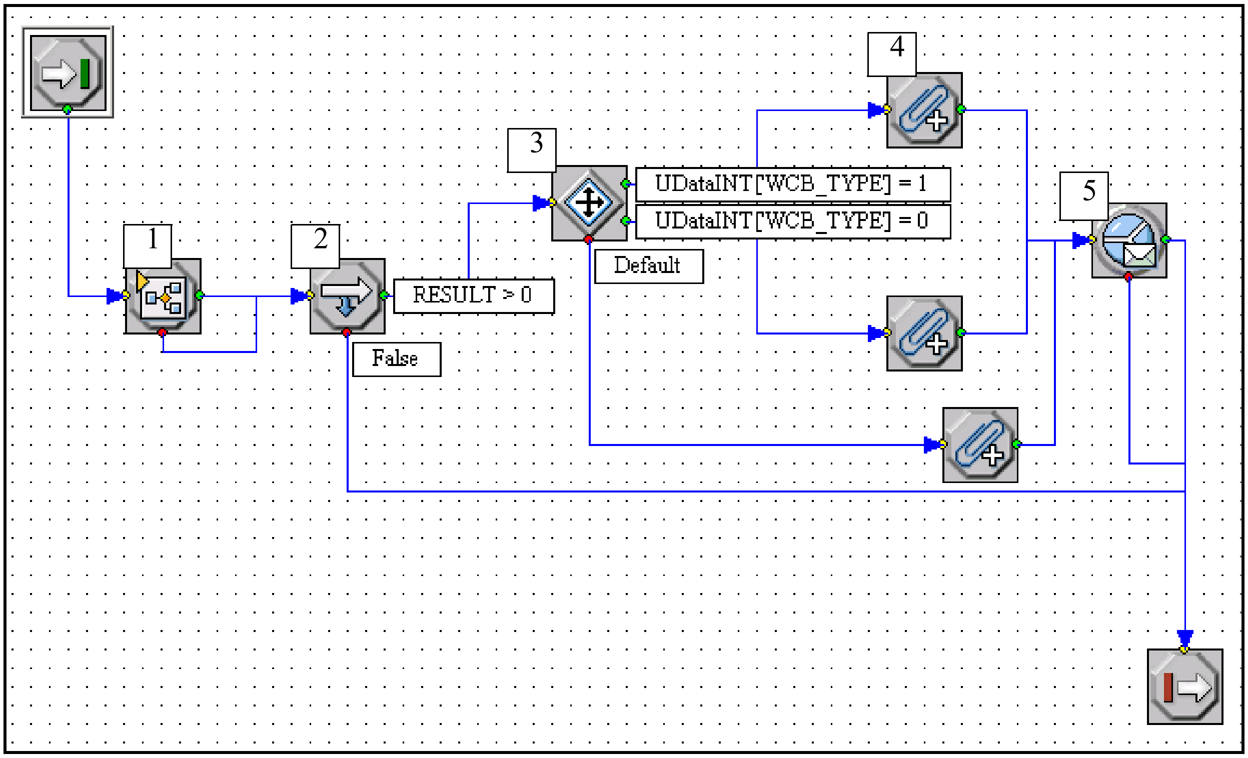


**Figure 214: Schedule Web Callback Subroutine**

## Summary of Flow

**Note:** The IRD objects that are described this section are keyed to the numbers in Figure 214.

1. The `Check Interaction` subroutine is called.
2. If the result is `0`, the interaction exits the subroutine.
3. If the result is greater than `0`, a check is made as to whether the value of `WCB_TYPE` in user data is equal to 1 or `0`.
4. `ScheduledAt` in user data is updated with `WCB_START_TIME` from user data.
5. The interaction is placed in the Scheduled queue, and the interaction exits the subroutine.

# Send Email Notification Subroutine

The purpose of this subroutine is to send e-mail notifications to the customer. Figure 215 shows the `Send Email Notification` subroutine.

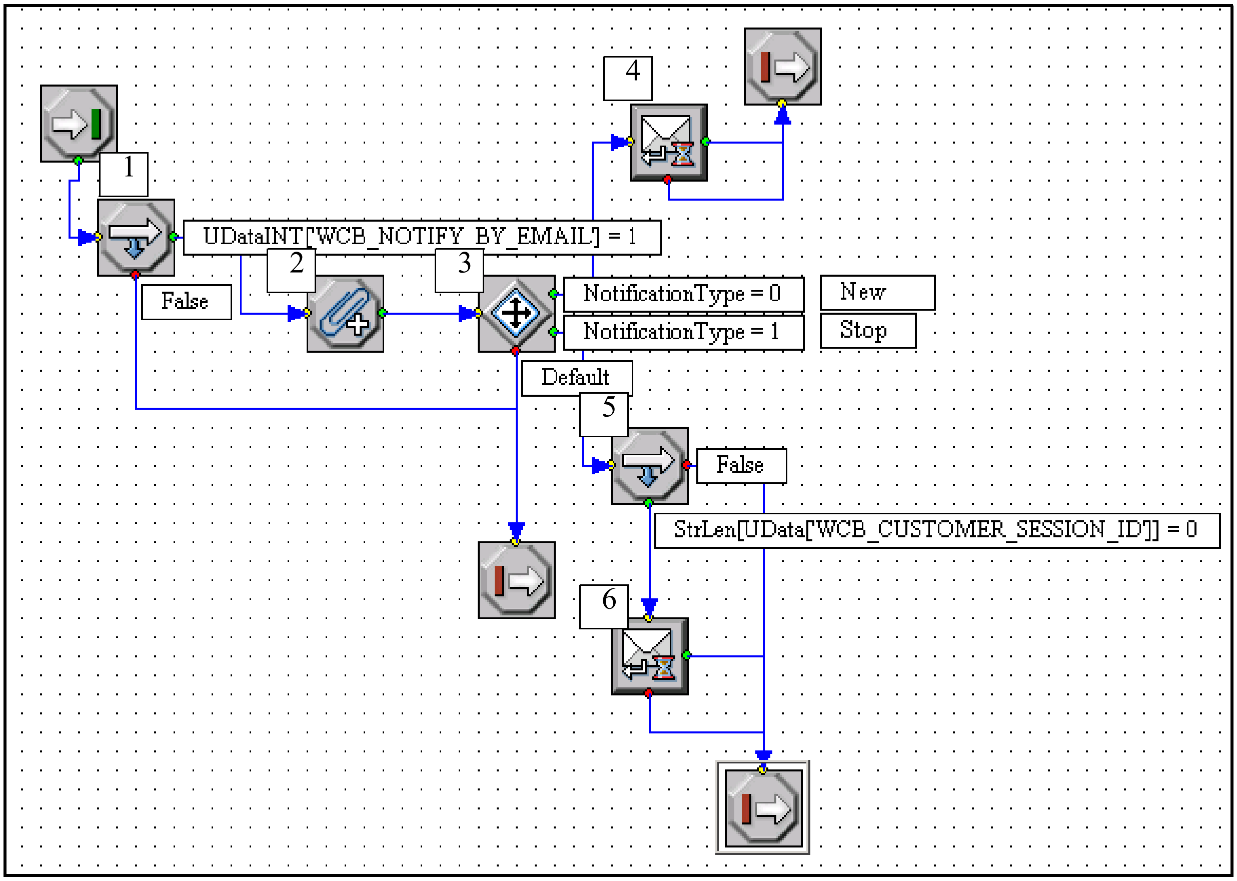

1
UDataINT['WCB_NOTIFY_BY_EMAIL'] = 1
False
2
3
4
NotificationType = 0
NotificationType = 1
New
Stop
Default
5
False
StrLen[UData['WCB_CUSTOMER_SESSION_ID']] = 0
6

**Figure 215: Send Email Notification Subroutine**

## Summary of Flow

**Note:** The IRD objects that are described this section are keyed to the numbers in Figure 215.

1. A check is made as to whether `WCB_NOTIFY_BY_EMAIL` is set to `1`. If it is not, the subroutine is exited. If it is, processing continues.
2. `_OutboundToAddress` in attached data is updated with `WCB_NOTIFICATION_EMAIL` from attached data.
3. The NotificationType is checked.
4. If NotificationType is `0`, an Acknowledgement Receipt e-mail is sent that notifies the customer of a new callback. The subroutine is exited.
5. If NotificationType is `1`, `WCB_CUSTOMER_SESSION_ID` is checked. If it has no value, the subroutine is exited.

6. If WCB_CUSTOMER_SESSION_ID has a value, an Acknowledgement receipt e-mail is sent that notifies the customer of a stopped callback . The subroutine is exited.

# Stop Web Callback Subroutine

The purpose of this subroutine is to stop the web callback interaction. Figure 216 shows the Stop Web Callback subroutine.

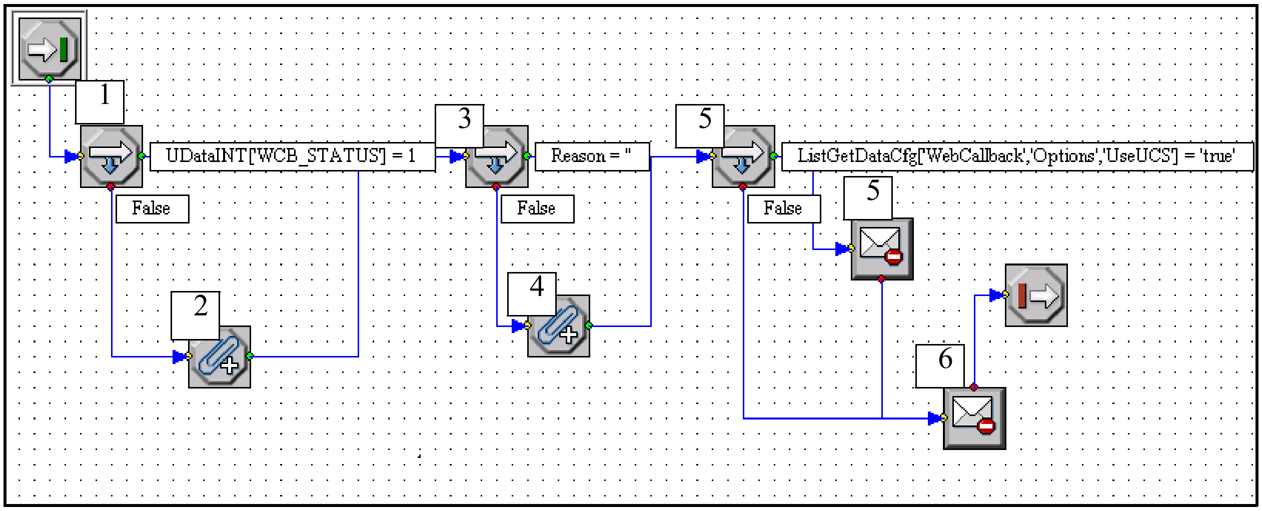

1
UDataINT['WCB_STATUS'] = 1
False
2
3
Reason = ''
False
4
5
ListGetDataCfg['WebCallback','Options','UseUCS'] = 'true'
False
5
6

**Figure 216: Stop Web Callback Subroutine**

## Summary of Flow

**Note:** The IRD objects that are described this section are keyed to the numbers in Figure 216.

1. A check is made as to whether WCB_STATUS is equal to 1.
2. If WCB_STATUS is not equal to 1, WCB_STATUS is set to 1.
3. A check is made as to whether the Reason input parameter is empty.
4. If he Reason input parameter is not empty, update WCB_RESULT in user data with the value of the Reason input parameter.
5. A check is made as to whether the UseUCS configuration option is set to true.
6. If UseUCS is set to true, a notification is sent to UCS.
7. The interaction is stopped.

# How To: Attach Classification Categories

This business process demonstrates how to attach e-mail classification categories to new inbound interactions for the purpose of segmenting those interactions to take different paths in the strategy.

**Note:** If you need additional information on classification categories, see *Universal Routing 8.1 Business Process User's Guide*.

Figure 217 shows the `How to: Attach Classification Categories and use the MultiScreen Object` business process.

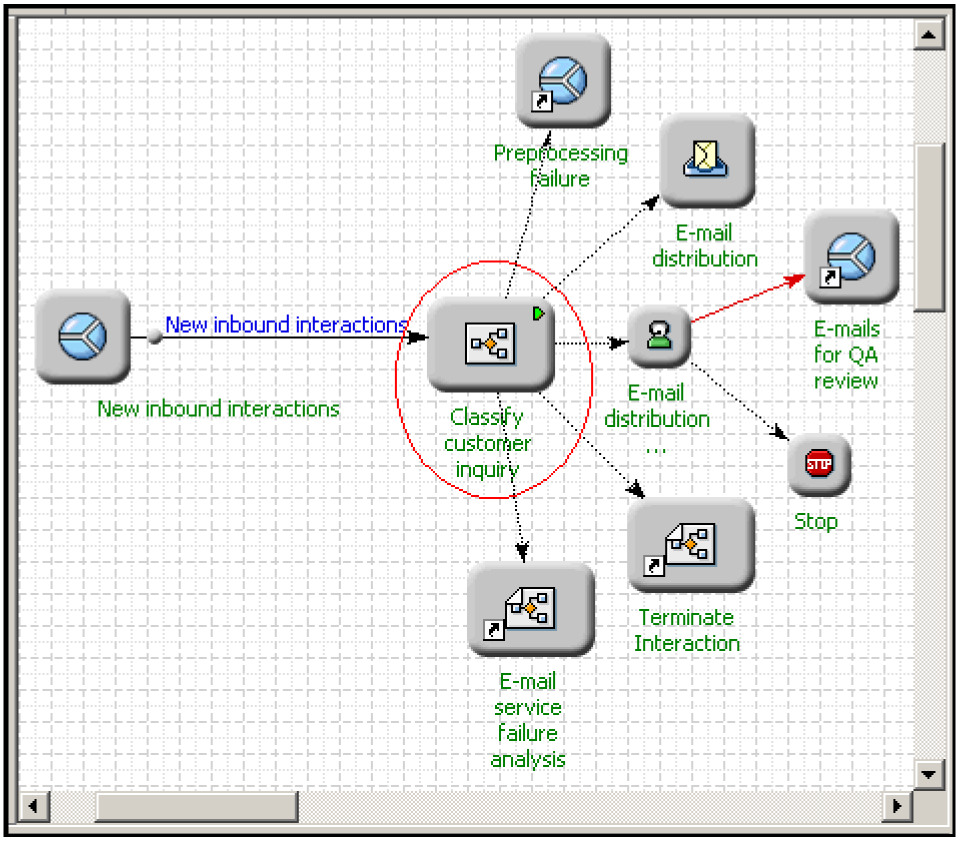


**Figure 217: How to: Attach Classification Categories and Use the MultiScreen Object Business Process**

## Processing Objects

This section describes the various objects in Figure 217.

A view (`New inbound interactions`) attached to a queue (`New inbound interactions`), with no Conditions or Order By information, extracts interactions and sends them to the strategy `Classify customer inquiry (see` Figure 218`)`.

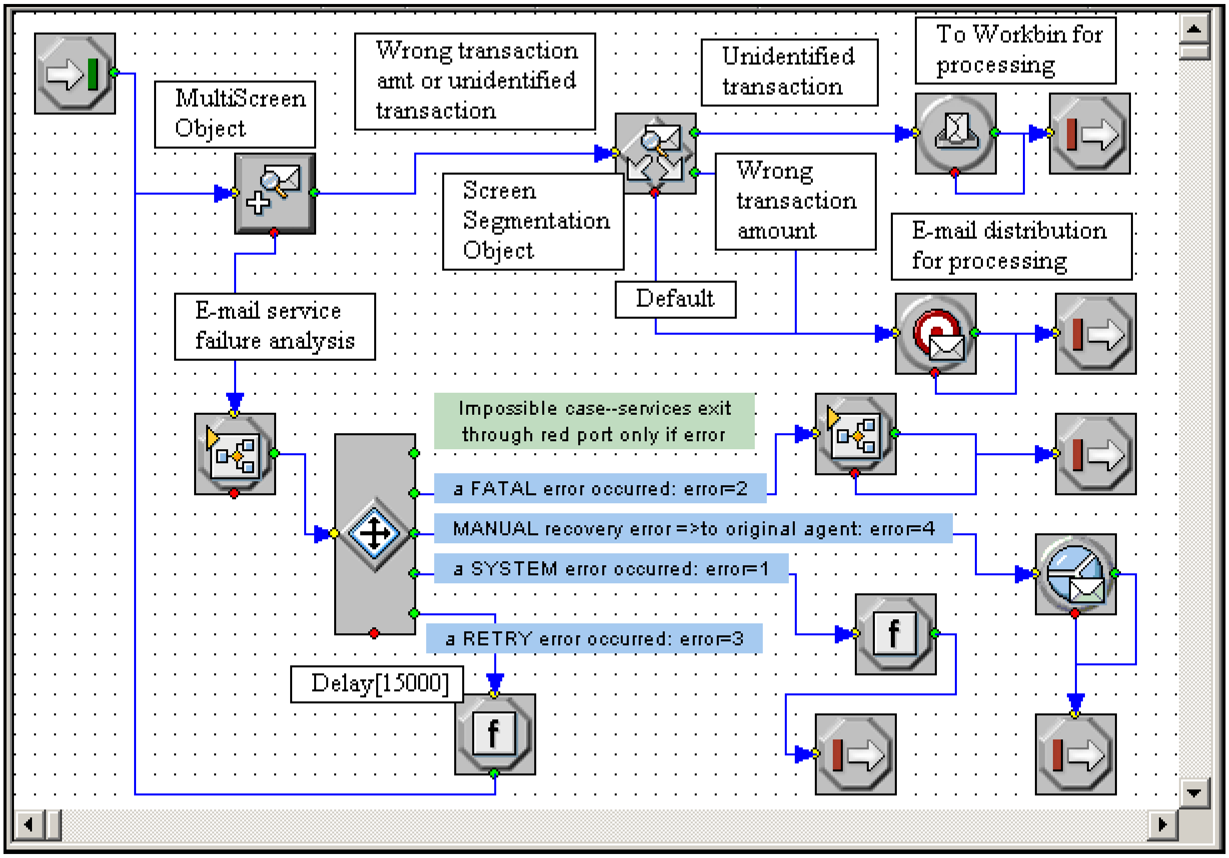
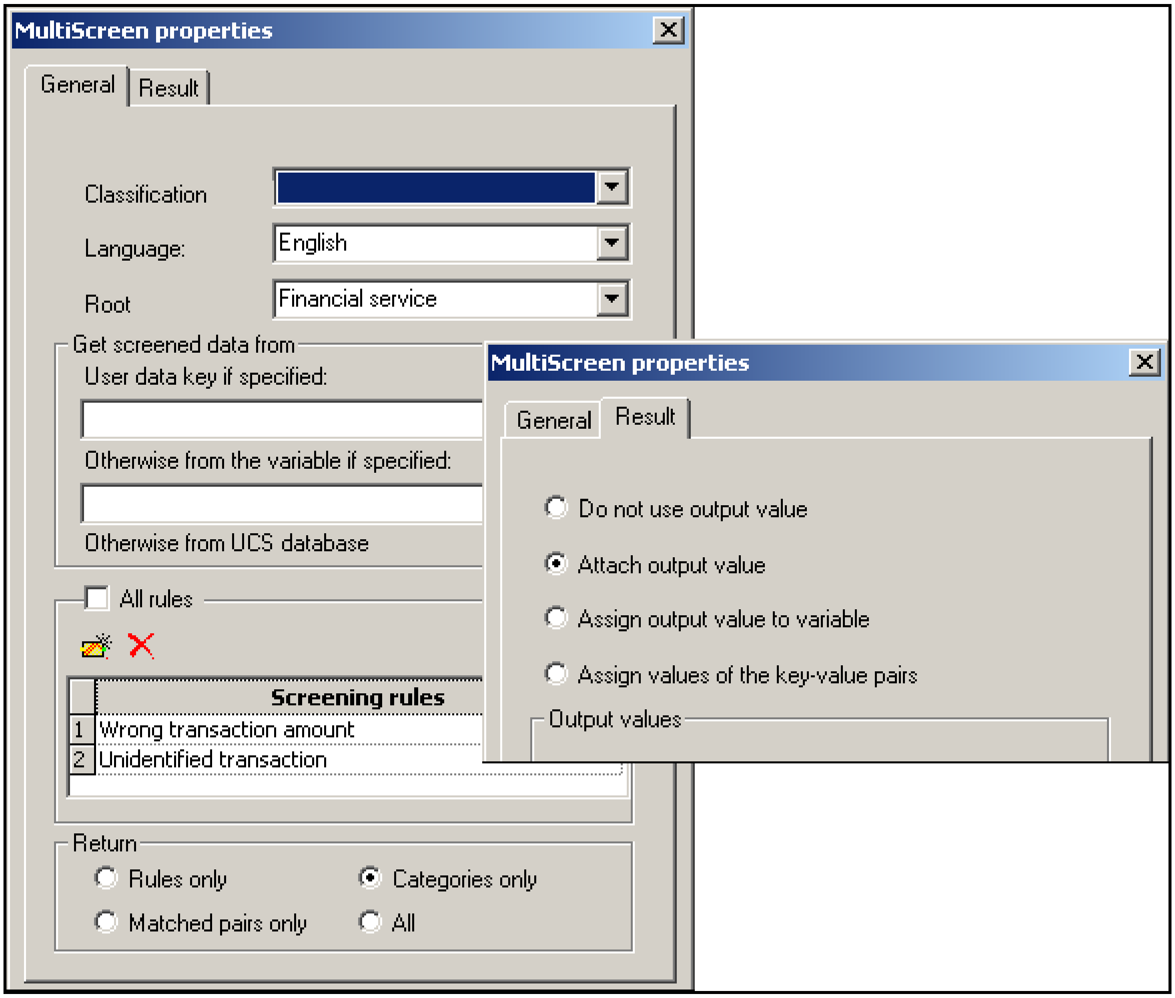

MultiScreen Object
Wrong transaction amt or unidentified transaction
Unidentified transaction
To Workbin for processing
Screen Segmentation Object
Wrong transaction amount
E-mail distribution for processing
Default
E-mail service failure analysis
Impossible case--services exit through red port only if error
a FATAL error occurred: error=2
MANUAL recovery error =>to original agent: error=4
a SYSTEM error occurred: error=1
a RETRY error occurred: error=3
Delay[15000]

**Figure 218: Classify Customer Inquiry Strategy**

## MultiScreen Object

This strategy starts with a MultiScreen object that screens for certain words or word patterns using multiple screening rules and returns the categories associated with the screening rules. You can see the screening rules used and instruction to return classification categories when you open the MultiScreen object properties dialog box (see Figure 219).



MultiScreen properties

General | Result

Classification

Language: English

Root: Financial service

Get screened data from

User data key if specified:

Otherwise from the variable if specified:

Otherwise from UCS database

□ All rules

|   | Screening rules |
|---|---|
| 1 | Wrong transaction amount |
| 2 | Unidentified transaction |

Return

○ Rules only ● Categories only

○ Matched pairs only ○ All

MultiScreen properties

General | Result

○ Do not use output value

● Attach output value

○ Assign output value to variable

○ Assign values of the key-value pairs

Output values

Figure 219: MultiScreen Properties Dialog Box

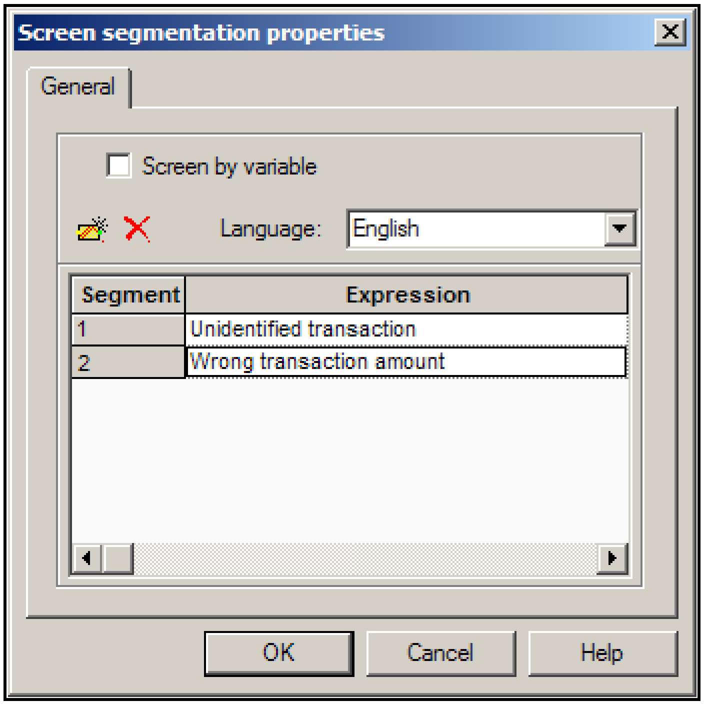
The screening rules are `Wrong transaction amount` and `Unidentified transaction` (included with the strategy samples).

**Note:** When you install the samples as described in the *eServices (Multimedia) 8.0 Deployment Guide*, you use the Knowledge Manager Import function to import a file containing screening rules (`UCS_impex.kme`). If screening rules for a sample strategy do not appear in your MultiScreen Properties dialog box, check whether the rules have been imported into Knowledge Manager.

On the `Return` pane, note that `Categories only` is selected. Categories can be returned because screening rules can be associated with categories in Knowledge Manager. Note also that the Result tab in Figure 219 instructs URS to attach the categories to the interaction.

## Screen Segmentation Object

Interactions with categories attached by the MultiScreen object go to a Screen Segmentation object, which causes interactions to take different paths in the strategy based on screening rules. Figure 220 shows the Screen Segmentation properties dialog box.



Screen segmentation properties

General

□ Screen by variable

Language: English

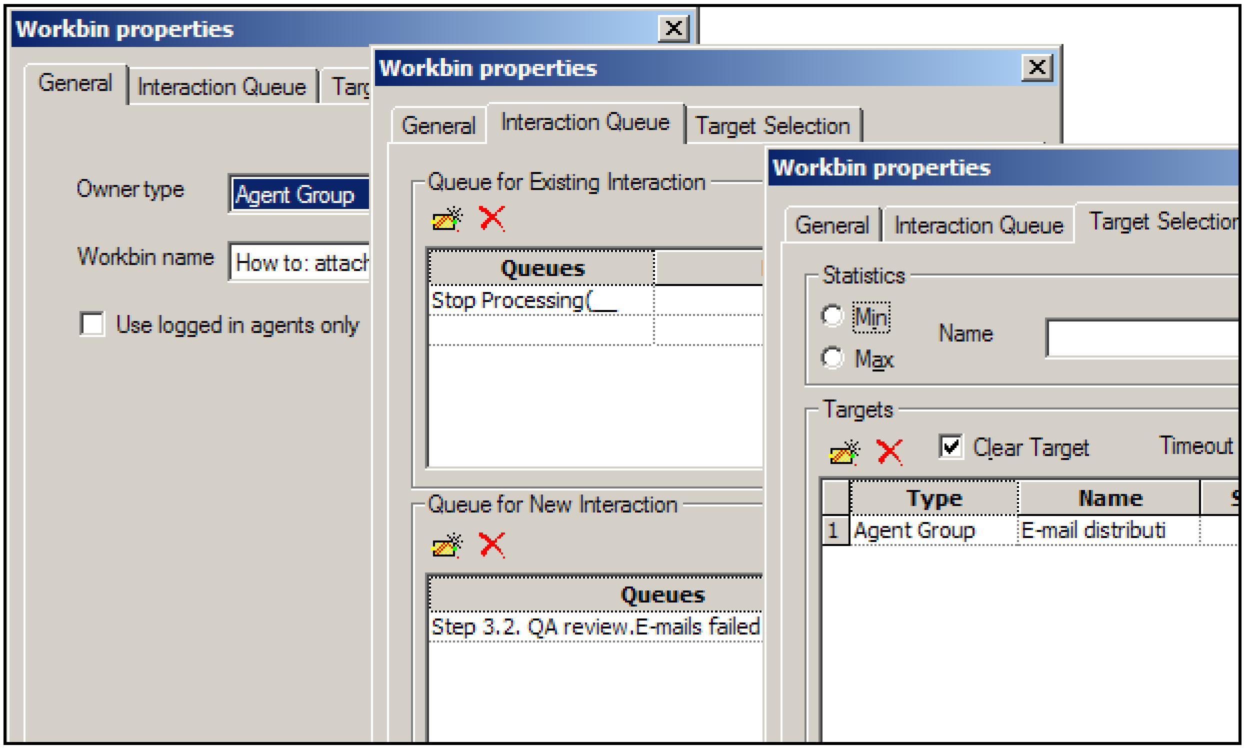
| Segment | Expression |
|---|---|
| 1 | Unidentified transaction |
| 2 | Wrong transaction amount |

**Figure 220: Screen Segmentation Properties Dialog Box**

Each segment in the Screen Segmentation properties dialog box generates its own output port in the strategy.

- In the strategy in Figure 218 on page 184, interactions screened as `Unidentified transactions` (needing an agent response instead of a Standard Response) go a Workbin object. This is represented as the `E-mail distribution` Workbin strategy-linked node in Figure 217 on page 183.

  Figure 221 shows the various tabs in the Workbin properties dialog box.



Owner type: Agent Group

Workbin name: How to: attach

□ Use logged in agents only

Queue for Existing Interaction

| Queues |
|---|
| Stop Processing(__ |

Queue for New Interaction

| Queues |
|---|
| Step 3.2. QA review.E-mails failed |

Statistics

○ Min ○ Max Name

Targets

☑ Clear Target Timeout

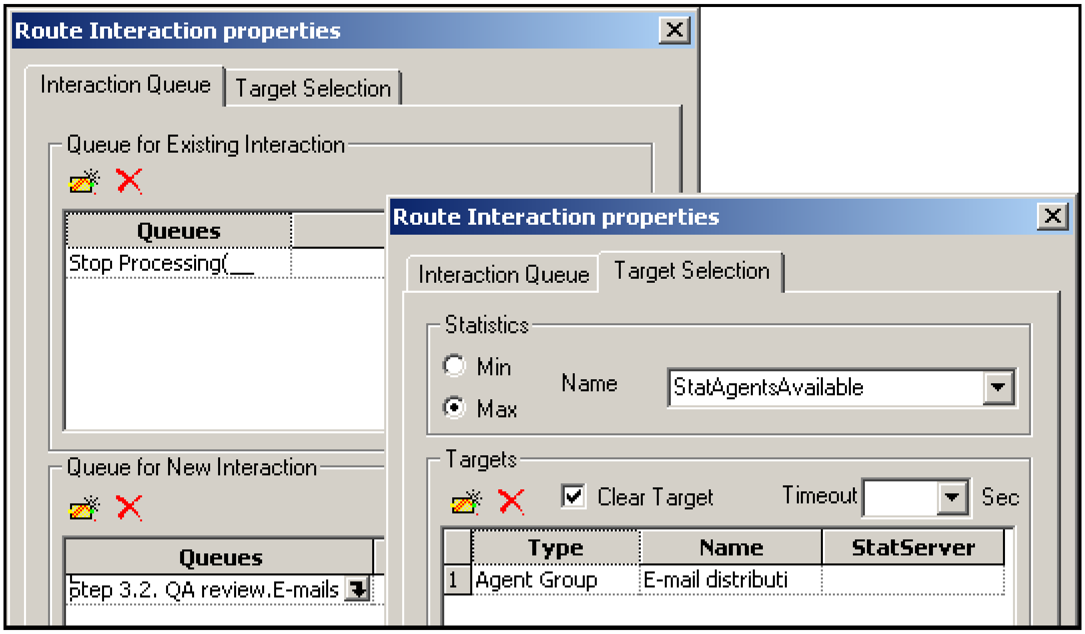
| | Type | Name |
|---|---|---|
| 1 | Agent Group | E-mail distributi |

**Figure 221: Workbin Properties Dialog Box**

- In the Target Selection tab, an Agent Group workbin named E-mail Distribution is selected. This is where the customer's e-mail will reside while a response is constructed.
- In the Interaction Queue tab, under Queue for New Interaction, the selection is E-mails failed QA found in the Step 3.2 QA Review business process (see page 126). This is where the agent places the new interaction (the e-mail response to customer) so it can be checked by QA prior to sending. This is represented by the E-mails failed QA strategy-linked queue node in Figure 217 on page 183.
- The Stop Processing_) queue (a Genesys predefined queue) is represented by the Stop strategy-linked node in Figure 217 on page 183. It indicates a notification to Interaction Server that processing of the existing interaction has stopped.

- Interactions matching the screening rule Wrong transaction amount go to a Route Interaction object.

  Figure 222 shows the Route Interaction properties dialog box.



Route Interaction properties

Interaction Queue | Target Selection

Queue for Existing Interaction

| Queues |
|---|
| Stop Processing(__ |

Queue for New Interaction

| Queues |
|---|
| Step 3.2. QA review.E-mails |

Route Interaction properties

Interaction Queue | Target Selection

Statistics

○ Min
● Max

Name: StatAgentsAvailable

Targets

☑ Clear Target Timeout Sec

| | Type | Name | StatServer |
|---|---|---|---|
| 1 | Agent Group | E-mail distributi | |

**Figure 222: Route Interaction Properties Dialog Box**

- In the Target Selection tab, an Agent Group named `E-mail distribution` is selected. This target is also represented as the `E-mail distribution` strategy-linked node in Figure 217 on page 183.
- In the `Interaction Queue` tab, under `Queue for New Interaction`, the selection is `E-mails failed QA`. This is the same queue used by the Workbin object (see Figure 221 on page 187). This is where the agent places the new interaction (the e-mail response to the customer) so it can be checked by QA prior to sending. This is also represented by the `E-mails failed QA` strategy-linked queue node in Figure 217 on page 183.
- The `Stop Processing_)` queue (a Genesys predefined queue) is represented by the Stop strategy-linked node in Figure 217 on page 183. It indicates a notification to Interaction Server that processing of the existing interaction has stopped.

## Error Handling

The following objects in Figure 217 on page 183 are the result of error handling:

- `E-mail service failure analysis` subroutine (see Figure 169 on page 144).
- `Terminate interaction` subroutine (similar to the subroutine in Figure 169 on page 144).

# How To: Place the Interaction Into the Workbin

This business process, listed in Figure 88 on page 80, demonstrates how to route an interaction to the original agent. If the original agent is not immediately available, the strategy places the interaction in the original agent's workbin for later handling.

**Note:** If you need a review of workbins, see *Universal Routing 8.1 Business Process User's Guide*.

Figure 223 shows the `How to: Place the Interaction Into the Workbin` business process.

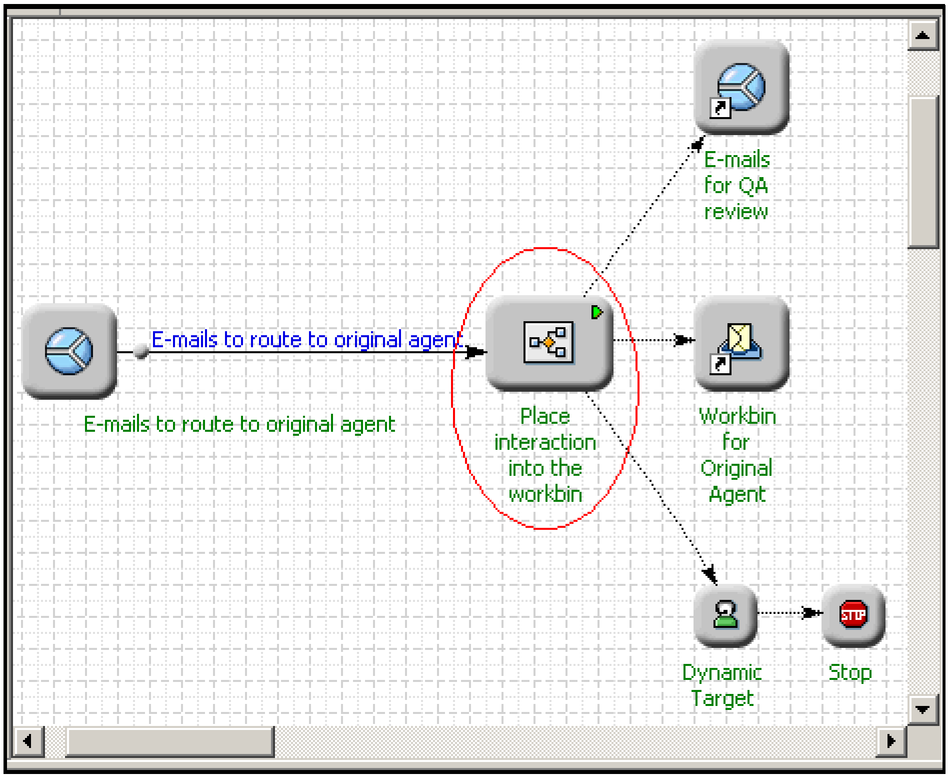

E-mails to route to original agent
E-mails to route to original agent
Place interaction into the workbin
E-mails for QA review
Workbin for Original Agent
Dynamic Target
Stop
STOP

**Figure 223: How to: Place Interaction Into Workbin Business Process**

## Processing Objects

This section describes the various objects in Figure 223.

A view (`E-mails to route to original agent`), with no conditions or order-by information, is attached to a queue (`E-mails to route to original agent`). It extracts interactions and sends them to the strategy `Place Interaction into the workbin` (see Figure 224).

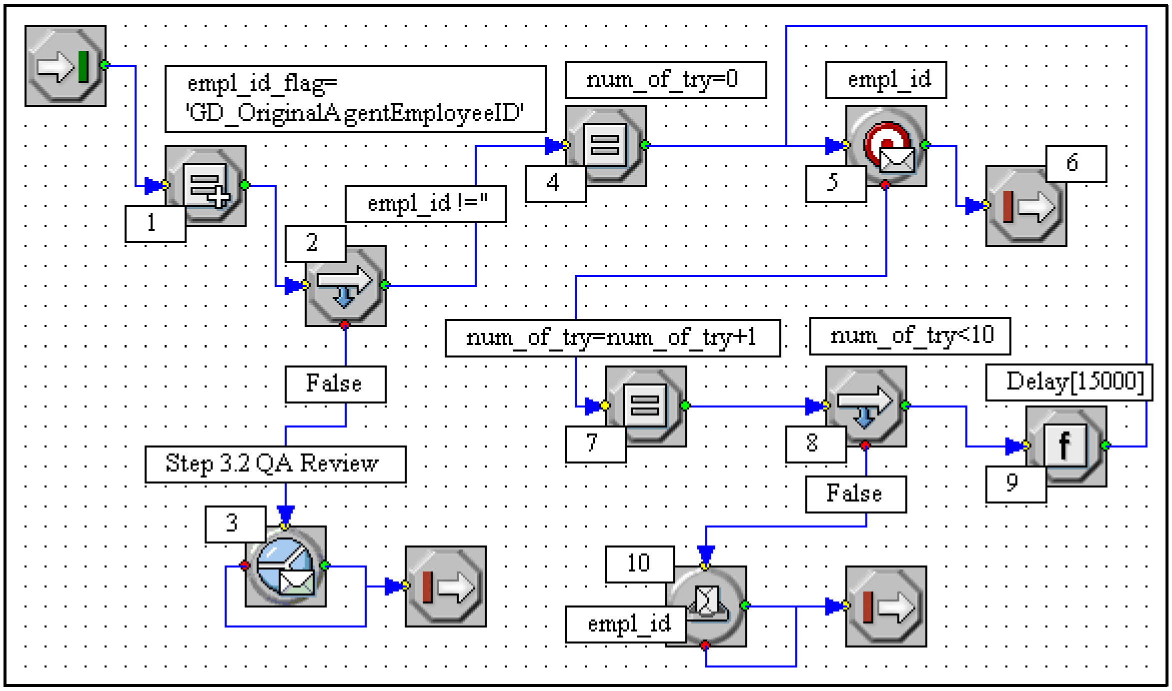


**Figure 224: Place Interaction into the Workbin Strategy**

This strategy assumes the original agent employee ID is contained in the interaction attached data. It is very similar to the strategy documented in "Routing E-mails To the Original Agent" on page 93.

The general processing flow is as follows:

**Note:** The numbers in the IRD objects in Figure 224 are keyed to the numbers below.

1. A MultiAssign object assigns interaction attributes to three predefined variables (see Table 3).

### Table 3: Place Interaction Into Predefined Variables

| Variable | Description |
|---|---|
| empl_id. | The Udata function attempts to get a value from the attached data key empl_id_flag and write it to this variable. If this key contains a value, it indicates there is an original agent for routing. |

**Table 3: Place Interaction Into Predefined Variables (Continued)**

| Variable | Description |
|---|---|
| empl_id_flag. | The original agent employee ID number. `GD_OriginalAgentEmployeeID` is an attached data key that may contain the identifier. |
| num_of_tries | This value is initialized to `0`. It functions as a counter used to control the number of tries the strategy makes to route to the original agent. |

2. After setting the variables, an If object initializes the `empl_id` variable so that it does not contain a value.

3. If the expression in the If object is true—that is, `empl_id` does *not* contain a value indicating there is no original agent, the e-mail goes out the red port to a Queue Interaction object specifying the `E-mails for QA review` queue (strategy-linked node in Figure 223 on page 189. This is used for a collaboration reply e-mail from one agent to another.

4. If the expression is false (if there is an original agent), the e-mail goes out the green port of the If object to an Assign object, which sets a counter (variable `num_of_try`) to zero.

5. A Route Interaction object attempts to route the interaction to the original agent contained in the `empl_id` variable. This target is represented by the `Dynamic target` strategy-linked node in Figure 223 on page 189.

6. If the routing is successful, the strategy has achieved its goal and an Exit object is connected to the green port of the Route Interaction object.

7. If the agent cannot accept the interaction for some reason, the e-mail goes out the red port of the Route Interaction object to an Assign object, which increments the `num_of_try` counter.

8. The e-mail goes out the green port of the Assign object to an If object, which is used to decide whether the number of processing cycles to determine the original agent is 10 or fewer.

9. If the number of tries to route to the original agent is less than `10,` the interaction goes out the green port of the If object to a Function object to delay strategy execution.

   Once the delay is established, the e-mail goes out the green port of the Function object back to the Route Interaction object to attempt to route to the original agent.

10. If the expression is false (that is, the number of tries is `10` or greater), the e-mail goes out the red port of the If object to a Workbin object (`Workbin for original agent` strategy-linked node in Figure 223 on page 189). This workbin is associated with the agent identified in the `empl_id variable`.

# How To: Screen Multiple Rules and Use Screening Switch

This business process demonstrates how to screen incoming interactions using multiple screening rules by using the MultiScreen object.

**Note:** MultiScreen differs from Screen in that it supplies additional return options and does not require a conditional test to determine whether a match occurred.

The strategy also demonstrates how to segment interactions to take different paths based on screening rules.

Figure 225 shows the `How to: Screen Multiple Rules and Use Screening Switch` business process.

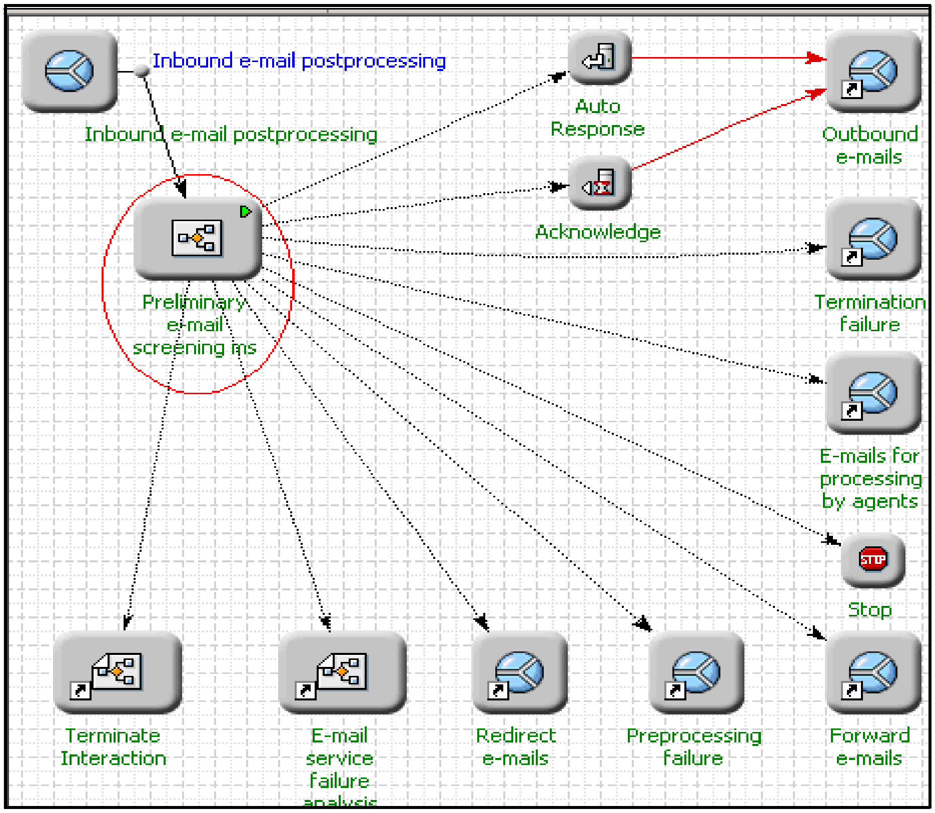

Inbound e-mail postprocessing
Inbound e-mail postprocessing
Auto Response
Outbound e-mails
Acknowledge
Preliminary e-mail screening ms
Termination failure
E-mails for processing by agents
Stop
Terminate Interaction
E-mail service failure analysis
Redirect e-mails
Preprocessing failure
Forward e-mails

**Figure 225: How to: Screen Using Multiple Rules and Using Screening Switch BP**

# Processing Objects

This section describes the various objects in Figure 225.

A view (`Inbound e-mail postprocessing`) with no conditions or order-by information attached to a queue (`Inbound e-mail postprocessing`) extracts interactions and sends them to the strategy `Preliminary e-mail screening ms`. Figure 226 shows the top part of the strategy.

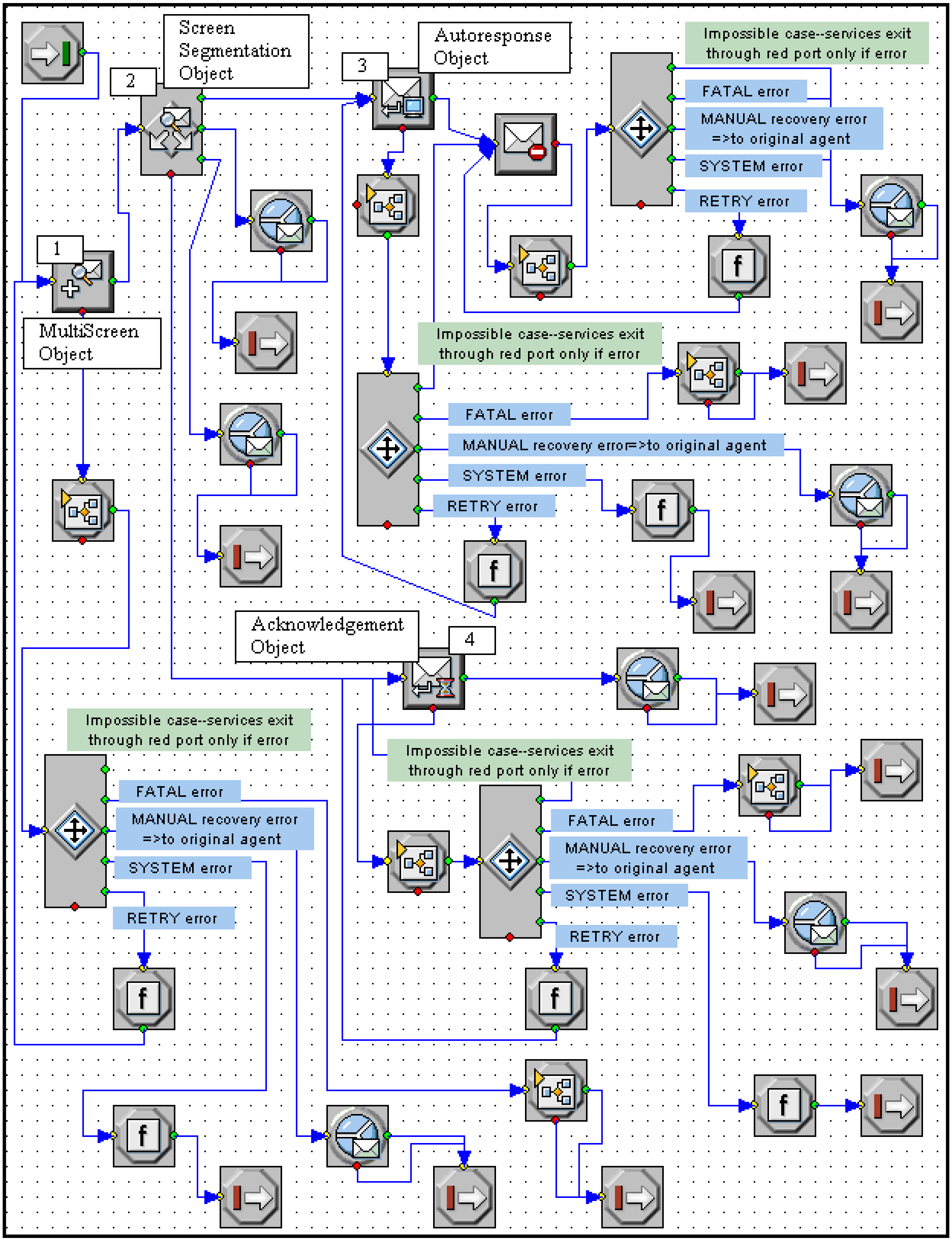

Screen Segmentation Object
Autoresponse Object
Impossible case--services exit through red port only if error
2
3
FATAL error
MANUAL recovery error =>to original agent
SYSTEM error
RETRY error
1
MultiScreen Object
Impossible case--services exit through red port only if error
FATAL error
MANUAL recovery error=>to original agent
SYSTEM error
RETRY error
Acknowledgement Object
4
Impossible case--services exit through red port only if error
FATAL error
MANUAL recovery error =>to original agent
SYSTEM error
RETRY error
Impossible case--services exit through red port only if error
FATAL error
MANUAL recovery error =>to original agent
SYSTEM error
RETRY error

**Figure 226: Preliminary E-mail Screening MS Strategy**

**Note:** *MS* at the end of the strategy name stands for *MultiScreen*.

1. This strategy uses the MultiScreen object and three different screening rules (`Auto Response Available`, `Warranty Problem`, and `Tech Support`) to screen inbound e-mails for words or word patterns.
2. A Screen Segmentation object connected to the green output port of the MultiScreen object causes interactions with different screening results to take different paths in the strategy.

## Screening #1

The purpose of the initial screening is determine whether the e-mail can be responded to with a Standard Response.

3. If yes, an Autoresponse object generates a Standard Response (see the `Autoresponse` strategy-linked node in Figure 225 on page 192). A Stop Interaction object (see Figure 225 on page 192) notifies Interaction Server that processing of this interaction is finished (see `Stop` strategy-linked node in Figure 225 on page 192).

## Screening #2

If a Standard Response cannot be used, a second screening determines whether the e-mail contains text that indicating it is from a customer with a warranty problem and therefore should be placed in a queue for forwarding.

If yes, the e-mail is placed in that queue (see the `Forward e-mails` queue in Figure 225 on page 192). A Stop Interaction object (see Figure 225 on page 192) notifies Interaction Server that processing of this interaction is finished (see `Stop` strategy-linked node in Figure 225 on page 192).

## Screening #3

If the second screening indicates a warranty problem is not involved, the e-mail goes through a third screening to determine if it contains text indicating it is from a customer with a Technical Support problem and therefore should be placed in a queue for redirecting.

If yes, the e-mail is placed in a queue for redirecting (see the `Redirect e-mails` queue in Figure 225 on page 192). A Stop Interaction object notifies Interaction Server that processing of this interaction is finished (see `Stop` strategy-linked node in Figure 225 on page 192).

## Stage 4—Acknowledgement

If the third screening does not produce results, an Acknowledgement object generates an acknowledgement Standard Response. The e-mail then goes to a queue for processing by agents (see the `E-mails for processing by agents` queue in Figure 225 on page 192).

### Error Handling

The following objects in the business process in Figure 225 on page 192 are the result of error handling:

- `Preprocessing failure` queue.
- `E-mail service failure analysis` subroutine (see Figure 116 on page 104).
- `Terminate interaction` subroutine.
- `Termination failure` queue.

## MultiScreen Versus Screen

This strategy is the functional equivalent of the `Preliminary e-mail screening` strategy discussed on “Screening of Inbound E-mails” on page 99.

The difference between the two is as follows:

- `Preliminary e-mail screening` uses three separate Screen objects. After each screen object, the strategy uses a If object and performs a conditional test to determine whether a screening rule match occurred. This adds a level of complexity and generates additional code. Figure 228 on page 198 shows the three Screen objects and If objects in the strategy.

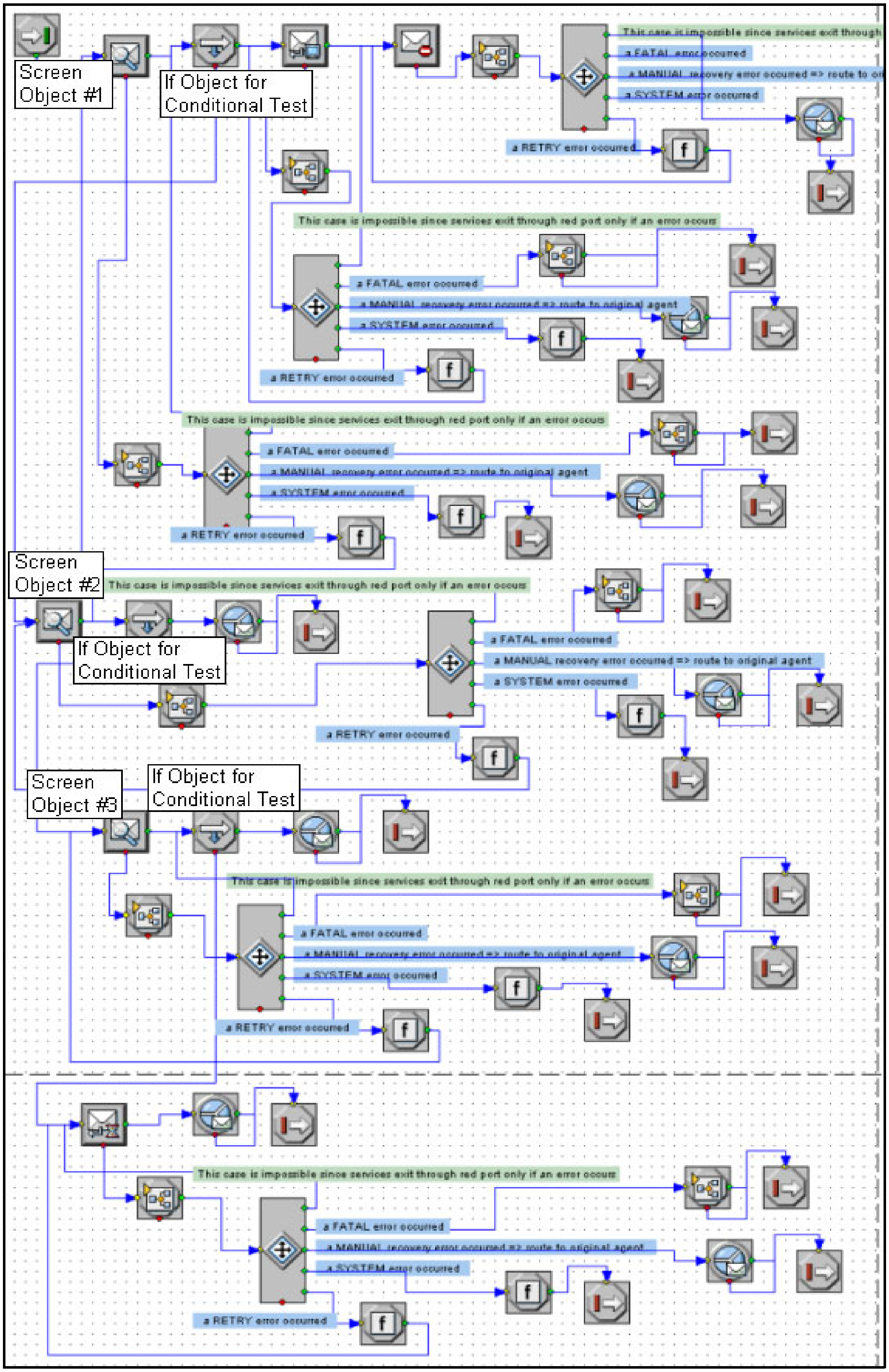
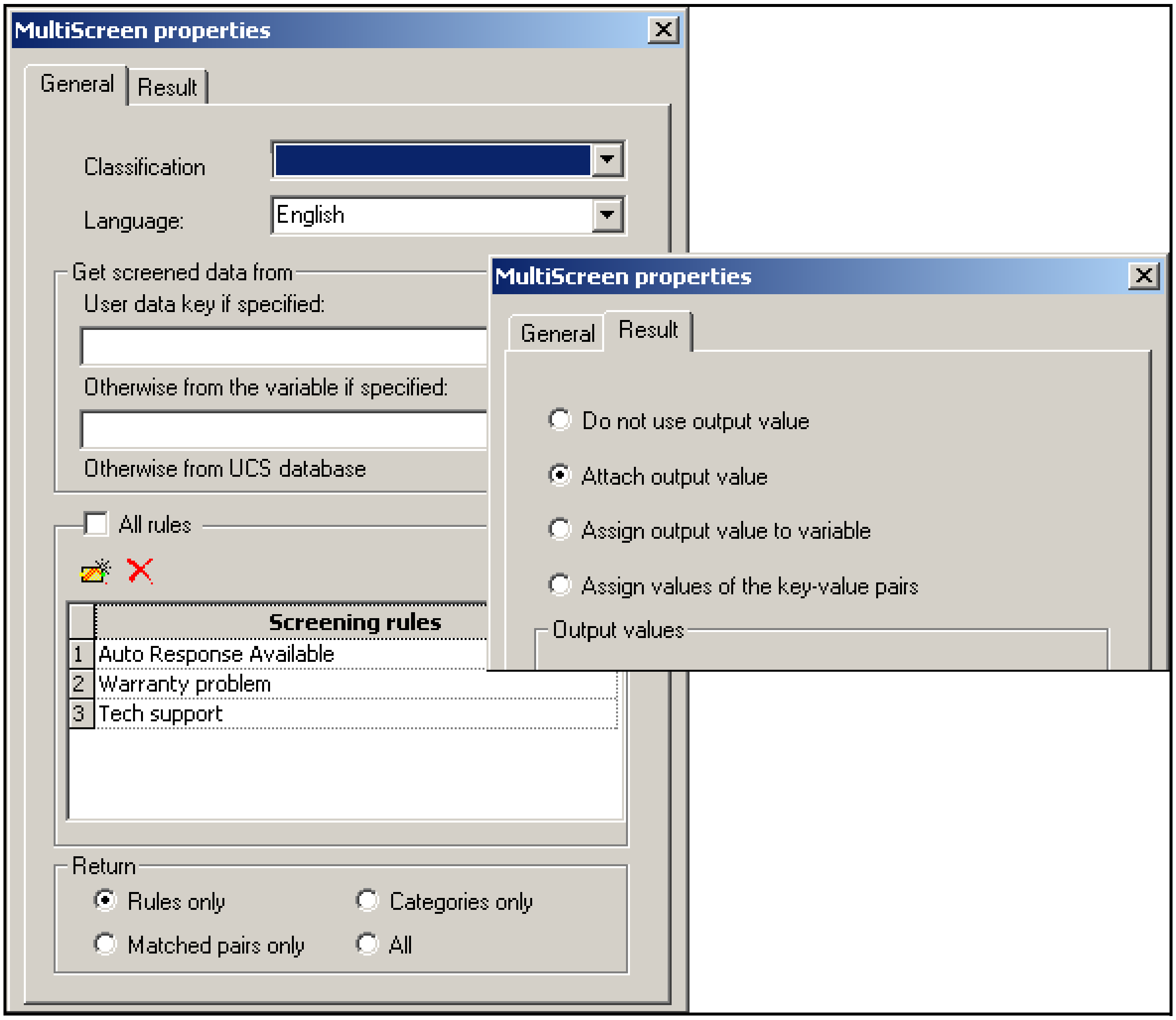

Screen Object #1
If Object for Conditional Test
This case is impossible since services exit through
a FATAL error occurred
a MANUAL recovery error occurred => route to o
a SYSTEM error occurred
a RETRY error occurred
This case is impossible since services exit through red port only if an error occurs
a FATAL error occurred
a MANUAL recovery error occurred => route to original agent
a SYSTEM error occurred
a RETRY error occurred
This case is impossible since services exit through red port only if an error occurs
a FATAL error occurred
a MANUAL recovery error occurred => route to original agent
a SYSTEM error occurred
a RETRY error occurred
Screen Object #2
This case is impossible since services exit through red port only if an error occurs
If Object for Conditional Test
a FATAL error occurred
a MANUAL recovery error occurred => route to original agent
a SYSTEM error occurred
a RETRY error occurred
Screen Object #3
If Object for Conditional Test
This case is impossible since services exit through red port only if an error occurs
a FATAL error occurred
a MANUAL recovery error occurred => route to original agent
a SYSTEM error occurred
a RETRY error occurred
This case is impossible since services exit through red port only if an error occurs
a FATAL error occurred
a MANUAL recovery error occurred => route to original agent
a SYSTEM error occurred
a RETRY error occurred

**Figure 227: Preliminary E-Mail Screening Strategy**

Compare the strategy in Figure 227 on page 197 with the strategy `Preliminary e-mail screening ms`, shown in Figure 226 on page 194, which uses the MultiScreen object. Strategy complexity and the amount of byte code are reduced because a single object handles multiple rules and does not require a conditional test after each screening. Figure 228 shows the properties dialog box for the MultiScreen object.



MultiScreen properties

General | Result

Classification:

Language: English

Get screened data from

User data key if specified:

Otherwise from the variable if specified:

Otherwise from UCS database

□ All rules

| | Screening rules |
|---|---|
| 1 | Auto Response Available |
| 2 | Warranty problem |
| 3 | Tech support |

Return

● Rules only ○ Categories only

○ Matched pairs only ○ All

MultiScreen properties

General | Result

○ Do not use output value

● Attach output value

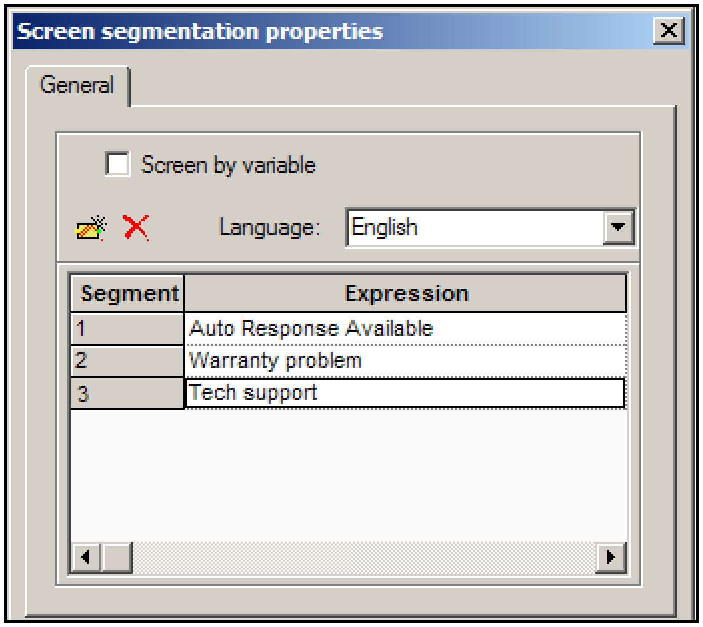
○ Assign output value to variable

○ Assign values of the key-value pairs

Output values

**Figure 228: MultiScreen Object Properties Dialog Box**

Figure 229 shows the properties dialog box for the Screen Segmentation object that is connected to the exit port of the MultiScreen object.



☐ Screen by variable

Language: English

| Segment | Expression |
|---|---|
| 1 | Auto Response Available |
| 2 | Warranty problem |
| 3 | Tech support |

Figure 229: Screen Segmentation Properties Dialog Box

Each segment in Figure 229 generates its own output port (see the Screen Segmentation object in the full strategy, shown in Figure 226 on page 194).

# Identifying a Contact and Creating an Interaction

The business process `How to: Identify Contact and Create Interaction` contains the strategy `Identify Contact and Create Interaction` described in this section (see Figure 230).

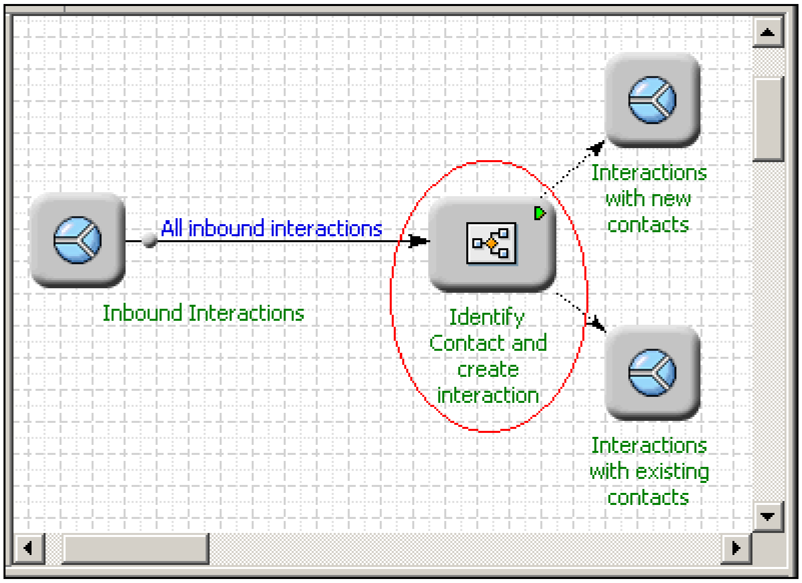

All inbound interactions
Inbound Interactions
Identify Contact and create interaction
Interactions with new contacts
Interactions with existing contacts

**Figure 230: How to: Identify Contact and Create Interaction Business Process**

## Identify Contact and Create Interaction Strategy

The `Inbound Interactions` queue in Figure 230 receives all inbound interactions. A submitter object submits them to the `Identify Contact and Create Interaction` strategy shown in Figure 231.

Figure 231 shows the `Identify Contact and Create Interaction` strategy in the IRD Routing Design window.

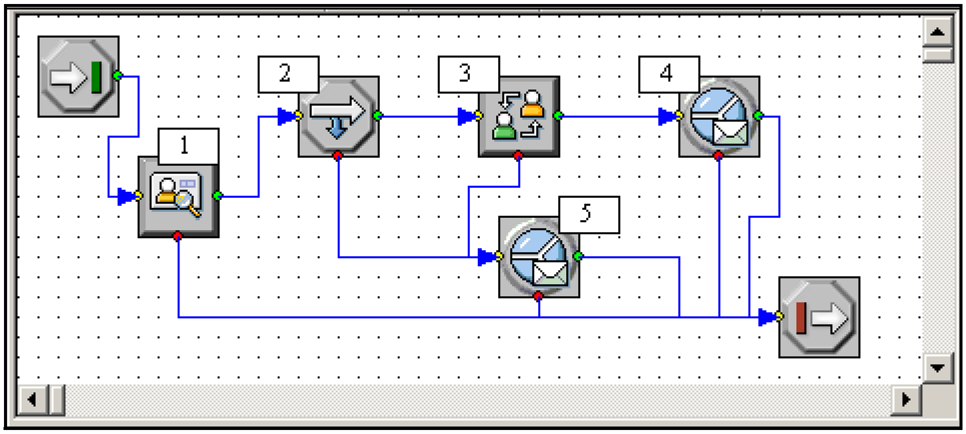
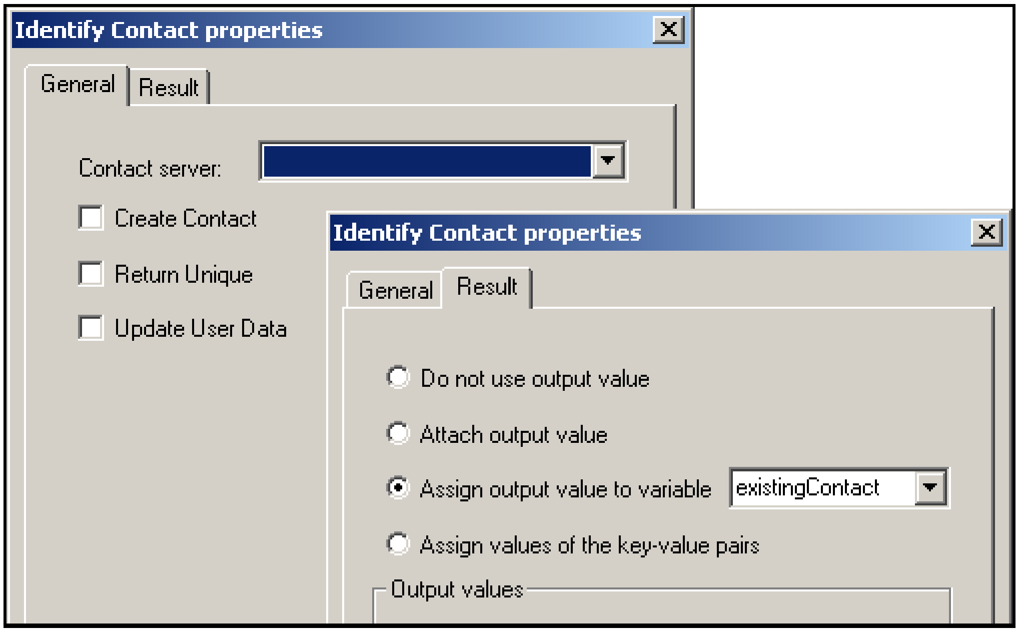


**Figure 231: Identify Contact and Create Interaction Strategy**

## Summary of Flow

**Note:** The IRD objects that are described this section are keyed to the numbers in Figure 231 on page 200. The description excludes descriptions of some error processing objects.

1. The Identify Contact object tries to match data attached to the interaction with that in the Universal Contact Server database to determine whether the customer is already entered as a contact in the database or is a new customer. It then assigns the result to the variable `existingContact` (see Figure 232).



Contact server:

□ Create Contact

□ Return Unique

□ Update User Data

○ Do not use output value

○ Attach output value

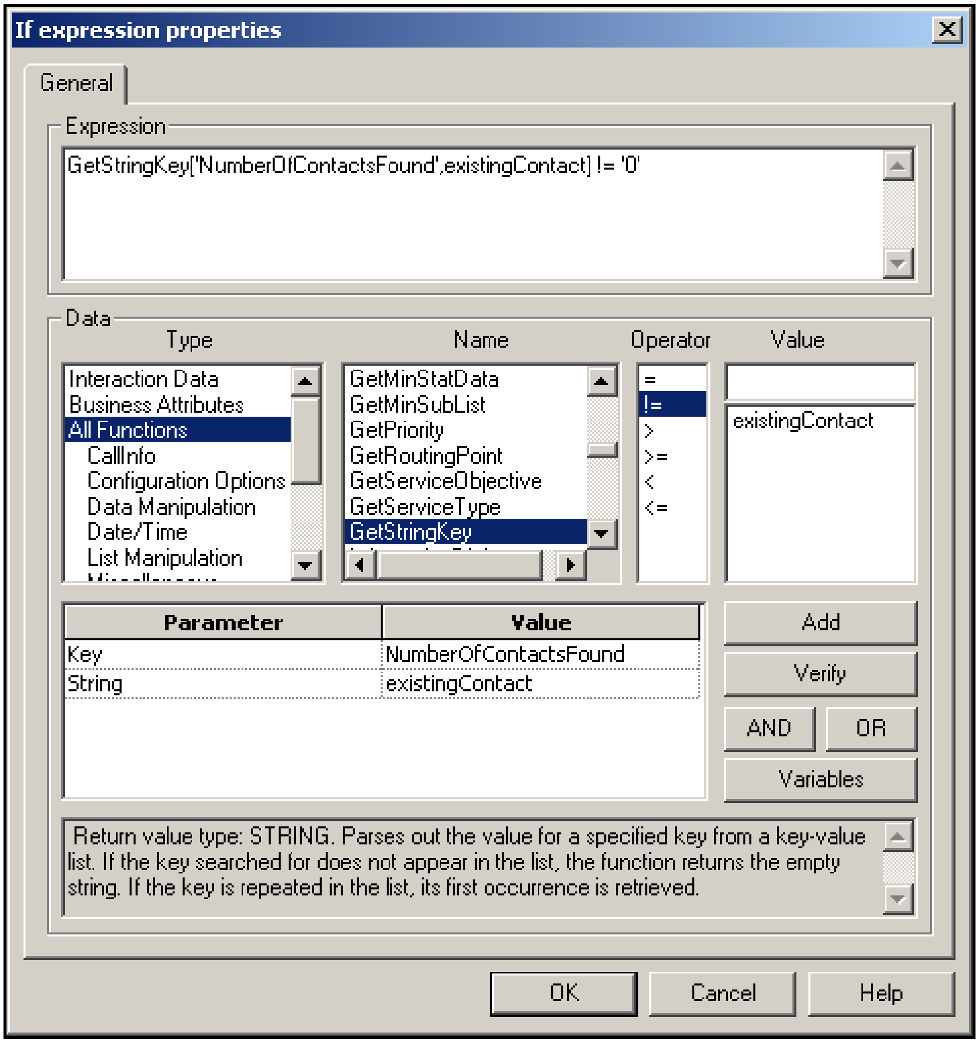
● Assign output value to variable existingContact

○ Assign values of the key-value pairs

Output values

**Figure 232: Identify Contact Object Properties Dialog Box**

2. The interaction then proceeds to an If object, shown in Figure 233, which analyzes the `existingContact` variable to determine whether the returned value indicates a new contact (no matches in the UCS database) or an existing one (one or more matches).



GetStringKey['NumberOfContactsFound',existingContact] != '0'

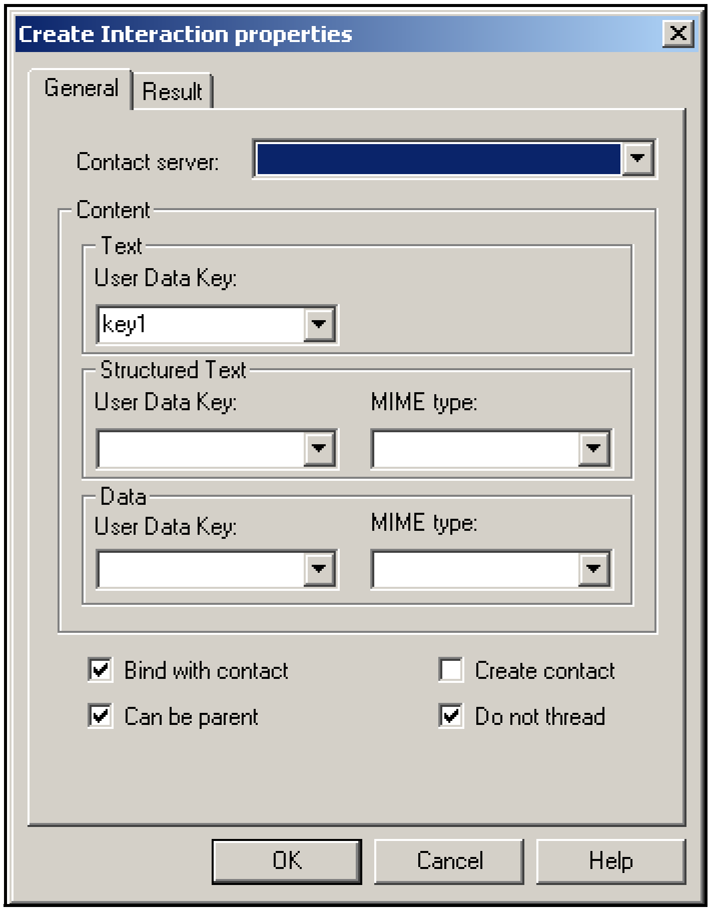
| Parameter | Value |
|---|---|
| Key | NumberOfContactsFound |
| String | existingContact |

Return value type: STRING. Parses out the value for a specified key from a key-value list. If the key searched for does not appear in the list, the function returns the empty string. If the key is repeated in the list, its first occurrence is retrieved.

**Figure 233: If Object Determines Whether Contact Exists in UCS Database**

Interactions from new contacts go directly to the `Interactions with new contacts` queue (see Step 5). Interactions with existing contacts are next processed by a Create Interaction object.

3. The Create Interaction object creates a record for this interaction in the UCS database (see Figure 234). The information entered in the various fields—such as first and last name, account number, email address, and so on—are taken from the user data attached to the interaction.



Contact server:

Content

Text

User Data Key: key1

Structured Text

User Data Key:

MIME type:

Data

User Data Key:

MIME type:

☑ Bind with contact

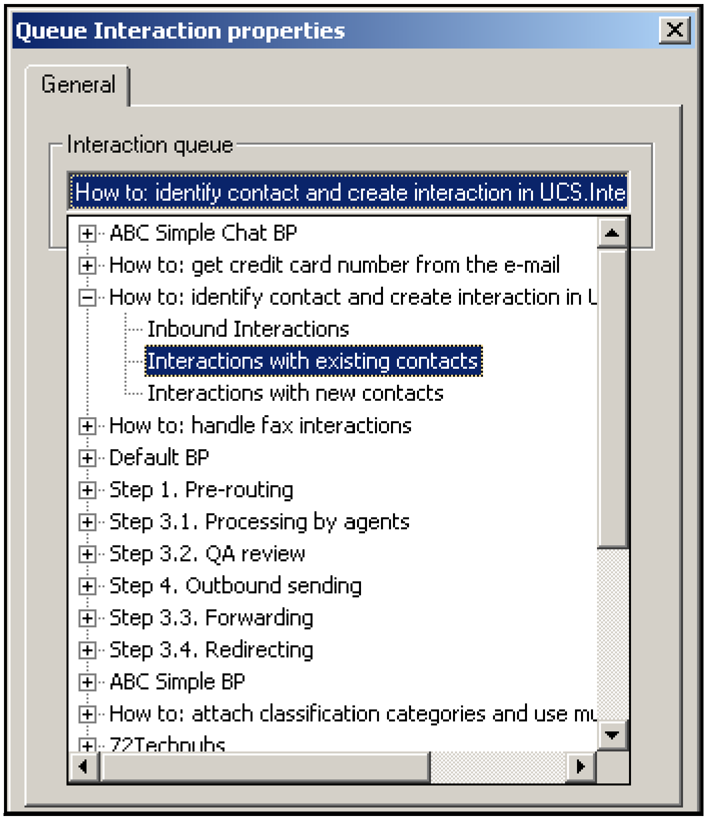
☐ Create contact

☑ Can be parent

☑ Do not thread

**Figure 234: Create Interaction Object**

4. Interactions from existing customers are then processed by a Queue Interaction object, which sends them to a queue for existing contact handling (see Figure 235).



Queue Interaction properties

General

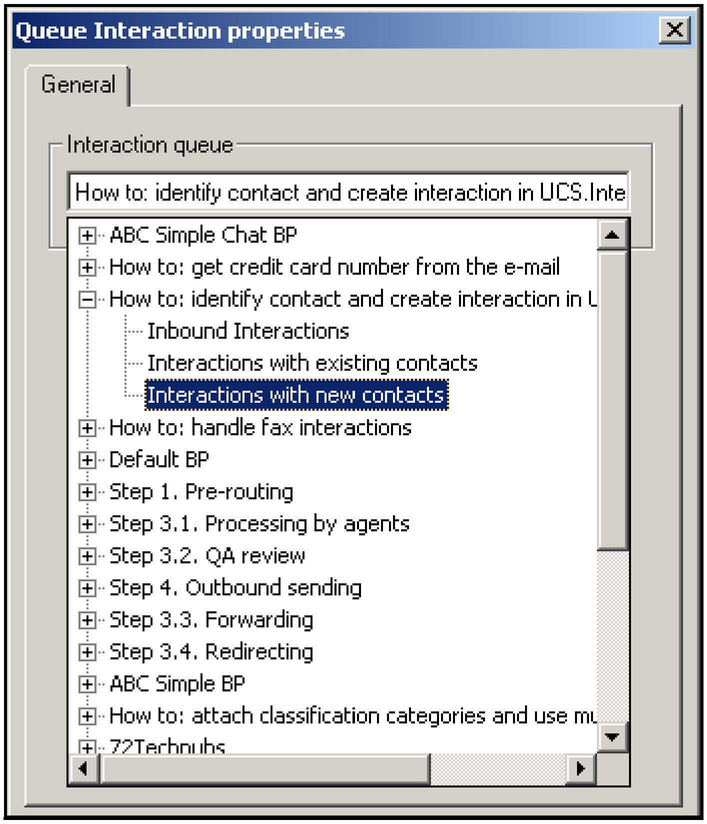
Interaction queue

How to: identify contact and create interaction in UCS.Inte

- ABC Simple Chat BP
- How to: get credit card number from the e-mail
- How to: identify contact and create interaction in U
  - Inbound Interactions
  - Interactions with existing contacts
  - Interactions with new contacts
- How to: handle fax interactions
- Default BP
- Step 1. Pre-routing
- Step 3.1. Processing by agents
- Step 3.2. QA review
- Step 4. Outbound sending
- Step 3.3. Forwarding
- Step 3.4. Redirecting
- ABC Simple BP
- How to: attach classification categories and use mu
- 72Techpubs

**Figure 235: Queue Interaction Object—Existing Contacts**

5. No interaction is created in the UCS database for interactions from new customers. Such interactions are next handled by a Queue Interaction object that sends them to a queue dedicated to handling interactions from new contacts (see Figure 236).



Queue Interaction properties

General

Interaction queue

How to: identify contact and create interaction in UCS.Inte

- ABC Simple Chat BP
- How to: get credit card number from the e-mail
- How to: identify contact and create interaction in U
  - Inbound Interactions
  - Interactions with existing contacts
  - Interactions with new contacts
- How to: handle fax interactions
- Default BP
- Step 1. Pre-routing
- Step 3.1. Processing by agents
- Step 3.2. QA review
- Step 4. Outbound sending
- Step 3.3. Forwarding
- Step 3.4. Redirecting
- ABC Simple BP
- How to: attach classification categories and use mu
- 72Techpubs

**Figure 236: Queue Interaction Object—New Contacts**

# Screening a Fax Interaction

The `How to: Handle Fax Interactions` business process receives interactions that arrived in the form of faxes and have been translated into a format usable by Genesys interaction handling. The `How to: Handle Fax Interactions` business process is shown in Figure 237.

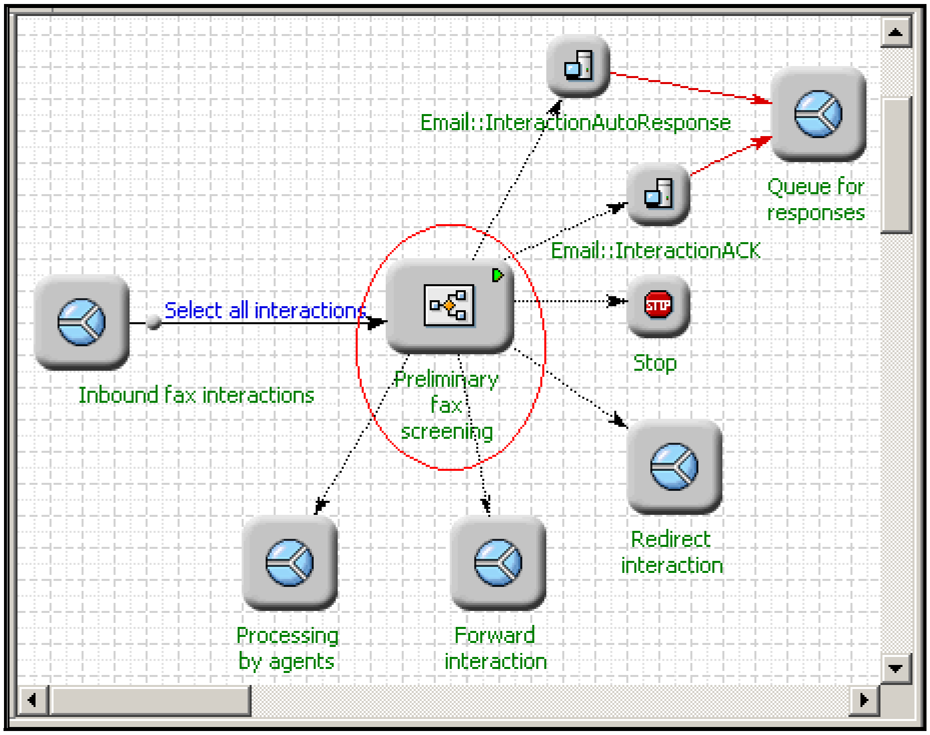


**Figure 237: How to: Handle Fax Interactions Business Process**

The `Inbound Fax Interactions` queue in Figure 237 receives all inbound fax interactions. A submitter object submits them to the `Preliminary Fax Screening` strategy shown in Figure 238.

This strategy is the same as the `Preliminary e-mail screening ms` strategy (shown in Figure 226 on page 194), but generalized to handle all interaction types, no only e-mail.

Figure 238 shows the `Preliminary Fax Screening` strategy in the IRD Routing Design window. The `Preliminary Fax Screening` strategy is shown without any of the error handling objects.

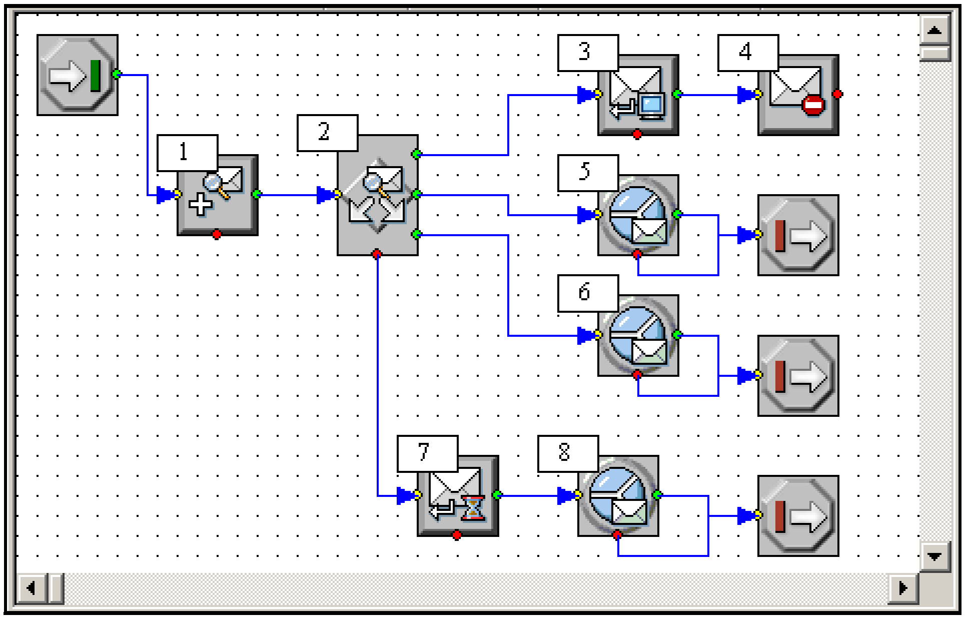
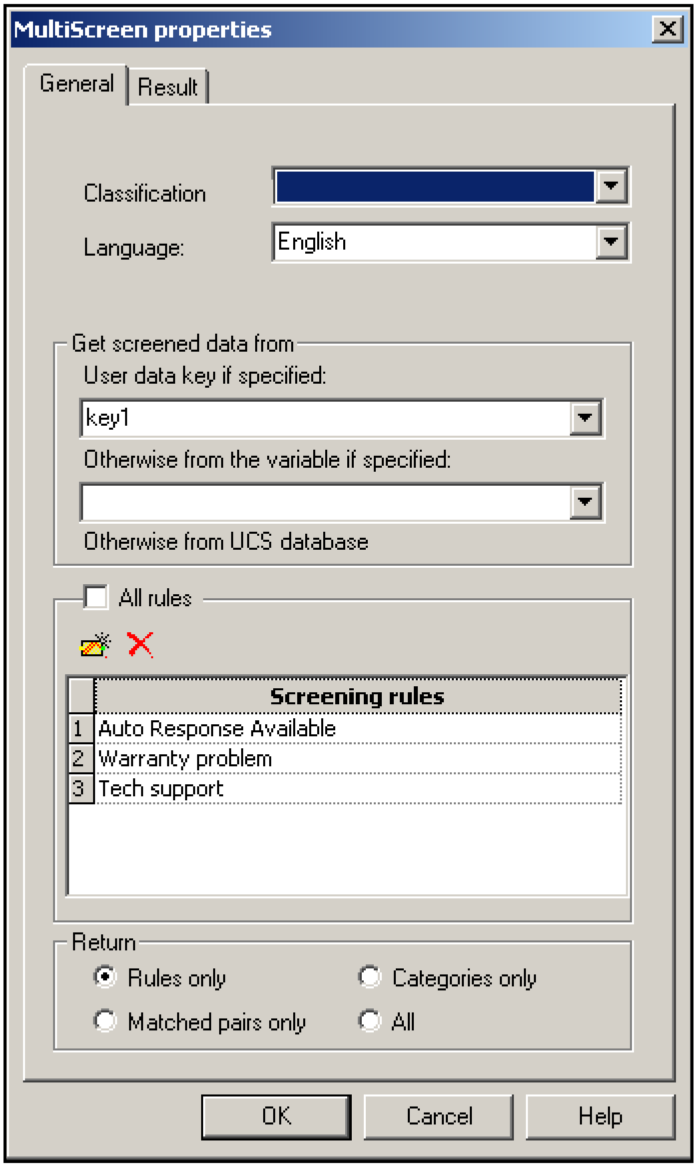


**Figure 238: Preliminary Fax Screening Strategy**

## Summary of Flow

**Note:** The IRD objects that are described this section are keyed to the numbers in Figure 238. The discussion excludes error processing objects.

1. The MultiScreen object takes interaction information from a user data key, `key1`. It instructs Classification Server to perform a screen using three screening rules and then attaches the results to the interaction (see Figure 239).



Classification

Language: English

Get screened data from

User data key if specified: key1

Otherwise from the variable if specified:

Otherwise from UCS database

☐ All rules

|  | Screening rules |
|---|---|
| 1 | Auto Response Available |
| 2 | Warranty problem |
| 3 | Tech support |

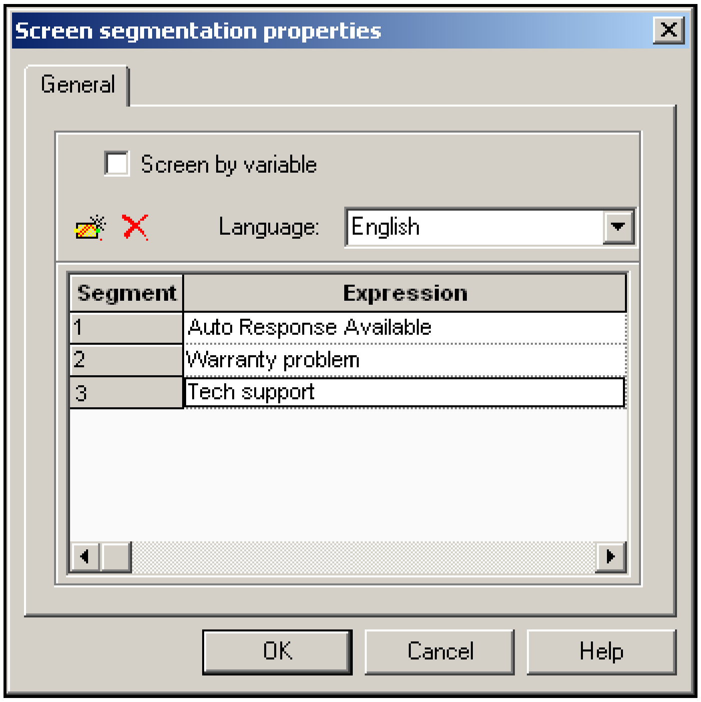
Return

● Rules only ○ Categories only

○ Matched pairs only ○ All

**Figure 239: Using the MultiScreen Object to Screen a Fax**

2. The Screen segmentation object (shown in Figure 240) evaluates the results returned by Classification Server and directs the interaction accordingly. There are four possible outcomes:
   - The interaction can be handled by an autoresponse message. If so, the interaction goes to Step 3.
   - The interaction is a warranty issue. It then goes to Step 5.
   - The interaction is a tech support issue. It then goes to Step 6.
   - The interaction cannot be classified. The interaction exits through the red error port and proceeds to Step 7.



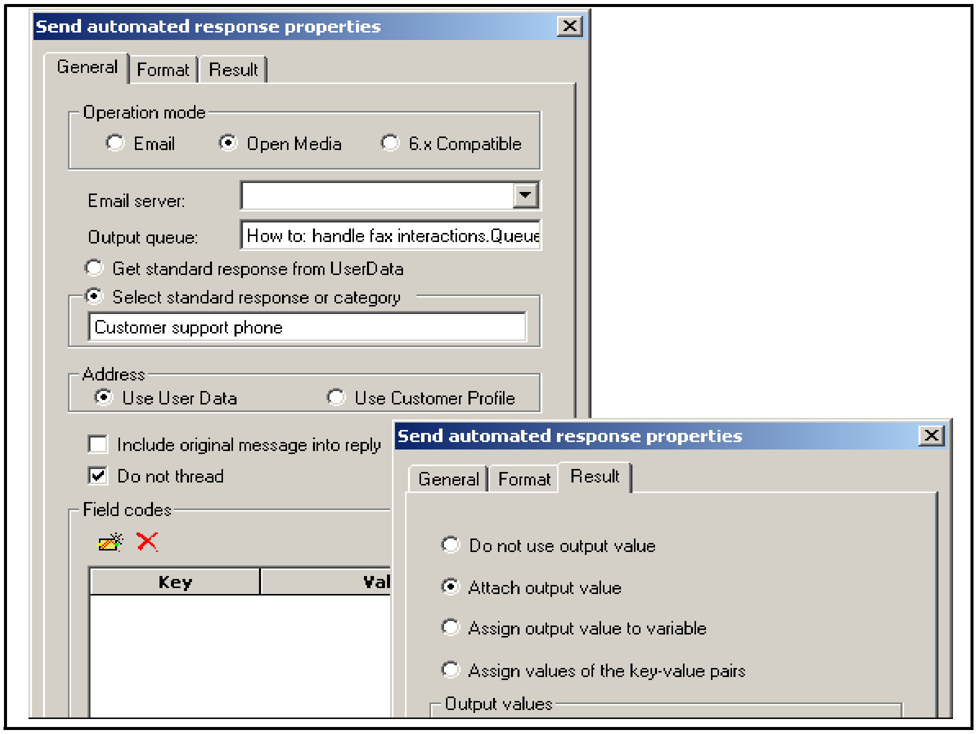
□ Screen by variable

Language: English

| Segment | Expression |
|---|---|
| 1 | Auto Response Available |
| 2 | Warranty problem |
| 3 | Tech support |

**Figure 240: Screen Segmentation Object—Screening a Fax**

3. If the screening determines that the fax can be answered with a response from the standard response library, the interaction proceeds to an Autoresponse object (see Figure 241).


Send automated response properties
General
Format
Result
Send automated response properties
General
Format
Result

Operation mode
○ Email ● Open Media ○ 6.x Compatible

Email server:

Output queue: How to: handle fax interactions.Queue

○ Get standard response from UserData

● Select standard response or category

Customer support phone

Address
● Use User Data ○ Use Customer Profile

□ Include original message into reply

☑ Do not thread

Field codes

| Key | Val |
|---|---|

○ Do not use output value

● Attach output value

○ Assign output value to variable

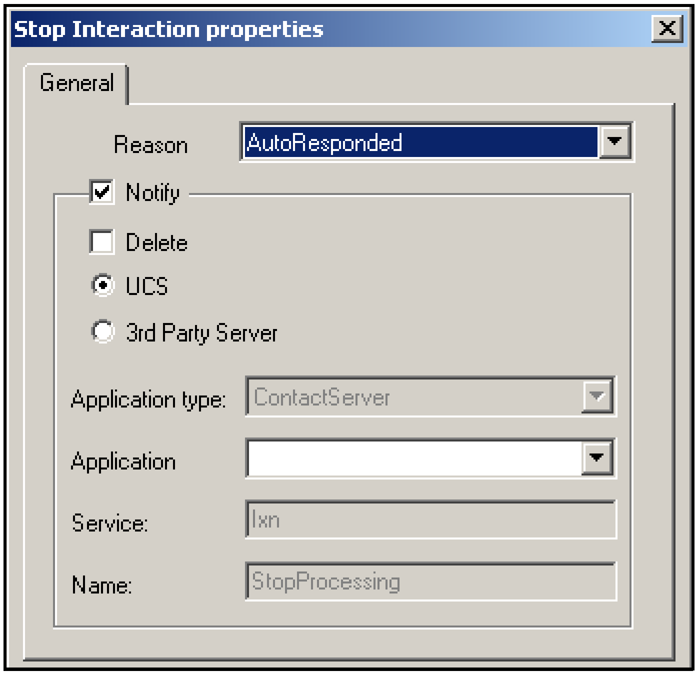
○ Assign values of the key-value pairs

Output values

**Figure 241: Sending Autoresponse E-mail in Response to Fax**

Although the incoming interaction was in the form of a fax, the autoresponse is an e-mail. The Open Media radio button is selected to indicate that the original interaction was of a medium other than e-mail. The autoresponse e-mail is sent to the specified queue. The current interaction continues to a Stop Interaction object (Step 4).

4. The Stop Interaction object (shown in Figure 242) terminates interaction processing and notifies UCS of the fact, including the Autoresponded reason code in the notification message.



Reason: AutoResponded

☑ Notify

☐ Delete

● UCS

○ 3rd Party Server

Application type: ContactServer

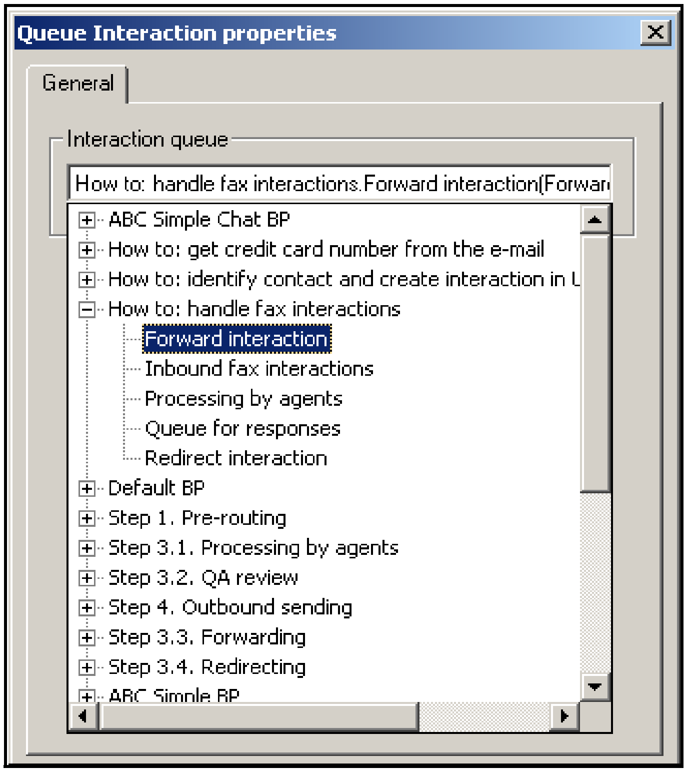
Application:

Service: Ixn

Name: StopProcessing

**Figure 242: Stop Interaction Object—Autoresponded**

5. If the screening determines that the interaction content concerns a warranty issue, the interaction is handled by a Queue Interaction object, which directs the interaction to a queue that forwards the interaction appropriately (see Figure 243).



Queue Interaction properties

General

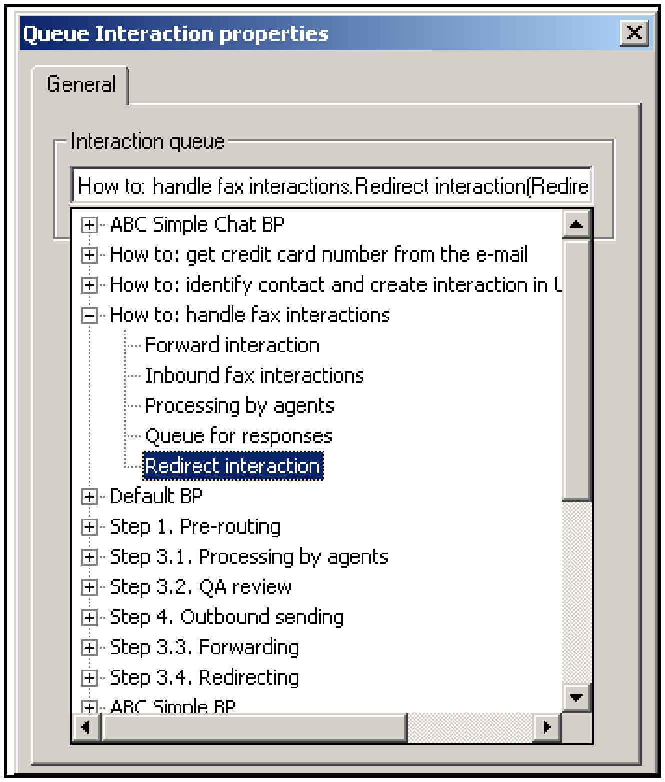
Interaction queue

How to: handle fax interactions.Forward interaction(Forwar

**Figure 243: Queue Interaction Object—Queue for Forwarding Faxes**

Once the interaction is placed in the Forward Interaction queue, it has left this strategy and is handled by a different strategy, possibly similar to "Forwarding an E-mail" on page 114.

**6.** If the screening determines that the interaction content concerns a tech support issue, the interaction is handled by a Queue Interaction object, which directs the interaction to a queue that redirects it appropriately (see Figure 244).



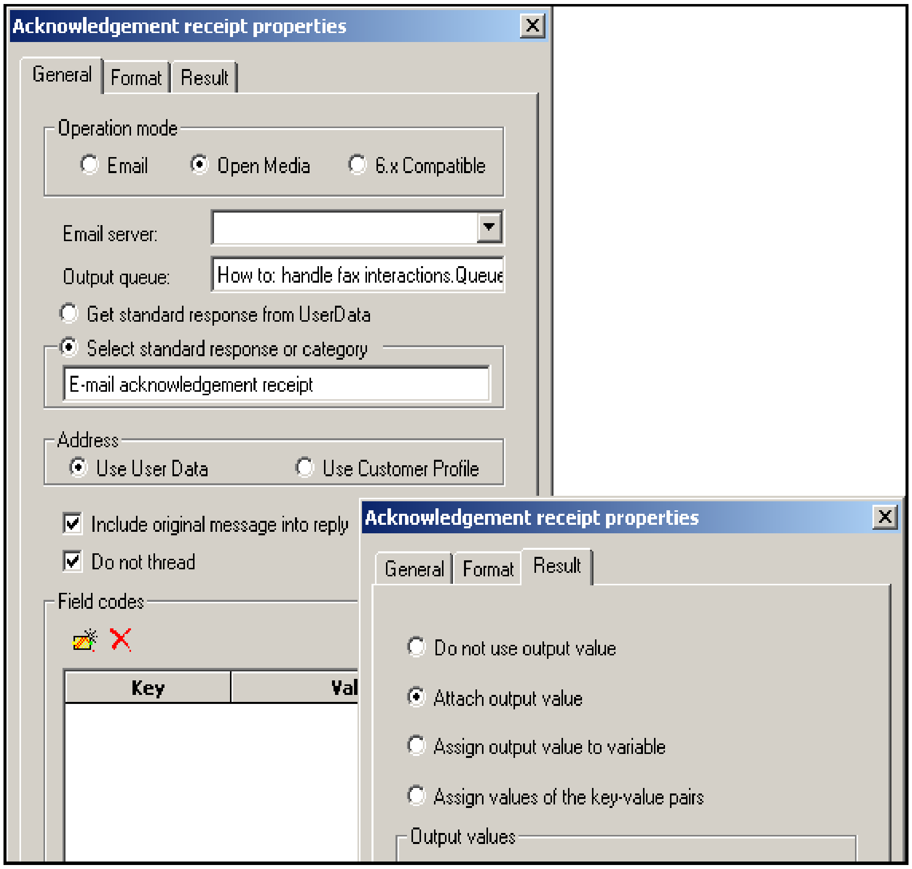
How to: handle fax interactions.Redirect interaction(Redire

- ABC Simple Chat BP
- How to: get credit card number from the e-mail
- How to: identify contact and create interaction in L
- How to: handle fax interactions
  - Forward interaction
  - Inbound fax interactions
  - Processing by agents
  - Queue for responses
  - Redirect interaction
- Default BP
- Step 1. Pre-routing
- Step 3.1. Processing by agents
- Step 3.2. QA review
- Step 4. Outbound sending
- Step 3.3. Forwarding
- Step 3.4. Redirecting
- ABC Simple BP

**Figure 244: Queue Interaction Object—Queue for Redirecting Faxes**

Once the interaction is placed in the Redirect Interaction queue, it has left this strategy and is handled by a different strategy, possibly one similar to “Redirecting an E-mail” on page 110.

7. If the interaction does not contain content that matches any of the screening rules, it exits through the Screen Segmentation object’s red error port and proceeds to an Acknowledgement object (see Figure 245).



Acknowledgement receipt properties

General | Format | Result

Operation mode: ○ Email ● Open Media ○ 6.x Compatible

Email server:

Output queue: How to: handle fax interactions.Queue

○ Get standard response from UserData

● Select standard response or category: E-mail acknowledgement receipt

Address: ● Use User Data ○ Use Customer Profile

☑ Include original message into reply

☑ Do not thread

Field codes

| Key | Val |
|---|---|

Acknowledgement receipt properties

General | Format | Result

○ Do not use output value

● Attach output value

○ Assign output value to variable

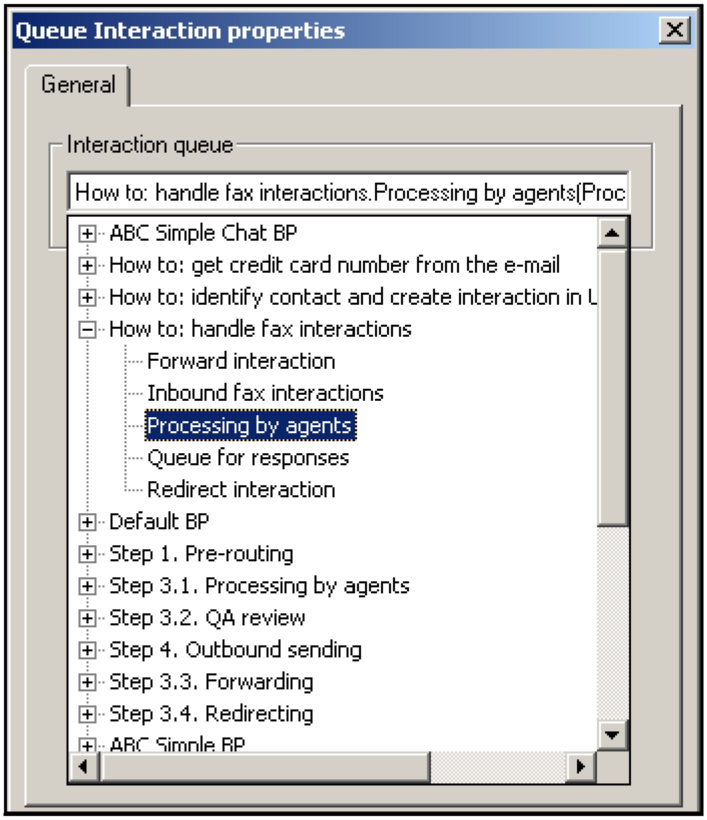
○ Assign values of the key-value pairs

Output values

**Figure 245: Acknowledgement Object—Sending Acknowledgement of Fax Received**

This object is configured similarly to the Autoresponse object in Step 3. It also returns an e-mail to the customer in response to the incoming open media interaction (in this sample, a fax). In this case, however, it lets the customer know that the fax was received and will be processed.

8. The interaction then goes to a Queue Interaction object which places the interaction in a queue for agent processing (see Figure 246).



Queue Interaction properties

General

Interaction queue: How to: handle fax interactions.Processing by agents(Proc

**Figure 246: Queue Interaction Object—Faxes for Agent Processing**

Again, this queue marks the exit point of the interaction from this strategy. The interaction is now handled according to the workflow you determined in the business process that includes this strategy.

# Related Documentation Resources

The following resources provide additional information that is relevant to this software. Consult these additional resources as necessary.

## Universal Routing

- *Universal Routing 8.1 Reference Manual,* which explains routing strategies, IRD objects, functions and options for Universal Routing Server and other servers, number translation, pegs, and the statistics used for routing.
- *Universal Routing 8.1 Business Process User's Guide*. This guide contains step-by-step instructions for creating interaction workflows (*business processes*), which direct incoming customer interactions through various processing objects in order to generate an appropriate response for the customer.
- *Universal Routing 8.1 Deployment Guide*. The first part of the guide provides information you will need to get started: a high-level overview of Universal Routing features and functions, including product architecture, system availability, redundancy information and deployment-planning. The second part of the guide provides instructions for deploying Universal Routing components, and describes how to start and stop these components once you have configured and installed them.
- *Universal Routing 8.1 Interaction Routing Designer Help,* which describes how to use Interaction Routing Designer to create routing strategies and how to use the Interaction Design window to create business processes.

## eServices/eServices

- *eServices (eServices) 8.0 Deployment Guide,* which includes a high-level overview of features and functions of Genesys eServices together with architecture information and deployment-planning materials. It also introduces you to some of the basic concepts and terminology used in this product.
- *eServices (eServices) 8.0 User's Guide,* which provides information and recommendations on the use and operation of Genesys eServices (formerly eServices).
- Universal Routing Log Events and eServices (eServices) Log Events in *Framework 8.0 Combined Log Events Help,* which is a comprehensive list and description of all events that may be recorded in Management Layer logs.

## Genesys

- *Genesys Technical Publications Glossary,* which ships on the Genesys Documentation Library DVD and which provides a comprehensive list of the Genesys and computer-telephony integration (CTI) terminology and acronyms used in this document.
- *Genesys Migration Guide*, which ships on the Genesys Documentation Library DVD, and which provides documented migration strategies for Genesys product releases. Contact Genesys Technical Support for more information.
- *Genesys Events and Models Reference Manual*, which presents a set of basic interaction models, showing the components involved and the messaging (requests and events) sent among them.
- Release Notes and Product Advisories for this product, which are available on the Genesys Technical Support website at `http://genesyslab.com/support`.

Information about supported hardware and third-party software is available on the Genesys Technical Support website in the following documents:

- *Genesys Supported Operating Environment Reference Manual*
- *Genesys Supported Media Interfaces Reference Manual*

For additional system-wide planning tools and information, see the release-specific listings of System Level Documents on the Genesys Technical Support website, accessible from the `system level documents by release` tab in the Knowledge Base `Browse Documents` Section.

Genesys product documentation is available on the:

- Genesys Technical Support website at `http://genesyslab.com/support`.
- Genesys Documentation Library DVD, which you can order by e-mail from Genesys Order Management at `orderman@genesyslab.com`.

# Document Conventions

This document uses certain stylistic and typographical conventions—introduced here—that serve as shorthands for particular kinds of information.

## Document Version Number

A version number appears at the bottom of the inside front cover of this document. Version numbers change as new information is added to this document. Here is a sample version number:

80fr_ref_06-2008_v8.0.001.00

You will need this number when you are talking with Genesys Technical Support about this product.

## Screen Captures Used in This Document

Screen captures from the product graphical user interface (GUI), as used in this document, may sometimes contain minor spelling, capitalization, or grammatical errors. The text accompanying and explaining the screen captures corrects such errors *except* when such a correction would prevent you from installing, configuring, or successfully using the product. For example, if the name of an option contains a usage error, the name would be presented exactly as it appears in the product GUI; the error would not be corrected in any accompanying text.

## Type Styles

Table 4 describes and illustrates the type conventions that are used in this document.

**Table 4: Type Styles**

| Type Style | Used For | Examples |
|---|---|---|
| Italic | • Document titles<br>• Emphasis<br>• Definitions of (or first references to) unfamiliar terms<br>• Mathematical variables<br>Also used to indicate placeholder text within code samples or commands, in the special case where angle brackets are a required part of the syntax (see the note about angle brackets on page 218). | Please consult the *Genesys Migration Guide* for more information.<br>Do *not* use this value for this option.<br>A *customary and usual* practice is one that is widely accepted and used within a particular industry or profession.<br>The formula, $x +1 = 7$ where $x$ stands for . . . |

### Table 4: Type Styles (Continued)

| Type Style | Used For | Examples |
|---|---|---|
| Monospace font<br>(Looks like `teletype` or `typewriter text`) | All programming identifiers and GUI elements. This convention includes:<br>• The *names* of directories, files, folders, configuration objects, paths, scripts, dialog boxes, options, fields, text and list boxes, operational modes, all buttons (including radio buttons), check boxes, commands, tabs, CTI events, and error messages.<br>• The values of options.<br>• Logical arguments and command syntax.<br>• Code samples.<br>Also used for any text that users must manually enter during a configuration or installation procedure, or on a command line. | Select the `Show variables on screen` check box.<br>In the `Operand` text box, enter your formula.<br>Click `OK` to exit the `Properties` dialog box.<br>T-Server distributes the error messages in `EventError` events.<br>If you select `true` for the `inbound-bsns-calls` option, all established inbound calls on a local agent are considered business calls.<br>Enter `exit` on the command line. |
| Square brackets ([ ]) | A particular parameter or value that is optional within a logical argument, a command, or some programming syntax. That is, the presence of the parameter or value is not required to resolve the argument, command, or block of code. The user decides whether to include this optional information. | `smcp_server -host [/flags]` |
| Angle brackets (< >) | A placeholder for a value that the user must specify. This might be a DN or a port number specific to your enterprise.<br>**Note:** In some cases, angle brackets are required characters in code syntax (for example, in XML schemas). In these cases, italic text is used for placeholder values. | `smcp_server -host <confighost>` |

# Index