

Aspect Uniphi Suite v6.1

# eFlow Developer Guide

© Copyright 2004 by Aspect Communications Corporation. All rights reserved.

Aspect Communications Corporation is headquartered in San Jose, California.

Aspect Communications Limited is headquartered in Stockley Park, Uxbridge, Middlesex, United Kingdom.

Aspect Communications B.V. is headquartered in Hoofddorp, the Netherlands.

Aspect Communications GmbH is headquartered in Neu-Isenburg, Germany.

Aspect Communications (S) Pte Ltd. is headquartered in Singapore.

Aspect Communications (HK) Ltd. is headquartered in Wanchai, Hong Kong.

Aspect Communications Pty Ltd. is headquartered in Sydney, Australia.

Aspect Communications Japan Ltd. is headquartered in Tokyo, Japan.

Aspect Communications is headquartered in Paris, France.

Aspect Communications is headquartered in Markham, Ontario, Canada.

Aspect and the Aspect logo are trademarks and/or service marks of Aspect Communications Corporation in the United States and/or other countries. All other product or service names mentioned in this Guide may be trademarks of the companies with which they are associated.

# Contents

\_ \_ \_ \_ \_ \_ \_ \_

About This Guide	.vii
Who Should Use This Guide	. vii
How This Guide Is Organized	. vii
Where to Find Additional Information	.viii
Technical Support	. viii
Chapter 1: Overview	7
What Is the Aspect eFlow System?	7
eFlow System and Uniphi Suite	8
What Is an eFlow?	9
Introducing Aspect eBusiness Architect	. 10
Opening eBusiness Architect	. 11
Using the Step Palette	. 12
Using the Watch	. 13
Navigating Online Help	. 14
eFlow Configuration and Administration	. 14
Chapter 2: Creating eFlows	.17
Administrative Prerequisites	. 17
Building an eFlow	. 18
Starting a New eFlow	. 18
Opening an Existing eFlow	. 20
Adding Steps to the eFlow	. 22
Changing the Step Caption	. 23
Adjusting Step Width	. 23
Connecting Steps	. 23
Setting Step Properties	. 24
Creating eFlow Variables	. 25
Adding Comments to the eFlow	. 27
Using Labels in eFlows	. 27
Selecting Steps.	. 28
Rearranging Steps	. 29
Finding a Step	. 30

\_ \_

\_ \_ \_ \_

. . . . . . .

| |

Saving an eFlow
Saving an eFlow on the Server
Saving an eFlow Locally
eFlow Properties
Using Subflows
Debugging an eFlow
Creating Event Rules for eFlow Execution
Managing eFlows
Displaying eFlow Statistics
The Overlay Menu
Adjusting the Statistics Collection Period
General eFlow Statistics
Changing Statistics Display Colors
User Events and eFlows
Chapter 2: a Elow Stops Poference
General Steps 52
Aspect CMI Steps 53
Database Steps
Routing Steps 54
Media Control Steps 55
Advanced IVR Steps 55
Reporting Steps 56
Scheduler Steps
Agent Interaction Steps
System Interaction Steps
System Management Steps
Legacy Routing Steps
Chapter 4: Task-Oriented eFlow Samples 59
Data Access eFlows
Database Access
Data Repository Access
Private Data Access
Agent Task eFlows
Remote Log On
Remote Log Off
Media Task eFlows
Collecting Caller Responses
Kecording an Announcement
Advanced IVK eFlow
Chapter 5: Sample Routing eFlows 69
Basic Routing eFlows

Simple Push eFlow
Simple Pull eFlow
Push eFlow with Task Escalation72
Advanced Routing eFlows
A Simple RMS eFlow
Two-Tiered Agent Search eFlow76
Selecting an Agent by Extension
Select Agent by Extension eFlow (TAPI Switch)
Select Agent by Extension eFlow (Aspect Switch)
Select Agent Group eFlow
Skill-Based Routing
Chapter 6: Interswitch Routing
Interswitch Routing Concepts
Interswitch Output Parameters
Interswitch Input Parameters
Correlation Keys
Interswitch Example for Aspect Adapter
Sample Interswitch eFlow
Aspect ACD Configuration
Outbound CCTs
Inbound CCTs
Interswitch Example for Cisco CallManager
Sample Interswitch eFlow
Configuring Interswitch Routing for Cisco CallManager 100
Cisco CallManager Configuration
UnIphi Administrator Configuration
Gateway Configuration 102
Index

About This Guide

This guide provides information about the Aspect eFlow System and describes how to use Aspect eBusiness Architect to create and modify eFlows.

## Who Should Use This Guide

This guide is written for eFlow developers, system administrators, and other personnel who need to create, reconfigure, or maintain eFlows for use with Aspect Uniphi Suite.

Users of this guide should also:

- Be familiar with the general concepts of contact routing and workflow technology.
- Understand the basics of Aspect Uniphi Suite, as described in the Aspect Uniphi Suite Product Reference Guide.
- Be familiar with the Uniphi Suite System Administrator software, as described in the *Aspect Uniphi Suite System Administrator Guide*.

## How This Guide Is Organized

This guide is organized as follows:

- Chapter 1, Overview, introduces eFlows, Architect, and eFlow System and describes how all three fit in to the Uniphi Suite product.
- Chapter 2, Creating eFlows, gives an overview of the process for creating eFlows to handle customer contacts.
- Chapter 3, eFlow Steps Reference, describes each of the eFlow steps available in eBusiness Architect.

- Chapter 4, Task-Oriented eFlow Samples, presents examples of eFlows used to perform specific tasks, such as logging on to Uniphi or recording an announcement.
- Chapter 5, Sample Routing eFlows, presents example eFlows used to route outside contacts to an agent or resource.
- Chapter 6, Interswitch Routing, describes how to set up call routing between two or more call centers, using Aspect Uniphi as an interswitch router.

# Where to Find Additional Information

For a list of all the documents provided with Uniphi Suite, see the Documentation Roadmap file located on the *Aspect Uniphi Suite v6.0 Documentation CD*.

For the latest versions of the documentation, access the following Web site: http://www.aspect.com/services/login.cfm. You must have a valid Aspect support contract to log on to this site.

# **Technical Support**

Aspect provides technical support under the Aspect Support Agreement. If you have a question or problem that you are unable to resolve by reading the manual or online Help, call your Aspect representative.

When you call, have your documentation handy, and be prepared to provide the following information:

- Your name and address
- Your Aspect site number
- The version of the Aspect software you are running
- A concise, clear description of the problem, including any error messages that appeared, and of the actions you were taking when you encountered the problem



# Overview

Aspect eFlow System is the component of Aspect Uniphi Suite that stores and executes eFlows. Aspect eBusiness Architect is the application used to create and modify eFlows. This chapter introduces eFlows, Architect, and eFlow System and describes how all three fit in to the Uniphi Suite product. It contains the following sections:

- What Is the Aspect eFlow System?
- What Is an eFlow?
- Introducing Aspect eBusiness Architect
- eFlow Configuration and Administration

## What Is the Aspect eFlow System?

If Uniphi Administrator is the brains of the Uniphi Suite product, Aspect eFlow System is the central nervous system: accepting input from various adapters, consulting with repositories and database servers, interacting with subsystems and incoming contacts, and routing tasks and data through adapters back to outside switches. Aspect eFlow System reaches out to every branch of the Uniphi Suite system. It is the web that weaves it all together.

There are three major components of the eFlow System:

- **eBusiness Architect** is the Windows application used to create and modify eFlows using a drag-and-drop graphic interface.
- The LDAP (Lightweight Directory Access Protocol) server stores eFlows and eFlow configuration information in an LDAP data store.
- The **eFlow Engine** is the system that executes eFlows stored in the LDAP data store, based on business rules set in the Uniphi Administrator.

Each of these components is configured in the Uniphi Administrator, where eFlow resources and connections are defined and rules for eFlow execution are set up (see eFlow Configuration and Administration on page 14).

The Aspect eFlow System can:

- Execute eFlows in response to events (such as a route request generated by an incoming call), based on rules set up in the Uniphi Administrator.
- Accept route requests from multiple ACDs and direct a call to any location within the enterprise.
- Access data stored in a customer database or data repository to determine how to best route an incoming contact.
- Apply the same business rules to all contact channels, blending voice, Web, and e-mail contacts.
- Deliver customer data along with the call to the agent's desktop.
- Collect, track, and distribute customer data among multiple switch systems, allowing different vendors to share information.
- Execute an eFlow based on a user-defined event, such as an agent clicking a button on a web page.
- Use the Resource Matching Service (RMS) option to route contacts to agents with specific skills, such as fluency in Spanish.

# eFlow System and Uniphi Suite

Aspect eFlow System is a part of the Uniphi Contact Center (UCC), which together with Uniphi Voice Portal (UVP) comprises the Aspect Uniphi Suite. UCC connects to one or more switch adapters, which allow eFlow System to communicate with outside switches. Some switches, such as the Aspect Call Center, connect to the Uniphi switch adapter through an Aspect CMI server. Other switches, such as TAPI, connect directly to the switch adapter.

eFlow System may also be connected (via UCC) to various media handlers, VXML servers, Web servers, reporting databases, customer databases, and repositories. Using eFlows created with eBusiness Architect, eFlow System can access and utilize all outside components available to Uniphi, merging them together seamlessly.

In a typical scenario, here's how Uniphi might use eFlow System to help route an incoming contact:

1. A contact arrives at a contact center. This could be a voice call through PSTN or IP PBX, an e-mail, or a Web contact.

- 2. The switch for the contact center forwards the contact information to a switch adapter (or CMI server/adapter pair), which generates a task ID for the contact and translates the contact data into a format that Uniphi can recognize.
- 3. While the contact waits on the switch, the switch adapter passes the task ID for the contact to the Uniphi Contact Center (UCC), along with any associated data.
- 4. The UCC generates an event requesting eFlow execution and presents the task to the Aspect eFlow System.
- 5. The eFlow System consults the rule set associated with that event (defined in the Uniphi Administrator) to determine which eFlow to execute for handling the task.
- 6. The eFlow Engine executes the appropriate eFlow.
- 7. The eFlow performs any number of functions. It could, for example:
  - Access data stored in a repository, an outside customer database, or from a private switch data field.
  - Enter data in a repository, database, or data field.
  - Utilize a switch adapter to play messages to callers and collect responses to prompts.
  - Send call data to a database used for reporting.
  - Connect Uniphi Suite with an Aspect Telephony Server and run a custom VXML script that performs speech recognition tasks.
  - Retrieve or change the current state of an agent.
  - Send requests to and retrieve messages from a Web server.
  - Construct and send an e-mail.
- 8. When the eFlow finishes, it usually returns routing information back through the adapter (or adapter/CMI server) to the switch.
- 9. The receiving switch routes the contact to an agent, based on routing information passed along from the eFlow System.

# What Is an eFlow?

An eFlow is a set of commands, called steps, that give the system instructions for carrying out a task. That task can be as simple as confirming an account number in a customer database or as complex as using skill-based routing parameters to direct a contact to the most appropriate agent.

This very basic eFlow example (Figure 1) finds an available agent for an incoming contact. It uses a database query to find the agent who has been available the longest, then reserves that agent and stores the agent ID in an eFlow variable that can be passed back to the switch adapter. The switch itself will make the actual connection between the contact and the agent.



Figure 1. Sample eFlow

Every eFlow is associated with an event, such as a routing request from the Uniphi Contact Center (UCC). When that event occurs, rules bound to the event determine which eFlow is executed. For example, when the UCC receives an incoming call from a PSTN switch, such as the Aspect Call Center, it presents a Route Request to the eFlow System. The Route Request event might have numerous sets of rules, or conditions, bound to it, each one pointing to a different eFlow. The first condition found to be true triggers the eFlow that is associated with it.

Each eFlow is also associated with a business application, such as Catalog Sales or Customer Support, for reporting purposes. All statistics collected for the eFlow, such as how many tasks the eFlow processed, are grouped according to business application.

Use the Uniphi Administrator to set up both event rules and business applications (refer to the *Aspect Uniphi Suite System Administrator Guide*). Use eBusiness Architect to create and modify eFlows.

## Introducing Aspect eBusiness Architect

Aspect eBusiness Architect is a visual design application used to create, modify, and debug eFlows. In Architect, the eFlow is presented graphically, in the form of a flowchart. Steps in the eFlow are executed sequentially, beginning with the Start step and ending with a Finish step.

#### **Opening eBusiness Architect**

To open the eBusiness Architect application, select **Start→Programs→Aspect Uniphi Suite 6.0→eBusiness Architect**.

The Architect application window appears (Figure 2).



Figure 2. eBusiness Architect Window

The Architect window is divided into the following sections:

- The top area contains the title bar, menu bar, and up to four toolbars. Select **View**→**Toolbars** to toggle the display of each toolbar.
- The *step palette* contains icons that represent steps for use in eFlows. Use these icons to add steps to the workspace to build an eFlow.
- The *workspace*, located in the center, displays the eFlow. This is where you arrange and connect steps and set step properties to orchestrate eFlow functions.
- The *watch*, located in the bottom portion of the window, is where information about the executed eFlow appears. It includes five tabs, which you can select to view associated variables and their contents, validation errors, and errors during debugging.

In the Architect window, you insert icons that represent eFlow steps into the workspace, set properties for the steps, and connect the steps to create an eFlow.

*Note:* For instructions about customizing the Architect window—including moving the palette, scaling the eFlow view, and splitting the workspace into panes —refer to the Architect online Help.

#### Using the Step Palette

The step palette (Figure 3) is divided into two areas. The top area lists categories that group steps by type: General, CMI, Database, and so on. The bottom area displays icons representing each step within the selected category. To insert a step in the active eFlow, drag its icon from the step palette to the workspace (see Adding Steps to the eFlow on page 22 for details).

G const	
	-
C Database	
CRouting	
C Media Cont	rol 🖃
₩2	갼
Call Subflow	Case
	a 🗉 🔤
	3
Comment	Finish —
함종	$\rightarrow$
GoTo	TE
3010	

Figure 3. The Step Palette

To change the appearance or behavior of the step palette, select **Tools** $\rightarrow$ **Options** and adjust settings on the Palette tab. For example, you can change the category option buttons to check boxes, allowing you to view steps from any combination of categories you select.

#### Using the Watch

The watch (Figure 4) stores information about the eFlow as it is debugged and executed.

×	Name	Value	Туре	Usage	
Π	BusAppIdNotFound	False	Boolean	User Defined	
	bCallbackIdNotFound	False	Boolean	User Defined	
Ш	bCallbackNumberNotF	False	Boolean	User Defined	
Ш	Calld		String	Parameter, In	]
Ш	CB_Day		String	Parameter, Out	
	CB Hour	[	China	Parameter Out	1
	Variables 🖉 Ou	itput λ Debug λ Exceptions λ	Call Stack /		

#### Figure 4. The Watch

The watch is divided into five tabbed pages:

- Variables lists all variables (both standard and user-defined) that can be used by the eFlow to temporarily store data. You can use this page to create, modify, and delete user-defined variables and to assign values to variables. (Right-click a variable to display a pop-up menu of available actions.)
- Output displays errors that occur when the eFlow is validated, messages encountered when the eFlow is debugged, and statistics that are collected during execution.
- Debug displays the contents of variables when the eFlow is run in Debug mode.
- Exceptions displays any error conditions logged during debugging.
- Call Stack tracks an eFlow's execution across multiple subflows during a debugging session.

To toggle display of the watch, select **View** $\rightarrow$ **Watch** or click  $\blacksquare$  on the toolbar.

### Navigating Online Help

eBusiness Architect includes an extensive online Help system that offers all levels of application information. There are numerous ways to get Help when you need it:

Table	1.	Ways to	Get	Help
-------	----	---------	-----	------

Access Point	Displays
₩?	To display pop-up Help on a particular area of the Architect window, click the <b>Help</b> button in the toolbar, then click <b>anywhere</b> in the window.
?	To display pop-up Help on a particular option in a dialog box, click the <b>Help</b> button in the top-right corner of the dialog box, then click an <i>option</i> in the dialog box.
F1	Press <b>F1</b> from within the Architect window to display the online Help window. If a step is selected in the workspace, the Help window displays information about that step.
Help button	Click the <b>Help</b> button in any dialog box to open the online Help window with information about that dialog box.
Help→Help Topics	Select <b>Help Topics</b> from the Help menu to open the online Help window with a Table of Contents for easy navigation.

From within the online Help window, click the **Help Topics** button to access a table of contents and index for the Help, or to use the search utility. Click **Related Topics** to display links to other pertinent Help topics. When overview Help for particular eFlow step is displayed, click **Properties** to access detailed Help about each of the properties available for the step.

For more information about using Windows online Help, select **Help** from the Windows Start menu. In the Contents tab, double-click **How to**, and then double-click **Use Help**. Click any of the topics listed.

# eFlow Configuration and Administration

Before you use Architect to create eFlows, use the Uniphi Administrator to do the following:

- Establish connectivity between eBusiness Architect and one or more LDAP servers.
- Define business applications for collecting eFlow statistics.
- Define outside resources, such as data sources and repository connections, so they can be accessed by eFlows.

- Define subflows to be created and used in eFlows.
- Configure components for the optional Advanced Routing feature (agent pools, queues, and routing rules).

After you create eFlows in Architect, return to the Uniphi Administrator to do the following:

- Create event rules to specify the circumstances that will trigger a given eFlow.
- Manage eFlows and subflows and view runtime statistics.

In addition, you can use Uniphi Administrator to create new user access profiles for Architect.

For detailed information on all these administration tasks, refer to the *Aspect Unipbi Suite Administrator Guide* and online help in the Uniphi Administrator.

Aspect {Product Name and Title}



# Creating eFlows

This chapter gives an overview of the process for creating eFlows to handle customer contacts. It includes the following sections:

- Administrative Prerequisites
- Building an eFlow
- Using Subflows
- Debugging an eFlow
- Creating Event Rules for eFlow Execution
- Managing eFlows

For detailed information about all Architect functions, see the Architect online Help.

## Administrative Prerequisites

Before you create a new eFlow, make sure the following administration tasks have been completed in Uniphi Administrator:

- Required resources are defined. Add or modify any required resource definitions, such as data sources, event rule sets, and repositories.
- Business applications are created.
- Subflow definitions are in place.
- For the Advanced Routing option, components for the Resource Matching Service (RMS) must be configured.

For details, refer to the *Aspect Uniphi Suite System Administrator Guide* or the online Help available from within the Uniphi Administrator.

After you create one or more eFlows, return to the Uniphi Administrator to establish rules for eFlow execution (see Creating Event Rules for eFlow Execution on page 38).

# Building an eFlow

When Uniphi Suite is installed, an Aspect professional works with you to create a standard set of eFlows for your particular business. As your system grows and changes, you might need to create new eFlows or edit old ones.

This section describes how to build an eFlow in Architect, including:

- Starting a New eFlow
- Opening an Existing eFlow
- Adding Steps to the eFlow
- Connecting Steps
- Setting Step Properties
- Creating eFlow Variables
- Adding Comments to the eFlow
- Using Labels in eFlows
- Selecting Steps
- Rearranging Steps
- Saving an eFlow

For detailed information about all Architect functions, see the Architect online Help.

#### Starting a New eFlow

When you first open the Aspect Architect program, the workspace, step palette, and watch area are all empty.

To start a new eFlow:

1. Select **File→New**.

The Events dialog box appears.

Events			?)
<u>S</u> erver:	Local eFlow Location	<b>_</b>	OK
Application:	SampleApp		Cancel
- Event-			
Source:	Aspect CTI		-
<u>N</u> ame:	RouteRequest		•
⊻ariables:	Name CurrentRoute CalingDevice RoutingDevice RoutedCall RouteSelAlgorithm AssociatedCalledDevice AssociatedCalledDevice Priority CorrelatorData UserData	Type ExtendedCallData ExtendedCallData CallConnection Long ExtendedCallData ExtendedCallData ExtendedCallData Blob	Value
	•		Þ

- *Note:* If you are not connected to an eFlow server, an error message appears. You cannot create a new eFlow without a server connection.
- 2. From the Server drop-down list, select the *name* of the eFlow server you want the eFlow or subflow to run against. (Select **Local eFlow Location** if Architect is installed on the server.)
- 3. From the Application drop-down list, select the *business application* you want to use the eFlow for.

Business applications are categories of business (such as Catalog Sales or Service) defined in the Uniphi Suite Administrator for the purpose of reporting. Statistics on executed eFlows are grouped according to application, so you can see how many contacts were presented to Sales, for example. When you create an eFlow, you must select the application you expect it to be used with. You can always associate it with a different application later (see "eFlow Properties" on page 33).

- 4. From the Source drop-down list, select the *source of the event* you expect to trigger this eFlow:
  - Agent Manager Subsystem is used to support Agent Override Services, such as the Front Office API, offered with the Uniphi UCC Core.
  - Aspect CTI is for events received from outside switches (such as an Aspect Call Center) through the Aspect CMI Server.
  - **Aspect Portal Multimedia** is for events received from a media switch adapter, such as TAPI.
  - Subflow is used to create a subflow to be called within another eFlow (see "Using Subflows" on page 35).
  - **User Event** is for user-defined events generated by the Event Scheduler or UDEII (see "User Events and eFlows" on page 44).

5. From the Name drop-down list, select either the *event* you want to trigger execution of this eFlow or the *subflow* you want to create.

The Variables area lists any variables defined for the specified event or subflow.

6. Click OK.

The palette is filled in with steps, and the watch area lists parameters for the selected event. The workspace contains an untitled eFlow with a Start step. You can add eFlow steps to the workspace to start a new eFlow. You can also add variables to the watch for use within your eFlow.

Aspect eBusiness Architec	t - [Sample.wf2] ools Window Help	_			_ 🗆 🗙
🗋 🗅 🖨 🖬 🎒 🗛 Aspect Wo	rkflow System 🔽 💦	% <b>b c</b> ×   • ∘ ∘	÷ # ∭∞ € Q 🖬		333
Callsbriew Case Connent Finish Goto If Goto If Gato If Gato If	(Start)				X
	ſ				
× Name	Value	Туре	Usage		
AgentGroup	0	Long	Parameter, In		
BusinessApplication	0	Long	Parameter, In		
HandlingId		String	Parameter, In		
Id		String	Parameter, In		
Result	0	Long	Parameter, Out	]	
SinkId	0	Long	Parameter In	1	•
Variables	utput $\lambda$ Debug $\lambda$ Exceptions	Call Stack			
For Help, press F1					

Figure 5. The Architect Window Ready for a New eFlow

### **Opening an Existing eFlow**

You can open an eFlow that is stored in the LDAP data store or on your computer's hard drive.

To open an eFlow:

1. Select **File→Open**.

Open			? ×
Ser <u>v</u> er:	Local eFlow Location		
Event So <u>u</u> rce:	Subflow		
<u>E</u> vent:	TSub	<u>D</u> <u>b-b-</u>	
₽			Open
TestSub1			Cancel
			Re <u>f</u> resh
			🔲 <u>R</u> ead Only
			Local File
			Properties
Workflow Name:	TestSub1	Version <u>N</u> umb	er: 0
Description:			

- *Note:* If you are not connected to an eFlow Server, an error message appears. You can still open an eFlow that is stored locally. Click **OK** to close the error message, then click **Local File** in the Open dialog box.
- 2. From the Server drop-down list, select the *server* on which the eFlow is stored. (Select **Local eFlow Location** if Architect is installed on the server.)
- 3. From the Source drop-down list, select the *source of the event* you expect to trigger this eFlow:
  - Agent Manager Subsystem is used to support Agent Override Services, such as the Front Office API, offered with the Uniphi UCC Core.
  - Aspect CTI is for events received from outside switches (such as an Aspect Call Center) through the Aspect CMI Server.
  - Aspect Portal Multimedia is for events received from a media adapter, such as TAPI, or created internally, such as when an agent becomes available.
  - Subflow is used to create a subflow to be called within another eFlow (see "Using Subflows" on page 35).
  - **User Event** is for user-defined events generated by the Event Scheduler or UDEII (see "User Events and eFlows" on page 44).
- From the Event drop-down list, select the actual *event* that triggers the eFlow, or select *subflow* to open a subflow. (Select **RouteRequest** to

specify events routed through the CMI Server.) eFlows for the event you specify appear in the list.

5. Select the *eFlow* that you want to open.

	pen	? ×
Specify the server on which the eFlow is stored, the source of events that trigger it, and the type of event.	Server: Local eFlow Location Event Source: Aspect CTI Event: RouteRequest	
Select this check boy to open	RepositTest	Open Cancel Refresh
the eFlow in read-only mode. In read-only mode, an eFlow cannot be changed or accidentally deleted.		Local File
Click here to navigate to an eFlow stored locally and open it.		Properties
	worktiow Name: In eposit Lest	Version <u>N</u> umber: 2

- 6. Click **Open**.
- *Note:* To quickly open an eFlow that has been opened recently, double-click the *eFlow* in the file list on the File menu.

#### Adding Steps to the eFlow

To build or add onto an eFlow, you insert steps from the step palette, linking them in the order in which you want them to be executed. Use the property sheet associated with each step to specify details about the execution of the step. Be sure to end the eFlow with a Finish, GoTo, or Route Call step.

*Note:* To view steps from two or more categories at the same time, right-click the *top balf* of the step palette and select **Select Multiple**. The option buttons become check boxes, and you can select any number of categories to add steps to the lower half.

You can add steps to the workspace in two ways:

**Dragging.** Drag the *step* icon from the palette into the workspace. If you release on an existing step, step connectors are linked automatically. (This works only if the existing step has a single output connection point.)

- Double-clicking. Select an existing *step* in the workspace, then double-click a *step* icon in the palette. The new step is inserted to the right of the selected step, and step connectors are linked automatically.
  - *Note:* This works only if the selected step has at least one unconnected output connection point. If the selected step includes multiple output connection points, the top output connection is linked to the new step.

For a list of available steps and brief descriptions of their purpose, refer to Chapter 3, eFlow Steps Reference. For details about each of the steps and their individual properties, refer to the Architect online Help.

#### Changing the Step Caption

By default, all steps are given the name of the selected step and numbered starting with 1. For example, the first comment you add is named Comment1, and this name appears underneath the step.

To change the name of a step to something more meaningful:

1. Double-click the *step* in the workspace.

A property dialog box for the step appears. (See "Setting Step Properties" on page 24 for more on property dialog boxes.)

- 2. Select the **Caption** tab at the top of the dialog box.
- 3. Enter *different text* in the Caption field. You can use up to 40 characters. Do not include spaces or special characters.
- 4. Click **OK**.

#### Adjusting Step Width

When inserted, a step is displayed at its minimum width. When the step is selected, a blue arrow appears at the bottom-right of the step. To adjust the width of the step, drag the **arrow** horizontally.

#### **Connecting Steps**

When Uniphi processes an eFlow, it begins at the Start step, then moves from step to step, following the flow of step connections. Each step has an input connection point (on the left side), which accepts connection from a previous step, and at least one output connection point directing control to one or more steps, depending on the outcome of the step.



Figure 6. Input and Output Connections

When you add steps to an eFlow, you connect both input and output connection points to other steps. Depending on how you inserted the steps (see Adding Steps to the eFlow on page 22), some of these connections are made for you.

You can connect steps manually in two ways:

- **Drag and Drop.** Drag the **mouse pointer** from one connection point to another to connect them.
- Click and click. Click one connection point then another to connect the two. (This option allows you to scroll without canceling a connection.)

To remove a connection, select the *connection* and press **Delete**.

You cannot connect an input label or output label to the Start step or to a Finish step, and you cannot connect input labels and output labels to each other.

*Note:* A **A** appears on the step in the workspace if any required branches are not connected.

### Setting Step Properties

Each eFlow step has a set of options, called properties, that you set to determine exactly what that step does. To display the Properties dialog box for the step,

double-click a *step* in an eFlow. Properties are arranged in tabbed pages. To display properties on a different page, select its *tab name*.

Set Variable Pre	operties	? X
General Capti	ion	
_ Set Variable		- 1
Name:	AgentGroup	
	Member:	
_ With Value 0	Df	
Expression:		
	OK Cancel Apply H	elp

Figure 7. A Step Properties Dialog Box

Each Properties dialog box has a Caption tab that you can use to change the name that appears underneath the step in the eFlow.

For context-sensitive help about a particular property, click  $\ref{eq:sensitive}$  in the top-right corner and click the **property control**. For help about the displayed property page, click the **Help** button in the dialog box. To display general help for a step, select the **step** in the workspace and press **F1**.

When you return to the workspace, crucial property setting information appears within the step. If any required properties are not set properly,  $\bigwedge$  appears in the step.



Figure 8. Step with a Warning Icon

#### Creating eFlow Variables

Each Uniphi Suite event comes with a set of predefined parameters, which allow the eFlow to receive data from the source of the event, and in some cases to return data. You can use these variables store data collected by the eFlow. You can also create your own variables for use in the current eFlow only.

To create a user-defined variable for an eFlow:

1. Select **Edit→Edit Variables**.

The eFlow Variables dialog box appears, listing currently defined variables, including event parameters.

Workflow Variables	? ≍
General	1
Name InOutString1	Iype String ⊻ ⊻alue "Test"
Delete New	
0	K Cancel <u>A</u> pply

- 2. Click New. A new variable (named New\_Variable#) appears in the list.
- 3. Double-click the *new variable name*.
- 4. Type a *new name* for the variable.
- 5. In the Type drop-down list, select the *data type* you want to assign to the variable:
  - Boolean for "true" or "false" values
  - DateTime for date and time values
  - Double for floating point data
  - Long for integers
  - String for character strings
- 6. In the Value text box, select or enter a *default value* for the variable. This is the value that will be used if no other value has been stored in the variable.
- 7. Click **OK**.

Your new variable is listed in the Variables tab in the watch, located at the bottom of the Architect window, and will appear in variables lists available in step properties dialog boxes. You can use the variable to store data collected during the processing of an eFlow.

You can also use the Variables tab of the watch to create, edit, rename, clear, or delete user-defined variables. To access a shortcut menu of commands, right-click in the tab.

#### Adding Comments to the eFlow

You can add any number of free-floating comments to an eFlow to help explain sections of the eFlow.

To add a comment to an eFlow:

- 1. Drag the **Comment icon** from the step palette (General category) to anywhere in the workspace. The step appears as the word Comments enclosed in a blue dashed selection box.
- 2. Double-click the **Comment step** to display the Comment Properties dialog box.

Comment Properties	? X
General Caption	
OK Cancel Apply	Help

- 3. Type *any information* to insert as free-floating text in the eFlow.
- 4. Click **OK**.

The text you typed replaces the "Comments" text in the workspace.

5. Drag the **blue arrow** in the bottom-right corner of the selection box horizontally to adjust the width of the comment.

Comments are not connected to any particular step in the eFlow. If you rearrange steps in an eFlow, remember to manually move any comments relating to the steps as well.

#### Using Labels in eFlows

The Input Label and Output Label steps create labels that you can use to connect steps by name, instead of with lines. You connect an output label to a step's output connection point, and then connect an input label with the same name to the input connection of the next step in your logic flow. Making connections by name can simplify the layout of a complex eFlow. For example, multiple steps in an eFlow could branch to the same step or series of steps that handle errors. Using lines to make these connections would make the eFlow hard to follow. Instead, put an output label named "Error" at the output of each step that branches to error handling. Then, put an input label named "Error" at the beginning of the error handling steps.

To use labels to connect steps:

- 1. Drag the **Output Label** icon from the step palette (General category) to the workspace.
- 2. Double-click the **Output Label** step and enter a unique *label name*.
- 3. Click **OK**.
- 4. Link the **output connection** of a step to the Output Label step.
- 5. Drag the **Input Label** icon from the palette to the workspace.
- 6. Double-click the **Input Label** step and select the *name* of the output label you just created from the drop-down list.
- 7. Click **OK**.
- 8. Link the **Input Label** step to the input connection point of another step in the eFlow.

When the eFlow executes, it will jump from the output label to the input label with the same name.

#### Selecting Steps

Before a step can be edited, copied, or moved, you must select it.

To select a single step, click the *step*. A selection box appears around the step with a blue arrow in the bottom-right corner. To change the width of the step, drag the **arrow** horizontally.

[>=	Sleep 1 Seconds	
	Sleep2	]

Figure 9. A Selected Step

To select more than one step in the workspace, use any of the following methods:

- Shift-click. While holding down the Shift key, click a series of steps. If the step is already selected, pressing Shift and then clicking the step deselects it.
- Draw a selection box. Point to an empty area of the workspace and drag the mouse pointer around the steps. To be selected, a step must be entirely contained within the selection box.
- **Shift-drag.** Press **Shift** and then drag the **selected area**. This action toggles between the steps being selected and deselected.
- **Extend select.** Click a *step*, and then select **Edit**→**Select Extend**. This selects that step and all others branching from it to the right.
- **Select All.** Click a *step*, then select **Edit**→**Select All**. This selects all steps in the eFlow.

Dashed boxes appear around all selected steps and their connectors.

To deselect a step or group of steps, click **outside of the selection box**.

#### **Rearranging Steps**

To move a step or group of steps within an eFlow, select the **steps**, then drag them to a different location. The lines that link the steps move along with the steps.

You can use standard Windows commands on the Edit menu (Delete, Cut, Copy, and Paste) to remove selected steps or copy them elsewhere in the eFlow or to a different eFlow.

To align steps automatically while preserving the existing layout of the steps, select the **steps**, and select **Tools→Auto Arrange**.

To rearrange steps without preserving the existing layout, select **Tools**→**Regenerate Diagram**.

The location of Comment objects might change when you use the Auto Arrange or Regenerate Diagram commands. Be sure to move Comment objects to their proper locations after the steps are aligned.

*Caution:* If you have a large eFlow or an eFlow whose layout is acceptable, you might not want to use  $Tools \rightarrow Regenerate Diagram$ , because it could drastically change your step layout.

#### Finding a Step

Once an eFlow expands beyond the screensize, it is sometimes hard to locate a particular step in the eFlow. Architect's new Find feature makes it easy.

To find a step in an eFlow:

- 1. Click a *blank area* of the workspace to make sure no eFlow steps are selected.
- 2. Press CTRL+F. The Find a Step dialog box appears.



3. In the text box, type all or part of the *caption* appearing underneath the step you want to find.

You can use the drop-down box to select text from previous searches.

- 4. Select the precision of the search:
  - **Step name containing text** will find the text embedded in any step caption.
  - **Exact text of step name** will find the text only if it is the entire step caption.
- 5. Select **Match Case** if you want the search to be case-sensitive. Otherwise it will disregard capitalization in the search.
- 6. Click OK.

Note that the Find feature searches for text in the caption that appears underneath the step. This is not necessarily the name of the step as it appears in the palette.

#### Saving an eFlow

While you are building and debugging an eFlow, you can save it locally on your hard drive. Once the eFlow is debugged and ready to be loaded, you can save it on the eFlow execution server.

Each time you save an eFlow, either locally or on the server, Architect performs a validation, and any errors or warnings encountered are noted in the Output tab of the watch area. Double-click an error or warning to select the step where the problem exists.

#### Saving an eFlow on the Server

To save an eFlow on the eFlow execution server:

1. Select **File→Save As**.

A progress bar appears below the watch area while the eFlow is validated, checking for errors. If the eFlow is successfully validated, the Save As dialog box appears, listing all eFlows on the current server. If the eFlow contains one or more errors or warnings, the local Save As dialog box appears, and you can only save the file locally (see "Saving an eFlow Locally" on page 32).

Save As		? ×	ł
Ser⊻er: Event So <u>u</u> rce: <u>E</u> vent:	Local eFlow Location	<u>₽<u>₽</u> <u>₩</u> ₩</u>	
		Save Cancel Refresh	
 <u>W</u> orkflow Name: Description:	TestSub1	Properties	

- 2. If you like, select a different server to store the eFlow on.
- 3. Enter a *name* for the eFlow in the Workflow Name field.
- 4. Enter a *version number* for the eFlow in the Version Number field. (This allows you to create updated versions of an eFlow under the same name.)
- 5. Enter a *description* of the eFlow (optional).
- 6. Click Save.

The Workflow Properties dialog box appears, showing your eFlow by name, the date and time it was last modified, and the name of the business application it was associated with when created.

Workflow Properties		? ×
General		
<u>N</u> ame:	ACB AV Push Variable2	
Last Modified:		
Application:	Aspect Scheduled Callback	•
C <u>H</u> igh	Normal	C Low
- Reporting Options		
<u>T</u> iming:	No timing reports.	•
Counts:	No step counts.	•
<u>S</u> ample Interval:	1	÷
	OK Cancel	Apply

- 7. If you like, you can change the business application associated with the eFlow, the priority set for the eFlow, or any of the reporting options. See "eFlow Properties" on page 33 for details.
- 8. Click **OK**.

The eFlow is saved and its new name appears in the Architect title bar.

#### Saving an eFlow Locally

Whenever you save an eFlow, Architect performs a validation test to locate any possible errors. For example, it checks for resources used in the eFlow that are undefined, properties for a step that are not set, and mandatory connections that have not been made. This command also performs a database cross-reference, ensuring that no resources referenced in the Properties dialog box for the step have been deleted.

If the eFlow contains any errors, it cannot be saved on the server. You can, however, save it locally, and after validation, the local Save As dialog box appears. Use this dialog box to save the eFlow on your local drive.

Save As					? ×
Save jn:	🔄 Workflow		•	+ E 💣 📰•	
History Desktop My Documents My Computer	불] EchoServer.wf 불] LoanCalculator	2 .wf2			
	File <u>n</u> ame:	olve Subtypes2 (0, Aspect C	TI, Rou	iteRequest	<u>S</u> ave
My Network P	Save as <u>type</u> :	Architect Files (*.wf2)		<b>•</b>	Cancel

Figure 10. Local Save As Dialog Box

When an eFlow is saved on your hard drive, it cannot be executed, but you can open it up later for modifications. To open a file that is saved locally, click the Local File button on the Open dialog box.

To purposefully save an eFlow locally, select **Local File** in the Save As dialog box and specify the *name* and *location* for the eFlow.

*Note:* To validate an eFlow manually (without saving), select **Tools**→**Validate**.

#### eFlow Properties

When you save an eFlow, either locally or on the server, the Workflow Properties dialog box appears, showing the name of the eFlow, when it was last modified,

Workflow Properties	5	? ×
General		
<u>N</u> ame:	ACB AV Push Variable2	
Last Modified:		
Application:	Aspect Scheduled Callback	•
○ <u>H</u> igh	• Normal	C Low
- Reporting Options	3	
<u>T</u> iming:	No timing reports.	<b>_</b>
<u>C</u> ounts:	No step counts.	•
<u>S</u> ample Interval:	1	-
	OK Cancel	Apply

and current property settings for the eFlow. You can adjust any of these property settings here, and the new settings will be saved with the eFlow.

Figure 11. Workflow Properties Dialog Box

You can adjust any of the following eFlow properties:

- **Application.** This is the business application that is associated with the eFlow. Business applications are defined in the Uniphi Administrator for the purpose of summarizing statistics.
- Priority determines the urgency with which this eFlow will be executed. If two or more eFlows are queued for execution on the same engine, those with higher priority move to the top of the queue.
- **Timing** determines what kind of timing statistics the eFlow server collects:
  - No Timing Reports disables timing statistics.
  - *eFlow Execution Time* tracks time spent executing an eFlow (not including queue time).
  - Step Execution Time tracks time spent on each step in an eFlow.
  - *Task Execution Time* tracks time spent on an overall task (eFlow time + queue time).
- **Counts** determines what kind of count statistics the eFlow server collects:
  - No Step Counts disables counts.
  - *Workflow Execution Count* counts the number of times an eFlow is executed.
  - *Steps Executions Errors Only* counts the number of times steps within an eFlow triggered an error.
  - *Step Executions All* counts the number of times steps within an eFlow are executed (with or without errors).
- *Branch Executions Errors Only* counts the number of times different error branches are executed.
- *Branch Executions All* counts the number of times each step branch within an eFlow is executed.
- **Sample Interval** is the number of eFlow executions (1-100) that the statistics information is gathered on. For example, when set to 1, statistics are tracked for every eFlow execution. If set to 5, the server tracks statistics for every fifth eFlow execution.

The Reporting statistics are displayed in the watch area when you execute the eFlow from within the Administrator with the View Statistics button (see "Viewing Statistics" in the Uniphi Administrator online Help).

You can also change properties for an eFlow within the Architect program. With no steps in the eFlow selected (click outside of a step in the workspace), select **Edit→Properties**.

You can set eFlow properties from within the eFlows view of the Uniphi Administrator, as well.

### Using Subflows

If you have a series of steps that you use in multiple eFlows (to verify an account number, for example), you can save time and effort by storing those steps as a subflow. You can then execute, or call, that subflow from within any eFlow or other subflow using a single step. Any changes you later make to the subflow are automatically reflected in the eFlows that call it.

To create and use a subflow:

- 1. Before you actually create the subflow, you must define it in Uniphi Administrator. Here you give the subflow a name and number, enter a brief description of it, and set up parameters to use for passing data between the subflow and the eFlow that calls it. For details, refer to the *Aspect Uniphi Suite System Administrator Guide*.
- 2. Next, create the subflow in Architect:
  - a. Select **File→New**.

The Events dialog box appears.

b. Select the *server* you want to run the subflow against and set the Event Source option to Subflow. The Event box lists all subflows that have been defined in Uniphi Administrator, including the one you just defined.

- c. Select the *name* of the subflow you want to create, and click **OK**. A blank workspace appears in Architect.
- d. Build the subflow in the same way that you build an eFlow.
- e. When you are done, select **File** $\rightarrow$ **Save** and save the subflow.
- 3. To execute the subflow from within an eFlow or another subflow, use the Call Subflow step in eBusiness Architect. Here you specify the name of the subflow to execute and map parameters in the subflow with variables in the eFlow.

Use the Uniphi Administrator to manage subflows: create new subflows, edit existing subflows, and delete unused subflows. You can also change reporting options for a subflow, associate a subflow with a different application, and generate a list of eFlows or other subflows that call the subflow.

*Note:* You can maintain different versions of the same subflow, each with a different version number. To create a new version of a subflow, open it in Architect, make the modifications, then select **File→Save As**. Save the subflow under the same name, but select a different version number. When you call the subflow within an eFlow, you specify not only the name of the subflow, but the version number as well.

## Debugging an eFlow

After you build an eFlow, you can test the eFlow by debugging it. Debugging an eFlow simulates eFlow execution, but you have control over how quickly the steps are executed. You can execute steps one at a time, or start execution at a given step. You can set breakpoints at which the execution with pause. As the eFlow executes, the Debug tab in the watch area shows the current variable contents, so you can easily see if the eFlow is performing as expected.

*Note:* You cannot debug an eFlow that is stored locally on your hard drive. You must first save the eFlow on a server.

To debug an eFlow:

- With the eFlow you want to debug in the workspace, select Debug→Settings and select the name of the execution server to test the eFlow on.
- 2. Select **Debug**→**Start** (or press F5). The status of variables used in the eFlow appears in the Debug tab of the watch.
- 3. Choose a command from the Debug menu:

- To debug the entire eFlow, choose **Go** (Ctrl+F5)
- To debug the selected step only, choose Step Over (F10)
- To debug the eFlow starting with the selected step and stopping at the next breakpoint, choose **Break** (Shift+F11).

When an eFlow executes in Debug mode, the step currently being executed is highlighted by a heavy outline. If a breakpoint is set, the step that contains the breakpoint is indicated with a blue, dashed box.



Figure 12. An eFlow in Debug Mode

#### Tips:

- To set a point in the eFlow at which the debugger stops (until you click **Break** again), click a step, then select **Debug→Toggle Breakpoint**.
- To remove a breakpoint, select the breakpoint step and select **Debug→Toggle Breakpoint**. To clear all breakpoints, select **Debug→Clear All Breakpoints**.
- To display statistics while debugging an eFlow, select **Debug→Show Debug Statistics**. Set options on the Overlay menu for displaying statistics (see "The Overlay Menu" on page 42).
- To change the colors used to display overlay statistics, select **Tools**→**Options** and set options in the Overlay Highlights tab (see "Changing Statistics Display Colors" on page 44).
- To change colors used to indicate the next step to be executed, exceptions, and breakpoints, select **Tools**→**Options** and set options in the Debug Highlights tab.
- To connect to a different eFlow server or to toggle display of the Select Execution Server dialog box, select **Debug**→**Settings**.

 Alternatively, you can click buttons on the Debug toolbar instead of using commands on the Debug menu:



## Creating Event Rules for eFlow Execution

Once an eFlow has been saved and debugged, it is ready to be used. To place an eFlow in service, you must create an event rule to trigger it, then activate the rule set containing the new rule. This binds the eFlow both to a condition (the rule) and an event (such as a route request generated by an incoming contact), determining when the eFlow is executed. For the rule, you can enter a Boolean expression that must be met in order to run the eFlow (for example, Application = Sales) or specify the eFlow to execute under all conditions.

To create an event rule:

- 1. Log on to the Uniphi Administrator.
- 2. Use the navigation tree to navigate to **My Enterprise→Business** Administration→eFlows/Rule Sets→Rule Sets.
- 3. If you want to create a new rule set for the event rule, click the **New Rule Set** button ( ) and define a new rule set. Otherwise, you can add your new rule to an existing rule set.
- 4. Click the **Bindings** ( ) button in the Options column. A dialog box appears, showing any existing bindings for the rule set.
  - *Note:* The Bindings button does not appear for rule sets that are loaded. To modify rules in a loaded rule set, you must unload the rule set first.

#### 5. Click Add/Update.

- 6. In the displayed dialog box, specify the *event* you want to associate with your eFlow (select its *subsystem* first), then click **New Rule**.
- In the displayed dialog box, select your *eFlow* and *version number* and set up the condition you want to trigger the eFlow (or select Execute eFlow When This Expression Is True).
- 8. Set up any other conditions to add to the binding. You can use the Expression Builder to create expressions that need any of the available functions, such as Date/Time functions or text string handling. Also, any parameter that has data in it when the event arrives at the eFlow system can be evaluated to decide which eFlow to run.

- 9. Use the 💷 and 🖳 buttons to reorder the rules if necessary. Rules are processed in the order in which they appear in this list, and the first rule that returns a true value or is set to "always run" triggers the associated eFlow.
- 10. Click **OK** to return to the previous dialog box with your new binding displayed in the list.
- 11. To load the rule set so the eFlow is executed as directed, first select the *engine* you want to use to execute the eFlow, then click the the button in the Options column for that row. Click **Yes** when prompted for confirmation.

Any eFlow that is bound to a loaded rule set cannot be modified in Architect. Nor can you modify bindings for a loaded rule set. To unload a rule set, return to the Rule Sets view of the Administrator, select the *engine* on which the rule set is loaded, click the button for the rule set, and click **Yes** when prompted for confirmation. This releases all eFlows bound to the rule set so they can be modified.

For details about all these steps, refer to the *Aspect Uniphi Suite System Administrator Guide* or the online Help for Uniphi Administrator.

### Managing eFlows

The Uniphi Administrator makes it easy to keep track of eFlows that are saved on the execution server. With it, you can:

- View a list of active eFlows.
- Generate a cross-reference list of event rule sets that reference a specific eFlow.
- View or modify properties of an eFlow.
- Delete an eFlow.
- View runtime statistics for an eFlow.

For information about these procedures, refer to the *Aspect Uniphi Suite System Administrator Guide* or the online Help for the Administrator. For details on viewing runtime statistics, see the following section.

## **Displaying eFlow Statistics**

After an eFlow is loaded on the execution server and has been executing for some period of time, it may be useful to display overlay statistics. You can review overlay statistics to determine whether or not an eFlow is working as you intended it to.

For example, by reviewing statistics on which paths are being taken, you can more easily see if one path is never taken, which could indicate a problem in the eFlow logic. By examining the statistics, you will find that problems are more obvious.

To display eFlow statistics:

- 1. Log on to the Uniphi Administrator.
- 2. In the left navigation tree of the Administrator, select **My** Enterprise→Business Administration→eFlows/Rule Sets→Rule Sets.

The Rule Sets view appears in the right frame, listing all rule sets by name.

3. Click the 😰 Options button next to a loaded rule set that contains a rule bound to the eFlow.

The Add/Update Event Rules dialog box appears, showing bindings in the rule set, listed by *Subsystem: Event*.

Subsystem : Event							
	eFlow	Version	Start Condition	Active	Status	Options	
± Use	erEvent : A	ACB Place Ca	llback	-	-	-	
🗄 UserEvent : ACB Generic Query Callback							
± Use	erEvent : A	ACB Generic :	Schedule Callback				
🗄 UserEvent : ACB Generic Cancel Callback							
🗄 UserEvent : ACB Generic Change Phone Number							
E LiserEvent : ACB Generic Callback Declined							
					Add/U	pdate	

- 4. Click the + next to a binding to expand the listing, showing all rules within the binding. For each rule, you can see:
  - The associated eFlow and version number
  - The condition to be met in order for the eFlow to be executed
  - Whether the rule is active

- Whether the rule is currently loaded onto an engine

eFlow     Version     Start Condition     Active     Status     Options       UserEvent : ACB Place Callback     0     Always     Yes     Version     Version       UserEvent : ACB Generic Query Callback     0     Always     Yes     Version     Version       UserEvent : ACB Generic Schedule Callback     UserEvent : ACB Generic Schedule Callback     Version     Version     Version	Sut	osystem : Event						
UserEvent : ACB Place Callback         Image: Section of the section		eFlow	Version	Start Condition	Active	Status	Options	
ACB Place Callback       0       Always       Yes         UserEvent : ACB Generic Query Callback         UserEvent : ACB Generic Schedule Callback         UserEvent : ACB Generic Cancel Callback	∃ Use	rEvent : ACB Place	Callback	-				
UserEvent : ACB Generic Query Callback UserEvent : ACB Generic Schedule Callback UserEvent : ACB Generic Cancel Callback	*33 22	ACB Place Callback	0	Always	Yes			
UserEvent : ACB Generic Schedule Callback         UserEvent : ACB Generic Cancel Callback	± Use	rEvent : ACB Gener	ic Query C	allback				
UserEvent : ACB Generic Cancel Callback	🗄 UserEvent : ACB Generic Schedule Callback							
	± Use	rEvent : ACB Gener	ic Cancel C	allback				
- HearEvant - ACB Caparis Change Dhone Number	-							

5. If Architect is installed locally and the displayed rule set is loaded onto the engine, the Options column includes a button for viewing statistics:
Click next to the eFlow to launch Architect and view statistics for the eFlow.

🐳 Aspect eBusiness Archit	tect - AsptRM5 Push (1, Aspect Portal Multimedia, Aspect Voice)(Read-Only)	_ 8 ×
<u>File E</u> dit <u>V</u> iew Debug <u>O</u> v	verlay <u>T</u> ools <u>Wi</u> ndow <u>H</u> elp	
🛛 🗅 😂 🖬 🎒 🗛 Aspect V	Warkflow System 🔽 🛠 📗 & 🖻 🖻 🗙 🗠 ↔ 🖓 🗊 📾 🛛 🚾 🔍 🗨 🖬 🔝 🔍 🔍 💭 🖬 👘 💽 🔛 🚏 🖇 😤 🛱 🖤 🖤	
General Cont Database Contactor Cont	Image: Solution Council Association Council Associatio Council Association Council Association Council Associ	
Make On Predicti Connecti Select Agent Set Task Data		
Workflow run court (0) time     Total workflow runtime     Average workflow runtime     Average workflow runtime	Es. 30) mareos. (0.00) mseos. (0.00) mseos. \Output ∫ Debug ∑ Exceptions ∑ Call Stack ∕	
For Help, press F1	X 37, Y 14	NUM SCRL

#### Figure 13. Architect in Statistics View Mode

In Statistics View mode, general statistics settings are shown above the eFlow. Statistics for each step appear above the individual steps. To adjust the color of statistics display, use the Options pop-up menu (see "Changing Statistics Display Colors" on page 44).

#### The Overlay Menu

In Statistics or Debug mode, an Overlay menu is added to the main menu. Use commands on this menu to control statistics display.

- **Show** toggles display of statistics in the workspace.
- **Refresh** updates the display using fresh data and any newly selected options.
- **Calculation Type** determines how statistics are calculated. You have three options:
  - Absolute Values displays step statistics in actual or absolute numbers.
  - Step Percentage displays step statistics in percentages.
  - Branch Percentage displays branch statistics in percentages.
- **Execution Count** displays how many times the server executed the steps in the eFlow.
- **Error Count** displays the number of errors found in the eFlow.
- **Exception Count** displays the number of exceptions found in the eFlow.
- Average Runtime displays the average time taken to execute a step.
- **Total Runtime** displays the total time taken to execute a step over a period of multiple executions.
- **Maximum Runtime** displays the longest time taken to execute one step.
- **Settings** lets you set options for collecting statistics (see "Adjusting the Statistics Collection Period" on page 42).

The Error Count and Exception Count options are selectable only if errors or exceptions are found in the eFlow. Availability of other options depends on the reporting options selected in the eFlow properties page (see "eFlow Properties" on page 33).

#### Adjusting the Statistics Collection Period

To change the period of time over which eFlow statistics are summarized, select **Overlay→Settings**. The Statistics Display Settings dialog box appears.

ings 🛛 🛛 🗙
<b>_</b>
•
Cancel

Figure 14. Statistics Display Settings Dialog Box

This dialog box lets you adjust the following options:

- **Consolidate Interval** determines the intervals of data that are summarized for statistical display: Every 5 Minutes, Hourly, Daily, or Weekly.
- Rollover specifies the period of data to collect: Current, Rollover1, Rollover2, or Rollover 3. For example, if Consolidate Interval is set to Hourly and Rollover is set to Current, statistics for the current hour will be shown. If Rollover is set to Rollover1, statistics for the previous hour will be shown. Rollover2 would show statistics for the hour before that.

The Settings command is not available when viewing Debug statistics and is available only for execution statistics.

#### **General eFlow Statistics**

In Statistics mode, the following general statistics for the eFlow are shown in the Output tab of the watch area:

- **eFlow Run Count** shows the number of times the server executed the eFlow.
- **Total eFlow Runtime** shows the total time taken to perform all eFlow executions.
- **Maximum eFlow Runtime** shows the longest time taken to perform one eFlow execution.
- Average eFlow Runtime shows the average time taken to perform one eFlow execution.

Statistics are shown for the interval and rollover period selected in the Settings dialog box. Click **Overlay** $\rightarrow$ **Refresh** to update this display.

### **Changing Statistics Display Colors**

To change the colors used to display statistics in the workspace:

- 1. Make sure no steps are selected, then right-click an empty area of the workspace.
- 2. Choose **Options** from the pop-up menu.
- 3. Click the **Overlay Highlights** tab in the Options menu.
- 4. Make desired changes to the default colors. To use different colors for numbers that fall within a given range, type a name for the range in the Highlight Name text box and set options in the Properties area, including the start and end values for the range (lower bound and upper bound).
- 5. Click OK.

Options Workspace Palette Overlay Hi	ighlights Debug Highlights ]
Highlight Name alert Propertijes Background Eoreground Border Lower Bound 90.0000 Upper Bound 100.0001 Percent ?	New Delete  Default Background Foreground Border
0K	Cancel <u>Apply</u>

Figure 15. The Overlay Highlights Tab of the Options Menu

## User Events and eFlows

The Aspect CMI System includes a user event application programming interface (API) that you can use to create a client application for generating events. These events can then be sent to the eFlow engine for consideration in triggering eFlows. For events to be used with eFlows, several tasks must be completed:

- The client application must be configured to use the CMI Server as a hub for dispatching events.
- The CMI Server must be configured to direct events to the eFlow engine.

- The CMI Server must be defined in the Aspect Event Connections view of the Uniphi Administrator.
- User events must be constructed in the client application.
- User events to be used with eFlows must be defined in the Event Definitions view, using the same ID number and parameter settings as those set up in the client application.
- eFlows must be bound to the user event in the Rule Sets view, based on variable values generated by the event.

Once all these conditions are met, the following occurs:

- 1. When the client application sends an event to the CMI Server, it is redirected to the eFlow engine via event connections specified in the Aspect Event Connections view. Any parameter values supplied by the client application are sent as well.
- 2. The eFlow engine looks up the event in the Event Definitions view. If the client application did not supply all the parameter values specified in the definition, it uses the definition's default values for those parameters.
- 3. The eFlow engine populates eFlow variables with input parameter values associated with the event: both those supplied by the client application and any defaults carried over from the Uniphi Administrator event definition.
- 4. The eFlow bound to that user event (in the Rule Sets view) is automatically executed.
- 5. If the event includes an "Out" parameter, when an eFlow bound to it terminates, the eFlow engine sends the current contents of the eFlow variable back to the client application.

Aspect {Product Name and Title}



# eFlow Steps Reference

This chapter presents a table describing each of the eFlow steps available in eBusiness Architect. For details about how to use each of these steps, refer to the eBusiness Architect online help.

Icon	Step Name	Description
ø,	Advanced HTTP	Communicates with the Web server in any unstructured data format, such as XML.
<b>@</b>	Allocate Correlation Key	Allocates keys to correlate user data for later retrieval from the Data Store or TimesTen database.
	Begin Transaction	Establishes a <i>transaction</i> to which subsequent database steps in the eFlow can be assigned. Results of database steps assigned to the transaction can then be discarded all at once (with a Rollback Transaction step) or committed to the database (with a Commit Transaction step).
置 XML	Build XML	Builds an XML document and stores it in an eFlow variable.
	Call DLL	Executes a specified function in a user-supplied DLL file.
	Call Subflow	Executes a specified subflow using given values for defined parameters.
×å	Cancel Agent Selection	Cancels any pending agent selection request performed with the Select Agent step.
	Cancel Scheduled Event	Cancels a scheduled event, removing it from the list of scheduled events in the Event Scheduler system.
?	Case	Evaluates multiple conditions, then takes the path corresponding to the first condition found to be true.
8	Collect	Plays announcements and collects DTMF data from the contact.

Table 2. Architect eFlow Steps for Uniphi Suite

Icon	Step Name	Description
8	Comment	Lets you insert text in the eFlow workspace to help annotate the eFlow with notes.
ſ	Commit Transaction	Carries out the database changes initiated by all steps assigned to the named transaction (that have been executed up to this point in the eFlow).
	Connect Task	Connects a call to an agent's task.
•2	Connect Telephony Server	Connects Uniphi Suite with an available telephony server.
	Count Scheduled Event	Counts all events with a given ID and tag that fire within a specified time range.
ò <del>,</del>	Deallocate Correlation Key	Deallocates a non-rolling key allocated with the Allocate Correlation Key step.
	Delete Repository Data	Deletes data in a repository database.
×J	Delete Row	Executes an SQL query that removes all rows from a database table that meet a given condition.
) New Market	Disconnect	Ends a connection and, if the call is still in the eFlow, terminates the call. Used to end a contact in case of errors.
₽	End Route	Returns a call through the Aspect CMI Server back to the originating contact center switch without including any routing suggestions.
ਾ∎ ∎	Finish	Marks the end of an eFlow. All eFlow paths must end a Finish, GoTo, or Route Call step.
] <b>8</b> இ்	Get Agent State	Retrieves the current state of the specified agent.
ť	Get Business Data	Retrieves data that the eFlow has stored in one or more business data integration (BDI) fields with a previous Set Business Data step.
	Get CCD	
	Get eFlow System Data	Copies runtime eFlow system data, such as eFlow Task ID, into eFlow variables.
ij	Get First Row	Executes an SQL query that selects all rows in a database that meet specified conditions, then retrieves the contents of the first row, storing it in a structured variable.
Ĩ	Get Next Row	Executes an SQL command that retrieves the next row in a set of database rows selected by a preceding Get First Row step. It stores the data in a structured variable defined by the Get First Row step.

Table 2. Architect eFlow Steps for Uniphi Suite (continued)

Icon	Step Name	Description
Ĩ	Get Private Data	Retrieves private data from a switch-specific field attached to the routing request.
	Get Repository Data	Lets you extract data stored in a repository by an outside application.
الآن • ال	Get Task Data	Retrieves data associated with a given task from the MMB database. (The data must have been previously stored with the task record with a Set Task Data step.)
33	GoTo	Passes control from the current eFlow to another.
ø	НТТР	Enables communication with an HTTP server, so the eFlow can access data available through a Web site or interact with a Web application.
$\diamond$	lf	Evaluates a condition, then takes one path if the condition is true, another path if the condition is false.
ů≥j	Initiate Task	Initiates a task and stores the task ID in an eFlow variable.
	Input Label	Inserts an input label that corresponds to an existing output label name. You can use input and output labels to link steps by name instead of connecting the steps with lines.
J <sup>_</sup>	Insert Row	Executes an SQL command that inserts a row in a specified database table and fills it with data stored in a given variable.
≓ 	Invoke VXML	Lets you use a custom VXML (VXML 2.0) script to perform speech recognition tasks.
La,	Knowledge Base Lookup	Searches a specified knowledge base for specified information.
å <b>-(</b> )	Log Off Agent	Logs an agent off of all possible devices listed in their static configuration file.
°€	Log On Agent	Automatically logs an agent on to all media devices listed in the agent's configuration file, including TeleSets, telephones, and e-mail and web devices.
	Log Message	Stores a message in a log file.
_  <b>1</b> 70) 24,€8	Make Predictive Call	Makes an outbound call and optionally sends data to the switch.
	New Business Data	Creates a new, blank Business Data Integration (BDI) record, which can be used to store data for reporting.

|--|

Icon	Step Name	Description
S)	On Connection Cleared	Specifies the path to take when the specified Connection ID has been cleared.
٠	On Exception	Specifies paths to take when various types of exceptions arise during execution.
	Output Label	Inserts an output label that references a different area of the eFlow to jump to. corresponds to an existing output label name. You can use input and output labels to link steps by name instead of connecting the steps with lines.
《君 XML	Parse XML	Parses the XML document content and stores the result in an eFlow variable.
€ ∰	Play Announcement	Plays one or more announcements to the contact.
+ _   _ ↑	Play Form	Plays a series of announcements prompting a caller for information and collects up to 25 responses in eFlow variables defined as "fields." (Requires a VXML server.)
	Play Menu	Plays a menu of up to 12 choices to the caller, then takes the branch associated with the caller's response. (Requires a VXML server.)
	Playback	Plays a media file that was previously recorded with a Record step.
8£	Prompt	Plays announcements, collects a single DTMF digit from the contact, and takes a branch based on the digit.
	Record	Records a media file.
<b>\$</b>	Release Row Set	Frees up any resources (such as indexes, temporary tables, and file or row locks) that were acquired by a preceding Get First Row or Get New Row step, so they can be utilized by other steps in the eFlow.
	Reschedule Periodic Event	Lets you change the existing schedule for a periodic event (set up with a Schedule Periodic Event step).
	Reschedule Single Event	Lets you change the existing schedule for a one-time event (set up with a Schedule Single Event step).
*	Reserve Agent	Reserves or releases a Media Blending Agent in the available set of agents in the TimesTen database by changing the state field of a record in TimesTen to "RESERVED".
	Reserve Task	Reserves a task from the available set of tasks in the TimesTen database by changing the state field of a record in TimesTen to "RESERVED".

Table 2. Architect eFlow Steps for Uniphi Suite (continued)

Icon	Step Name	Description
2	Rollback Transaction	Discards any database changes initiated by steps in a given transaction (that have been executed up to this point in the eFlow).
	Route Call	Directs a call back through the Aspect CMI Server to a contact center switch.
	Save As	Saves the previously-recorded media file in a given format.
M	Schedule Periodic Event	Schedules an event to fire repeatedly, every given number of seconds, either indefinitely or until a given stop time is reached.
	Schedule Single Event	Schedules a one-time event.
-¢	Select Agent	Looks in a specified queue for an agent to handle an incoming contact. The agent selected depends on routing rules set up in the queue and parameter values passed along with the Select Agent request.
	Send E-mail	Sends an e-mail to the specified e-mail address.
<u>,</u>	Send Milestone Data	Sends Milestone events and related data for use in reporting.
	Send SNMP v1 Trap	Injects alerting information, in the form of an SNMP trap, into the SNMP network.
<b>8</b> û '+⊡	Set Agent State	Changes the state of an agent to either Unavailable or Available.
Ĕ	Set Business Data	Copies data from eFlow variables into one or more DataMart fields for use in reporting.
	Set CCD	
भू दन्द्	Set Language	Changes the language in which the eFlow plays media files (for example, announcements and prompts) and evaluates data-driven language (DDL) expressions.
(≁% Ƴ]	Set Multiple Variable	Stores values in multiple variables.
<b>∮a</b> ri	Set Music	Specifies the announcements that contain music on a per-call basis for calls on hold or in queue.
a L,	Set Private Data	Stores private data in a switch-specific field attached to a routing request.
	Set Repository Data	Stores data in a repository, where it can be accessed by call center software and other outside applications.

Table 2. Architect eFlow Steps for Uniphi Suite (continued)

Icon	Step Name	Description
₹ 2	Set Task Data	Sends data for a task to the T_MB_TASK_PROP table in the MMB database. You can later retrieve the data with a Get Task Data step.
[↑% Ƴ]	Set Variable	Stores a value in a given eFlow variable.
	Set Whisper	Specifies the announcements to be played to the receiving agent before connecting a voice call.
Ì	Sleep	Pauses the eFlow for a specified period of time.
) () ()	Speak Phrase	Plays an announcement and collects any DTMF digit entered by the contact.
e B B B B B B B B B B B B B B B B B B B	Speak Variable	Invokes the Data Driven Languages (DDL) processor to evaluate data and determine the set of media files to play to a contact.
Start	Start	Marks the beginning of an eFlow.
<u>ଚ</u> ୮	Transfer Call to Application	Transfers an incoming call to an application (such as an eFlow application, CCT, or other switch-dependent application).
₹ ¢	Transfer Task to Application	Transfers a task to an eFlow associated with a specific application.
Ø	Update Event Payload	Changes the value of input parameters for a single scheduled event or multiple events.
Ĵ	Update Row	Executes an SQL UPDATE statement that modifies the values in one or more rows of a connected database table.
$\mathbb{X}$	Wait	Pauses the eFlow for a given number of seconds, optionally playing music or announcements to the contact (for IP calls).
	Web Service	Invokes basic Web services using SOAP-encapsulated XML.

Table 2. Architect eFlow Steps for Uniphi Suite (continued)

# **General Steps**

The General step category contains eFlow steps that can be used in all types of eFlows. They include:

- Call Subflow
- Case

- Comment
- Finish
- ∎ GoTo
- If
- Input
- Output
- Send Milestone Data
- Set Multiple Variable
- Set Variable
- Sleep

You can use these steps to perform basic eFlow functions, such as inserting comments, jumping to a different part of the eFlow, setting variables, and so forth.

### Aspect CMI Steps

Aspect CMI steps are used to communicate through the CMI Server to a connected switch adapter. They include:

- Delete Repository Data
- End Route
- Get Private Data
- Get Repository Data
- Route Call
- Set Private Data
- Set Repository Data

You can use these steps to store data in a repository or switch-specific field, retrieve data from a repository or switch-specific field, and route a call back through the Aspect CMI Server.

### **Database Steps**

Database steps are used to access data in a connected ODBC database. (The database must first be defined as a data source in the Uniphi Administrator.) These steps include:

- Begin Transaction
- Commit Transaction
- Delete Row
- Get First Row
- Get Next Row
- Insert Row
- Release Row Set
- Rollback Transaction
- Update Row

You can use these steps to query a database and retrieve information and to update a database with new information.

### **Routing Steps**

Routing steps pertain to routing contacts to and from outside switches. They include:

- Allocate Correlation Key
- Cancel Agent Selection
- Deallocate Correlation Key
- Get CCD
- Get Task Data
- Make Predictive Call
- On Connection Cleared
- Select Agent
- Set CCD
- Set Task Data

- Transfer Call to Application
- Transfer Task to Application

You can use these steps to select an appropriate agent to handle a contact, transfer a call or task to a specific application, make an outbound call, and store contact data under a correlation key.

#### Media Control Steps

Media Control steps are used with the TAPI adapter to communicate with voice calls from an IP PBX switch. They include:

- Collect
- Play Announcement
- Playback
- Prompt
- Record
- Save As
- Set Language
- Set Music
- Set Whisper
- Speak Phrase
- Speak Variable
- Wait

You can use these steps to prompt a caller for information or for a menu selection, play music while a caller is on hold, specify an announcement to play to the agent before connecting the caller, and record the caller's voice as a media file.

## Advanced IVR Steps

Advanced IVR steps incorporate the use of an outside VXML server to integrate speech recognition into an eFlow. They include:

■ Connect Telelphony Server

- Play Form
- Play Menu
- Invoke VXML

Use these steps to connect to an outside telephony server that offers IVR functionality, to prompt callers for information or menu choices and accept verbal responses, and to use a custom VXML script to perform speech recognition and Text-to-Speech (TTS) tasks.

## **Reporting Steps**

The Reporting steps interact with the DataMart database, which is used to store information used in reports. They include:

- Get Business Data
- New Business Data
- Set Business Data

Use these steps to copy data from eFlow variables into DataMart fields and to retrieve data stored in DataMart fields.

### Scheduler Steps

The Scheduler steps let you schedule an eFlow to be run at a later time. They interact with the Event Scheduler Service, updating a queue of scheduled user-defined events. Each time an event is fired, the eFlow associated with the event (in the Uniphi Administrator) is executed. These steps include:

- Cancel Scheduled Event
- Count Scheduled Event
- Reschedule Periodic Event
- Reschedule Single Event
- Schedule Periodic Event
- Schedule Single Event
- Update Event Payload

Use these steps to schedule a user-defined event to be fired off either once or periodically, count occurrences of a given event, and reschedule events. Events you schedule must first be defined in the Uniphi Administrator. You can also use the Uniphi Administrator to schedule, monitor, edit, and delete scheduled events.

### Agent Interaction Steps

The Agent Interaction steps manage agent states. They include:

- Connect Task
- Get Agent State
- Initiate Task
- Log Off Agent
- Log On Agent
- Set Agent State

Use these steps to connect a call to an agent's task, check or change an agent's current state, log an agent on or off all media devices, and establish a new task ID.

### System Interaction Steps

The System Interaction steps interact with an external system. They include:

- Advanced HTTP
- Build XML
- Call DLL
- HTTP
- Knowledge Base Lookup
- Parse XML
- Web Service

Use these steps to send messages back and forth to an external system, such as a Web server or HTTP server, and to initiate a Web service. You can also use the Call DLL step to execute a function in a user-supplied DLL file.

## System Management Steps

The System Management steps include the following:

- Get eFlow System Data
- Log Message
- On Exception
- Send E-mail
- Send SNMP v1 Trap

Use these steps to communicate with the Uniphi system, for example, to specify paths to take when various types of exceptions occur in the eFlow and to send an e-mail.

## Legacy Routing Steps

The Legacy Routing steps are used with standard push/pull routing eFlows. They include:

- Reserve Agent
- Reserve Task

Use these steps to reserve a specific task for an available agent or to reserve an available agent for an incoming task.



# Task-Oriented eFlow Samples

This chapter presents several basic eFlows that perform specific functions. It contains the following sections:

- Data Access eFlows includes examples of eFlows that store and retrieve data from outside databases, including private data (used in reporting) and repositories.
- Agent Task eFlows includes examples of eFlows that can be used to log an agent on or off media devices and to reserve an agent or task.
- Media Task eFlows includes examples of using Media Control and Advanced IVR steps to interact with contacts.

The next chapter, Sample Routing eFlows, presents eFlows used to direct contacts to a connected call center or media switch.

### Data Access eFlows

There are three types of databases you can access with an eFlow:

- An Oracle database.
- A data repository.
- Private data.

This section includes an example of each.

#### Database Access

Steps in the Database step category let you retrieve information from an outside Oracle database, using an SQL query. You can also add new data to a database, overwrite existing data, and delete data. These steps are most often used to verify customer information (such as an account number) or to retrieve additional information about a customer.

This example eFlow searches through a database table for a record with "Sales" in the Name field. When it finds it, the eFlow releases the selected row set before continuing on to the next step.



Figure 16. A Database Access Sample

The steps in this eFlow perform the following functions:

- 1. **GetFirstRow.** This step queries a database for all records in a table named Group, storing the values of the first row in a structured variable named Group.
- 2. If. This step checks the row to see if the Name field contains "Sales."
- 3. **ReleaseRowSet.** This step releases the row set (defined by the Get First Row step) so that any resources locked by the query can be used to other steps in the eFlow.
- 4. **GetNextRow.** This step retrieves the next row in the set (queried with the Get First Row step).

#### Data Repository Access

Two eFlow steps in the Aspect CMI category (Set Data Repository and Get Data Repository) let you store data in and retrieve data from a repository, which can be accessed by call center software and other outside applications. This is especially useful when transferring calls between switches, or when coordinating screen-pop functions.

The example here presents two sections of an eFlow used to store and retrieve information associated with an interswitch call. (See Chapter 6, Interswitch Routing for a description of the full eFlow, as well as detailed information on the interswitch routing feature.)

The first section of eFlow logic (Transfer Call) executes when an agent is selected on a different switch than the one that received the initial call. It requests a correlation key for the call, then uses the Set Data Repository step to store the selected agent ID (paired with the correlation key) in the repository database.

The second portion (Incoming Transferred Call) executes when a call comes in from another switch. It looks in the data repository for a route number and DNIS digits associated with the call, then stores that data in two variables for later access.



Figure 17. A Data Repository eFlow Sample

The steps in the Transfer Call section perform the following functions:

- 1. **AllocateKey.** This Allocate Correlation Key requests that the system assign a correlation key to the call, which performs two functions. It allocates a DNIS number to use to transfer the call to an Aspect ACD, and it is used to store data associated with the call in the repository database.
- 2. **StoreRepositoryData.** This Set Repository Data step stores the ID of the agent selected by an earlier Select Agent step in the repository database under the key allocated to the call (stored in ALLOC\_DNIS).
- 3. **SetRouteParameters.** This Set Multiple Variables step sets output parameters to indicate that the selected agent is remote and to specify the DNIS number to use to transfer the call.

The steps in the Incoming Transferred Call section perform the following functions:

- 1. **GetRepositoryData.** This step looks in the repository database for the agent ID stored with the call's correlation key and stores that value as a string in the StoredAgt eFlow variable. If no agent ID is found, the step branches to the No Agent step.
- 2. **SetOutputParameters.** This Set Multiple Variables step sets output parameters to indicate that an agent has been found for the call and to specify the agent ID.
- 3. **NoAgent.** This Set Variable step sets the MB\_AGENT\_ID output parameter to 0, indicating that no agent ID was found.

#### **Private Data Access**

Two other steps in the Aspect CMI category (Get Private Data and Set Private Data) let you save and retrieve information in a switch-specific field attached to a routing request. That information is then passed back to the call center switch along with the routing response, where it can be used in routing decisions or given to the agent handling the call. These steps are used in conjunction with a Route Call or End Route step to pass the call back to the originating call center switch, along with the private data.

This example eFlow retrieves the caller's account number stored as private data with the call routing request, looks up the information about the account in the customer database, and passes that data back to the call center switch with the call...



Figure 18. A Private Data Access eFlow

The steps in this eFlow perform the following functions:

1. **GetPrivateData.** This step retrieves the account number entered by the caller from the call center call record.

- 2. **GetFirstRow.** This step looks up the name of the company associated with the account in an Accounts database and stores the name in a variable called "Result."
- 3. **SetPrivateData.** This step stores the caller's company name in a variable in the Aspect Call Center, where it can be accessed by the CCT routing the call.
- 4. **RouteCall.** This step routes the call back to the same CCT that issued the routing request.

### Agent Task eFlows

The example eFlows in this section can be used to log on or off the Uniphi System.

#### Remote Log On

The Log On Agent step automatically logs a specified agent on to one or more media devices, including TeleSets, telephones, and e-mail and web devices. For log-on information you can rely on data configured for the agent in the Uniphi Administrator, or you can override that data -- either with event data already associated with the eFlow or with properties you set for the step.

With the following example eFlow, an agent can call up from any remote location and log on to a TeleSet or telephone. It prompts the caller for an agent ID, then looks up the ID in the MMB database. If it is a valid ID, the eFlow logs the agent on to whatever media devices the agent is configured for (in the Uniphi Administrator), then plays an announcement relaying the success of the log on.



Figure 19. Remote Log On eFlow

The steps in this eFlow perform the following functions:

- 1. **CollectAgentID.** This Collect step prompts the caller for an Agent ID number and stores the collected digits in a variable named AgentID.
- 2. **ValidateAgentID.** This Get First Row step looks up the agent ID collected from the caller in the Uniphi Administrator database to make sure it is a valid ID.
- 3. **LogOnAgent.** This step logs the agent on to all media devices defined for the agent in the Uniphi Administrator.
- 4. **LogOnSuccessful.** This Play Announcement step plays an announcement saying that the caller has been successfully logged onto configured media devices.

#### Remote Log Off

The Log Off Agent Step logs an agent off of all devices defined for the agent in the Uniphi Administrator, including TeleSets, telephones, and e-mail and web devices.

With the following example eFlow (Figure 20), an agent can call in from a remote location to log off of all media devices. It prompts the caller for an agent ID, then looks up the ID in the MMB database. If it is a valid ID, the eFlow changes the agent state to unavailable, then logs the agent off all media devices and plays an announcement relaying the success of the log-off task.



Figure 20. Remote Log Off eFlow

The steps in this eFlow perform the following functions:

- 1. **CollectAgentID.** This Collect step prompts the caller for an Agent ID number and stores the collected digits in a variable named AgentID.
- 2. **ValidateAgentID.** This Get First Row step looks up the agent ID collected from the caller in the MMB database to make sure it is a valid ID.

- 3. **SetAgentState1.** This step forces the agent into the Unavailable state and registers a reason code of 5 (Task Completed).
- 4. **LogOffAgent1.** This step logs the agent off any media devices defined for the agent in the Uniphi Administrator and registers a reason code of 6 (System).
- 5. **LogOffSuccessful.** This Play Announcement step plays an announcement saying that the caller has been successfully logged off of all configured media devices.

### Media Task eFlows

There are two categories of steps that rely on outside media servers for completion: Media Control and Advanced IVR.

- Media Control steps use a TAPI media adapter to communicate with voice calls from a Cisco IP PBX switch. You can use these steps to play prompts, music, and announcements, to record the caller's voice, and to collect caller responses.
- Advanced IVR steps support Uniphi Voice Portal. They connect with a telephony server to incorporate IVR (Interactive Voice Response) functionality. You can use these steps to play announcements, collect information from callers (both digital and audio), and utilize voice recognition software.
- *Note:* Media Control steps are not supported by the VXML telephony server, so you cannot use a mix of steps from these categories. Once you have connected to a telephony server, using Advanced IVR steps, you cannot utilize any Media Control steps (other than Set Language).

The first two examples in this section use Media Control steps to communicate with callers over a TAPI server. The third uses Advanced IVR steps to collect information from callers using voice recognition.

#### **Collecting Caller Responses**

This example eFlow uses Media Control steps to welcome the caller and prompt for an account number before selecting an agent to handle the call.



Figure 21. An eFlow That Uses TAPI to Collect Customer Information

The steps in this eFlow perform the following functions:

- 1. **PlayWelcome.** This Play Announcement step plays an announcement welcoming the caller.
- 2. **CollectAccountNumber.** This Collect step plays an announcement asking the caller to enter an account number. It stores collected digits in a variable named Collected\_AccountNumber.
- 3. **SelectAgent.** This Select Agent step looks for an available agent in routing queue 101. If no agents are currently available, the eFlow takes the Queue branch and continues processing while the caller waits. As soon as an available agent is found, the eFlow returns to this step, and the ID of the selected agent is stored in the MB\_AGENT\_ID output parameter.
- 4. **PlayAnnouncement.** This Play Announcement step plays an announcement asking the caller to hold for the next available agent. While the caller waits, the step plays music, repeating the same announcement every 60 seconds. As soon as an agent becomes available, the eFlow returns to the Select Agent step.

#### **Recording an Announcement**

This eFlow example (Figure 22) can be used to record a Uniphi announcement in the case of an emergency. It plays an announcement with instructions, records the caller's voice, plays back the recording, saves the recording in an existing file that is then uploaded to the Media File Server. Finally, it plays an announcement saying that the recording has been saved. The media file that the recording is

stored in can either be one that is reserved for emergency use, or it can be a regularly used recorded that is replaced with the emergency recording.



#### Figure 22. A Record Announcement eFlow

The steps in this eFlow perform the following functions:

- 1. **LookUpResponse.** This Play Announcement step plays an announcement that instructs the caller to begin speaking after the tone.
- 2. **Record.** This Record step begins recording the caller's voice for up to 30 seconds or until the caller presses a termination digit. It stores the recording in an output parameter named Recorded\_Media\_File.
- 3. **Playback.** This Playback step plays the recorded message back to the caller.
- 4. **Save Recording.** This Save As step saves the recorded message in a file that is updated to the Media File Server, where it can be played by other eFlows.
- 5. **PlaySuccess.** The Play Announcement step plays an announcement to the caller that indicates the recording was made successfully.

#### Advanced IVR eFlow

This sample eFlow (Figure 23) combines Advanced IVR and database steps to collect a customer's ID number and use that number to access information about the customer in a database. It then offers the caller a menu of choices.



Figure 23. Advanced IVR eFlow

The steps in this eFlow perform the following functions:

- 1. **Connect\_VXML.** This Connect Telephony Server step connects Uniphi with a VXML adapter that offers the IVR functionality required by this eFlow. [PRECEDE WITH SET LANGUAGE STEP??]
- 2. **GetCustomerID.** This Play Form step plays a recording that asks the caller to enter a customer ID number. It stores the entered digits in an eFlow variable named customer\_id.
- 3. **GetCustInfo.** This Get First Row step uses the collected customer ID to look up information about the customer in an external customer database and stores that data in a structured variable named NewCustInfo. This info could be passed on to the receiving agent later in the eFlow.
- 4. **PlayMainMenu.** This Play Menu step asks the customer what they would like to do: Update their phone number, check their account balance, or speak with an agent. The direction that the eFlow takes depends on the digit the caller enters.
- 5. **UpdatePhoneNumber.** This output label jumps to a different (unshown) portion of the eFlow that prompts the caller for a new phone number.
- 6. **CheckBalance.** This output label jumps to an unshown portion of the eFlow that presents the caller with their current account balance.
- 7. **TransferToAgent.** This Transfer Task to Application step transfers the caller to another eFlow that will look for an available agent.



# Sample Routing eFlows

This chapter presents several sample eFlows that can be used to assist outside switches in the routing of contacts. It contains the following sections:

- Basic Routing eFlows presents examples of "push" and "pull" eFlows that can be used to connect incoming contacts with available agents.
- Advanced Routing eFlows includes examples of eFlows that incorporate the use of the Resource Matching Service (RMS) option. RMS greatly simplifies agent connection, allowing you to include both push and pull functions together in one eFlow and to take advantage of built-in skills-based routing facilities.

For information on routing calls between switch, see the next chapter, "Chapter 6, Interswitch Routing."

#### **Basic Routing eFlows**

In basic call (or contact) routing, Uniphi uses a combination of two types of eFlows to match a contact with an agent:

- Push eFlows are used to locate an agent for an incoming contact. When the Uniphi Contact Center receives a routing request from a switch adapter, it comes with a Task ID, which identifies the type of contact waiting on the switch. This could be an incoming call, a received e-mail, or a Web request. The UCC queues the task, then triggers a push eFlow to search the MMB database for a suitable agent to handle the task. If found, the eFlow passes the Agent ID and Task ID back to the adapter (via UCC) so the switch can make the connection. If no suitable agent is available, the task remains queued and the push eFlow is finished.
- Pull eFlows are triggered by an agent's state changing to Available. A pull eFlow looks in the virtual queue for a waiting task that is suitable for the available agent. If found, it pulls the task out of the queue and passes the Agent ID and Task ID back to the adapter so the switch can make the connection.

These two types of eFlows work in conjunction: one pushing a received task forward, another pulling it out of the queue. Together, they use routing logic and available data to determine the best agent/task match. For example, a push eFlow might prompt a caller to enter a digit corresponding to the department they want to connect with. A pull eFlow might search only for queued tasks of a certain priority, or e-mails with a specific subject line.

To help with the routing decision, eFlows can use:

- Data presented to Uniphi along with the contact, such as ANI, DNIS and customer-entered digits.
- Data (such as DTMF digits) collected by the eFlow itself, using Media Control steps.
- Data stored in a connected ODBC database.
- Data about the agents stored in the Uniphi Administrator database, such as the agent groups they are assigned to or their individual skills.

In addition, eFlows can execute algorithms that manipulate any of this data further, with the intention of determining the appropriate skill set required to match the task with the agent.

#### Simple Push eFlow

This very basic "push" eFlow is used to find an available agent for an incoming contact (voice call, e-mail, or Web request). If more than one agent is available, it selects the agent who has been available the longest. It then reserves the agent and passes the agent ID back to the switch where the contact is waiting. If no agent is available, the contact remains queued.

*Note:* Because this eFlow is appropriate for most types of switch adapters, it is a good one to use for testing, to ensure that an adapter is properly connected. When you start to create the eFlow, select an event specific to the adapter you want to test. This associates the eFlow with that event. Before you can run the eFlow, you must use the Uniphi Administrator to create event rules that tell the eFlow System under which circumstances to execute the eFlow when that event occurs. See the *Aspect Uniphi Suite Administrator Guide* for details on setting up event rules.


Figure 24. A Simple Push eFlow Sample

The steps in this eFlow perform the following functions:

1. **DefineExceptions.** This On Exception step jumps to the end of the eFlow if any of the specified exceptions occurs. It ensures that eFlow is completed even if there is a problem with the database query.

On Exception steps are often included near the beginning of an eFlow to set up a plan of action for handling various types of exceptions. If an exception occurs, the eFlow returns to the last On Exception step to determine what to do. For each type of exception, you can tell the eFlow to ignore the problem, jump to a given eFlow step, or halt eFlow execution.

- 2. **FindAvailableAgent.** This Get First Row step queries the MB.T\_MB\_AGENT table in the MMB database, where agent status is stored, searching for the agent who has been available the longest. It stores the agent's ID number in a structured eFlow variable named Result. If there are no available agents, it jumps to the end of the eFlow.
- 3. **ReserveAgent.** This Reserve Agent step reserves the agent identified by the previous step for the queued contact.
- 4. **StoreAgentID.** If an available agent is successfully reserved, this Set Variable step copies the Agent ID of the reserved agent from the Result variable to an output parameter (MB\_AGENT\_ID). This parameter will be passed back to the switch that initiated the routing event.
- 5. **ResetAgentID.** If the selected agent could not be reserved, this Set Variable step sets the Agent ID variable back to 0 (the default). This value is passed back to the routing service, indicating that no agent was found. The contact remains queued until an agent becomes available and a pull eFlow connects the two.

#### Simple Pull eFlow



This very basic eFlow is used to connect an available agent with a queued contact. It is triggered when an agent's state changes to Available.

Figure 25. A Simple Pull eFlow Sample

The steps in this eFlow perform the following functions:

- 1. **DefineExceptions.** This On Exception step jumps to the end of the eFlow if any of the specified exceptions occurs. It ensures that eFlow is completed even if there is a problem with the database query.
- 2. **FindQueuedTask.** This Get First Row step queries the T\_ASPT\_CALL table in the MMB database, where all Aspect call status data is stored, searching for the task that has been waiting the longest. This could be a queued call, for example, or an e-mail waiting to be answered. The step stores the task ID and the media ID in a structured variable named Result1. If there are no available tasks, it jumps to the end of the eFlow.
- 3. **ReserveTask.** This Reserve Task step reserves the task identified by the previous step for the agent whose availability triggered the eFlow.
- 4. **StoreTaskID.** This Set Multiple Variable step stores the reserved task's Task ID and Media ID in two separate parameters associated with the event. The Media ID identifies the switch adapter, so the eFlow engine knows where to send the information. The Task ID identifies which contact was waiting to be serviced.

#### Push eFlow with Task Escalation

This eFlow is like the simple push eFlow described on page 70, except that instead of queuing the contact if no agent is available (and requiring a push eFlow to make the connection), it sets task escalation parameters that will trigger another eFlow if no agent is found within two minutes. The new eFlow could

widen the search for an agent, for example, or give a contact the option of leaving a callback number.

This is a good example of a push eFlow that is used without a corresponding pull eFlow.



Figure 26. A Push eFlow with Task Escalation

The steps in this eFlow perform the following functions:

- 1. **DefineExceptions.** This On Exception step jumps to the end of the eFlow if any of the specified exceptions occurs.
- 2. **FindAvailableAgent.** This Get First Row step queries the MB.T\_MB\_AGENT table in the MMB database for the longest-available agent. It stores the agent's ID number in a structured eFlow variable named Result. If there are no available agents, it jumps to the SetEscalationVariables step.
- 3. **ReserveAgent.** This Reserve Agent step reserves the agent identified by the previous step for the queued contact.
- 4. **StoreAgentID.** If an available agent is successfully reserved, this Set Variable step copies the Agent ID of the reserved agent from the Result variable to an output parameter (MB\_AGENT\_ID), which will be passed back to the switch that initiated the route request.
- 5. **ResetAgentID.** If the selected agent could not be reserved, this Set Variable step sets the Agent ID variable back to 0 (the default), then jumps to the SetEscalationVariables step.
- 6. **SetEscalationVariables.** This Set Multiple Variable step sets two event parameters that are monitored by the eFlow Engine: Escalation Count and Escalation Time.
  - The Escalation Count parameter is incremented by one, indicating a single level of escalation. (Normally, the count will be 0 when this

eFlow executes.) The Escalation Count value is not used in this eFlow, but may be referenced by the next.

 The Escalation Time parameter is set to the current time plus 120 seconds. This tells the eFlow Engine to fire off a task escalation event two minutes from now. The eFlow triggered by that event depends on rules sets configured in the Uniphi Administrator.

### Advanced Routing eFlows

With an Advanced Routing license, you can take advantage of Uniphi's Resource Matching Service (RMS) option, which greatly simplifies call (or contact) routing. RMS integrates a new Select Agent step with agent pools and queues configured in the Uniphi Administrator to efficiently match tasks and agents.

Before you create an RMS eFlow in Architect, you must use Uniphi Administrator to do the following:

• Set up queues that represent categories for routing incoming tasks. For example, you could create queues that mirror your company's lines of business, such as Sales, Technical Support, and Billing. Or you could create queues for different skill groupings, such as languages spoken or product expertise.

Optionally, you can define routing parameters for the queues that help identify the attributes of an incoming task. For example, for the Sales queue, you could create a Product parameter. DNIS digits collected from the caller could be evaluated (using routing rules) to determine which product the caller is interested in.

• For each routing queue, set up a pool of agents designated to handling tasks for that queue. You can also define routing parameters for a pool to further identify the attributes or skills of agents within it. For example, with the Sales pool, you could create parameters that match those of the Sales queue, then use routing rules to link incoming tasks only with agents experienced with the product the customer is interested in.

Alternatively, you could create parameters that are entirely different from those of the associated queue. For example, you could use parameters that rank the experience level of agents to ensure that customers interested in your most expensive products are connected with your most experienced agents.

• Assign agents to pools. An agent can be in more than one pool. For example, an agent trained in both software and hardware technical support could be in added to the pools for each. Agents assigned to multiple pools must be given a ranking for each pool to determine where Uniphi first looks for a queued task.

When you assign an agent to a pool, you can specify a value to be passed to any of the routing parameters defined for that pool. For example, if the Sales queue has parameters for Appliances, Furniture, and Electronics, you could assign values that represent the agent's level of expertise in each.

• Create routing rules that associate each queue with a pool. A routing rule is an expression that the system evaluates when trying to best match a queued task with an available agent (or vice versa).

You can also define the sort order of a queue or pool. If there is more than one agent/task match that meets the conditions set forth by the routing rules, sort order determines the selection. A queue's sort order typically arranges tasks in order of arrival, so that those tasks that have been waiting the longest are given priority.

When an eFlow uses a Select Agent step to look for an available agent, it directs the task to a specific RMS queue. Uniphi looks for an agent in the associated pool that meets the criteria specified in routing rules assigned to the queue. If no routing rules are defined, it uses the sort order defined for the queue or pool, such as longest-waiting or longest-available.

If an appropriate agent is available, the eFlow releases the task to the routing service, which sends the routing information back to the switch where the original contact is parked. The switch then connects the selected agent and the contact.

If no agent meeting the criteria is available, the eFlow places the task in the routing queue and lets the routing service handle the connection. In the meantime, the eFlow takes the Select Agent step's Queued branch and continues processing steps. When an agent becomes available and the task is connected, the routing service sends a message indicating this to the eFlow System. The eFlow returns to the Select Agent step, this time taking the Success branch.

For more information on using RMS, refer to the *Unipbi Suite Product Reference Guide*. For details on configuring RMS queues, pools, parameters, and routing rules, refer to the *Unipbi Suite Administrator Guide*.

#### A Simple RMS eFlow

With RMS, the search for an available agent is done using routing rules in the Uniphi Administrator. All resource matching configuration is also done within the Administrator. This greatly simplifies the role of the eFlow in routing tasks. In fact, an RMS eFlow can be as simple as a single Select Agent step.

This sample RMS eFlow takes the place of two much lengthier push and pull eFlows. It looks for the longest available agent within a given queue. If an appropriate agent in the associated pool is available, the contact is connected. Otherwise, the task is queued until an available agent can be found. In this very



simple eFlow, the eFlow is finished, regardless of whether the task is queued or immediately connected.

Figure 27. An Extremely Simple RMS eFlow Sample

### Two-Tiered Agent Search eFlow

This sample eFlow (Figure 28) searches first for an agent in a specialized queue. If none is found within a minute, it uses a second Select Agent step to search for an agent in a wider queue.



Figure 28. A Two-Tiered Agent Search eFlow

The steps in this eFlow perform the following functions:

- 1. **SelectAgent1.** This Select Agent step looks for an available agent in queue 122. If none is available, it queues the contact. If no agent is found within five seconds, the eFlow goes to the second Select Agent step.
- **2. Sleep 1.** This Sleep step pauses the eFlow while the contact is queued. After ten seconds, the eFlow returns to the SelectAgent1 step and takes the Timed Out branch.

3. **SelectAgent2.** This Select Agent step looks for an available agent in queue 121. If none is available, the eFlow finishes.

#### Selecting an Agent by Extension

It is often desirable to allow incoming voice contacts to connect directly to a given extension. For example, a customer may want to consult with the support specialist they spoke with previously. With the Select Agent step, you can use routing rules and parameters to specify an individual agent to select.

Before you create an eFlow to select an agent by extension number, set up the following in the Uniphi Administrator:

- **Parameter.** Define a routing parameter for passing the agent extension number between the eFlow and Uniphi, for example AGT\_EXT. Assign a data type of Long. (Use String if agent extensions in your company are alphanumeric.)
- **Pool.** Define a pool of agents to search through for requested agents. Add the AGT\_EXT parameter to this pool.
- Queue. Define a queue in which to hold contacts waiting for a specified agent, for example AGT\_EXT\_QUEUE. Add the AGT\_EXT parameter to this queue.
- **Routing Rule.** Create a routing rule that associates the AGT\_EXT pool with AGT\_EXT\_QUEUE. Attach the condition AGT\_EXT EQ AGT\_EXT. This will only look for agents in the pool whose extension matches that in the queue parameter.
- Agent Assignment. Assign agents to the AGT\_EXT pool. For each agent to whom you want customers to be able to access by extension, add AGT\_EXT to list of associated routing pools and add AGT\_EXT to the list of routing parameters. Enter the agent's extension number as the fixed value of AGT\_EXT.

For details on all these administrative tasks, refer to the *Uniphi Suite 6.0 Administrator Guide* or the online Help within the Administrator.

After the required configuration is completed, you can use eBusiness Architect to create an eFlow that uses all these routing pieces to select an agent by extension number. Following are two example eFlows: one for use with a TAPI voice switch, and another for use with an Aspect voice switch.

#### Select Agent by Extension eFlow (TAPI Switch)

This example eFlow uses media control steps to communicate with the caller. It offers the caller two options: enter an extension to speak with a specific agent or

speak with any available agent. If the caller enters an extension, the eFlow attempts to select that agent for connection. If that agent is unavailable (or if the caller didn't enter a valid extension), the eFlow looks for any available agent in the general queue.



#### Figure 29. An eFlow that Searches for an Agent by Extension (using TAPI)

The steps in this eFlow perform the following functions:

- 1. **Prompt1.** This Prompt step plays an announcement that gives the caller two options: speak with a specific agent or connect to any agent. If the caller requests a specific agent, the eFlow goes to the Collect1 step. If the caller asks for any agent, the eFlow jumps to the SelectAgent2 step.
- 2. **Collect1.** This Collect step plays an announcement that asks the caller to enter an extension number.
- 3. **SetVariable1.** This Set Variable step stores the digits collected with the previous step in a local eFlow variable named AGT\_EXT.
- 4. **SelectAgent1.** This Select Agent step directs the call to routing queue 102 and binds the local variable containing the extension (AGT\_EXT) to the queue's parameter named AGT\_EXT\_QUEUE. Routing rules defined for the queue look for an agent in the associated pool with a matching extension. If that agent is unavailable, it queues the contact. If the agent does not become available within 10 seconds, the eFlow goes to the SelectAgent2 step.
- 5. **Sleep1.** This Sleep step waits while the contact is queued. This ensures that the eFlow does not finish before an agent is selected or the request has timed out.

6. **SelectAgent2.** This Select Agent step searches the general queue (101) for any available agent. If none is available, it queues the contact until one is.

#### Select Agent by Extension eFlow (Aspect Switch)

With an Aspect Call Center switch, communication with callers is done on the Call Center side with a CCT. This sample eFlow assumes that a prior CCT has prompted the caller for an agent extension number and that the number has been stored in Variable A. This eFlow looks up the extension number in the call data table of the MMB database and tries to select that agent. If unsuccessful, it looks for any available agent in the general queue.



#### Figure 30. An eFlow that Searches for an Agent by Extension (using Aspect Voice)

The steps in this eFlow perform the following functions:

- Get\_Agent\_Extension. This Get First Row step queries the MMB database, where all call data is stored. It retrieves the customer response digits (presumably an agent extension) stored in Variable A for this task (T\_ASPT\_CALL.DATA\_A) and copies it to the eFlow variable named Result. If this step is unable to retrieve the extension number, the eFlow jumps to the SelectAgent2 step.
- 2. **Valid\_Agent\_Extension.** If the previous step successfully retrieved customer digits, this Set Variable step stores those digits in a local eFlow variable named VAR\_AGT\_EXT.
- 3. **SelectAgent1.** This Select Agent step directs the call to routing queue 107 and binds the local variable containing the extension (VAR\_AGT\_EXT) to the queue's parameter named AGT\_EXT\_QUEUE. Routing rules defined for

the queue look for an agent in the associated pool with a matching extension. If that agent is unavailable, it queues the contact. If the agent does not become available within 10 seconds, the eFlow goes to the SelectAgent2 step.

- 4. Wait1. This Wait step plays music while the contact is queued.
- 5. **SelectAgent2.** This Select Agent step searches the general queue (108) for any available agent. If none is available, it queues the contact (for up to one hour) until one is.

### Select Agent Group eFlow

This RMS eFlow example (Figure 31) routes an incoming contact to an agent group based on customer input. It assumes that prior to this eFlow (in a call center CCT, for example), the contact was presented with a menu choice, such as "Press 1 for TV Sales or 2 for Stereo Sales." This eFlow looks up the customer response in the call data table of the MMB database and directs the call accordingly.

Before you can create this eFlow in Architect, in addition to standard RMS configuration, you must have also defined agent groups in the Uniphi Administrator and assigned agents to those groups.



Figure 31. A Select Agent Group eFlow

The steps in this eFlow perform the following functions:

1. **LookUpResponse.** This Get First Row step queries the MMB database, where all call data is stored. It retrieves the customer response digit stored in Variable A (T\_ASPT\_CALL.DATA\_A) and copies it to the eFlow variable named CustChoice.

- 2. **DetermineChoice.** This Case step looks at the value stored in CustChoice and determines whether the contact requested TV sales or Stereo sales, then branches accordingly.
- 3. **TV Sales**. This Set Variable step sets the eFlow variable AgtGrp to the value 501, which is the number of the agent group designated to handling TV sales.
- 4. **Stereo Sales.** This Set Variable step steps the eFlow variable AgtGrp to 502, which is the number of the agent group designated to handing Stereo sales.
- 5. **GoToSalesQueue.** This Select Agent step routes the call to the RMS Sales queue (103) and passes along the contents of the eFlow variable AgtGrp. Routing rules associated with the Sales queue will use the variable contents (set in the previous step) to determine which agent group the routing service will look in for an available agent.

#### Skill-Based Routing

With Uniphi's RMS (Resource Matching Service) option, you can use pools and routing parameters to align agents with specific skills, such as proficiency in various languages, or expertise on given products. You can then use data collected from contacts or extracted from a database to select an agent suited to handle the contact's needs.

The example presented here searches for an agent with skills to match the preferred language (Spanish or English) and the level of expertise required for the customer account. Before you create this sample eFlow, set up the following in Uniphi Administrator first:

- **Parameters.** Define two routing parameters with Long data type: Skill\_Level\_Required, and Skill\_Level and two routing parameters with Boolean data type: English and Spanish. These will be used to direct a contact to an English- or Spanish-speaking agent with the appropriate skill level.
- Pools. Define three pools of agents: Checking\_Savings\_Pool, CreditCards\_Pool, and Loans\_Pool. Add the English, Spanish, and Skill\_Level parameters to each of these pools.
- Queues. Define three queues to which contacts will be routed: Checking\_Savings\_Queue, CreditCards\_Queue, and Loans\_Queue. For each queue, add the English, Spanish, and Skill\_Level\_Required parameters.
- **Routing Rules.** Create routing rules that pair each pool with a similarly named queue. For the Loans pool/queue pair, use following condition:

Loans\_Pool.English EQ Loans\_Queue.English OR Loans\_Pool.Spanish EQ Loans\_Queue.Spanish AND Loans\_Pool.Skill\_Level GE Loans\_Queue.Skill\_Level\_Required

Create similar conditions for the other two pairs. This will only look for agents in the pool whose language setting matches that requested by the contact *and* whose skill level is greater than or equal to the level determined by the eFlow (based on info gathered from the customer database and length of time in queue).

For each pair, add a Pool Order By clause that orders available agents first by skill level, then by the length of time they have been available. For example, for the Loans pool/queue pair, use:

Loans\_Pool.Skill\_Level ASC, Loans\_Pool.RMS\_TIMESTAMP ASC

Also add a Queue Order By clause that orders waiting contacts first by requested skill level, then by length of time the contact has been in queue, for example:

Loans\_Queue.Skill\_Level\_Required DESC, Loans\_Queue.RMS\_TIMESTAMP

■ Agent Assignment. Assign agents to the pools, according to the line of business they are qualified to handle. An agent can be assigned to more than one pool. For each agent record, add the parameters English, Spanish, and Skill\_Level. Set English and Spanish to True or False, and set Skill\_Level to a value (0-25) that reflects the agent's ability to handle loan customers.

Each of these tasks is described step by step in Appendix A of the *Aspect Uniphi Suite Product Reference Guide*. This appendix covers all aspects of this same RMS example in detail, including the administration configuration.

The example eFlow shown here takes advantage of the described RMS configuration to direct callers to an agent with the appropriate skills in the selected line of business. It starts by asking callers for a language choice: English or Spanish, then prompts for an account number. It looks up the account in a customer database and prompts for a line of business (Checking and Savings, Credit Cards, or Loans). If the user chooses Loans, the eFlow sets a Skill Level variable based on the customer's current loan amount, then looks for an agent in the Loans queue whose assigned agent skill is at least the requested level and who is proficient in the selected language.



Figure 32. Skills-Based Routing eFlow (Section 1)

The steps in this eFlow perform the following functions:

**Language?** This Collect step plays an announcement that welcomes the caller and presents two options: Press 1 to speak with an agent in English, or press 2 to speak with an agent in Spanish. It stores the digit pressed by the caller in an eFlow variable named Var\_Language.

**SetLanguageVariables.** This Set Multiple Variables step uses Boolean expressions to set two eFlow variables: EnglishReq and SpanishReq. If the caller responded to the prompt by pressing 1, EnglishReq is set to True and SpanishReq is set to False. If the caller pressed 2, EnglishReq is set to False and SpanishReq is set to True.

**AccountNumber?** This Collect step plays an announcement requesting the caller to enter an account number. It then stores the collected digits in a variable named Var\_Account\_No.

**GetAccountNumber.** This Get First Row step looks up the collected account number in an outside customer database. It copies the balance, credit limit and loan amount values for that account in a structured variable named Result, with three data members: BALANCE, CREDIT\_LIMIT, and LOAN\_AMOUNT.

**Division?** This Prompt step plays an announcement asking the caller which line of business they would like to connect with: "Press 1 for Checking and Savings, 2 for Credit Cards, or 3 for Loans." The eFlow then jumps to the appropriate input label, depending on the digit pressed.

Figure 33 shows the logic for the Loans portion of the eFlow. Based on the customer's current loan amount, it sets a skill level variable that is used to determine the level of expertise required of the selected agent. If no agent of that level is found within two minutes, the skill level is lowered by five points and the search for an agent is begun again.

The other portions (CheckingAndSavings and CreditCards) could use a similar logic to the one shown here, using balance and credit limit values to determine required skill level.



Figure 33. Skills-Based Routing eFlow (Section 2)

The steps in this eFlow perform the following functions:

**LoanAmount.** This If step looks at the value stored in the LOAN\_AMOUNT field of the Result variable. If it is greater than 15,000, the eFlow takes the True branch (to SkillLevel=High). Otherwise, it takes the False branch (to SkillLevel=Medium).

**SkillLevel=High.** This Set Variable step sets the eFlow variable named SkillLevelofCall to 15. This variable will be used to determine the required level of expertise of the selected agent.

**SkillLevel=Medium.** This Set Variable step sets the eFlow variable named SkillLevelofCall to 10.

**SelectAgent.** This step looks for an available agent in the Loans queue whose assigned skill level is equal to or greater than the value stored in the SkillLevelofCall eFlow variable and whose language settings (English and Spanish) match that requested by the caller. If found, the step takes the Success branch and the eFlow is finished. If no available agent meeting the requested criteria is found, the contact is queued and the step takes the Queued branch.

**Sleep.** This step pauses eFlow execution for 180 seconds while waiting for an available agent. (If an agent becomes available within this time, the eFlow finishes immediately.)

**SkillLevel>0?** This If step checks the value stored in the SkillLevel parameter. If it is greater than 0, the eFlow moves to the LowerSkillLevel step.

**LowerSkillLevel.** This Set Variable step lowers the value in the SkillLevelofCall variable by five. The eFlow then loops back to the SelectAgent step to search for an agent with a lower skill level.

The following figures show what the property sheets of the Select Agent step should look like.

<u>√</u> ork Item	CONN_ID		
ueue Parameters			
Name	Туре	Value	Bind Varia.
WORK_TYPE_ID	LONG	0	<none></none>
Spanish	BOOLEAN		SpanishR.
Skill_Level_Required	LONG		SkillLevel.
English	BOOLEAN	-	EnglishRe
AGENT_GROUP_ID	LONG	0	<none></none>
•			
			<u>E</u> dit

Figure 34. General Properties for the Select Agent Step

Select Age	nt Properties	? ×
General	Standard Parameters Response Caption	
S	andard Request Parameters Priority Timeout C Specify ⊻alue C Specify gFlow Variable C Specify gFlow Variable	
	OK Cancel Apply	Help

Figure 35. Standard Parameters Properties for the Select Agent Step

Select Agent Properties	? ×
General Standard Parameters	Response Caption
Output Variables <u>R</u> equest Identifier	
<u>Ag</u> ent Id	MB_AGENT_ID
<u>I</u> ask Id	
OK	Cancel Apply Help

Figure 36. Response Properties for the Select Agent Step



## Interswitch Routing

This chapter describes how to use Uniphi to route calls among multiple switches. It contains the following sections:

- Interswitch Routing Concepts explains the basic setup of an interswitch routing environment and presents key concepts involved, including output parameters, input parameters, and correlation keys.
- Interswitch Example for Aspect Adapter presents a basic interswitch routing eFlow used to route calls coming into an Aspect ACD, along with sample inbound and outbound CCTs required on the ACD side.
- Interswitch Example for Cisco CallManager presents a sample interswitch routing eFlow used to route calls coming into a Cisco CallManager and gives details on required configuration on the Cisco CallManager, Uniphi Administrator, and Gateway.

### Interswitch Routing Concepts

If two or more call centers are physically connected with a TIE line, Uniphi can be used as an interswitch router, directing contacts presented to one switch to an agent or resource residing on another switch. Uniphi supports interswitch routing in both homogeneous deployments (all Aspect Call Centers) and heterogeneous deployments (a mix of Aspect Call Centers and Cisco Call Managers). Figure 38



shows a heterogeneous deployment in which one Aspect Call Center and one Cisco Call Manager are connected via adapters to a Uniphi server.

Figure 37. Interswitch Routing Connections

In an interswitch environment, agents on connected call centers are defined in Uniphi as one large resource set, with only a Media ID parameter identifying each agent's physical location. When a call or task is presented to a call center, the call center passes a route request to Uniphi, which then executes an eFlow to identify the most suitable agent for the task. If that agent is on a different switch than the one that received the call, the eFlow returns the task to the originating switch, along with the information required to route the contact to the remote agent.

Uniphi can accomplish interswitch routing with basic (push/pull) routing or with advanced (RMS) routing. With RMS, the Select Agent eFlow step determines the location of the selected agent. If the selected agent is on the same switch as the call, the Success branch is taken. If not, the Remote Agent branch is taken. In a traditional Push/Pull eFlow, once a suitable agent is identified (using Database steps), additional Database steps must determine whether or not the call and the selected agent are on the same media switch (comparing the agent's Media ID parameter with the MEDIA\_ID input parameter associated with the route request).

In either case, there are three essential resources used by an interswitch eFlow:

- Output parameters
- Input parameters
- Correlation keys

Output parameters are used to relay interswitch information to the Uniphi server. Input parameters contain information passed over from the switch initiating the eFlow event. Correlation keys are defined in the Uniphi System Administrator and are used to associate data with a transferred call.

#### Interswitch Output Parameters

The following four output parameters are used in interswitch routing. These parameters are provided as standard variables with any eFlow associated with a work request event for Aspect Portal Multimedia.

- ROUTE\_TYPE (Integer, Output only) specifies what kind of routing is required:
  - 0, the default, indicates that the selected agent is on the originating switch.
  - 1 indicates that the selected agent is remote.
  - 2 indicates that the call has been routed from another switch and an agent has already accepted the task.

The routing subsystem checks this parameter to determine what action to take. If it is 1, Uniphi tells the adapter to route the call to the destination switch. If it is 2, Uniphi tells the adapter to connect the call with the agent.

- ROUTE\_POINT (String, Output only) is used by the originating switch adapter to route the call to the remote switch. If the destination switch is an Aspect Call Center, this should be set to the number of a CCT used to route the call to the remote switch. If the destination switch is a Cisco CallManager, specify the Route Point number to use. If an outdial prefix is required to place an outside call, include it with the Route Point number. For example, 91001 includes the outdial prefix 9, followed by the Route Point number 1001.
  - *Note:* If there are only a few media switches involved, the CCT or Route Point number can be hard-coded in the eFlow. Otherwise, it may be necessary to maintain a table of CCTs or Route Points used in interswitch routing, then have the eFlow query the table to set the proper ROUTE\_POINT value.
- ALLOC\_DNIS (String, Input/Output) stores a correlation key assigned by Uniphi in response to an Allocate Correlation Key step. This key is used to transfer data with the call as it moves between switches. If the destination switch is an Aspect Call Center, this key relays the DNIS number of the trunk line to use to make the transfer. For a Cisco CallManager, the key refers to the CTI Route Point (not to be confused with the ROUTE\_POINT eFlow variable above).
- MB\_AGENT\_ID (Long, Input/Output) stores the agent ID of the selected agent. If the selected agent is remote, a Set Repository step stores the value in this parameter so it can be retrieved by the destination eFlow after the call is transferred.

An interswitch eFlow must set all these output parameters appropriately to successfully transfer calls among switches.

### Interswitch Input Parameters

In addition to the output parameters described in the previous section, interswitch routing relies on two input parameters passed over from the switch:

- ROUTE\_NUMBER is included with calls arriving at a Cisco CallManager. It stores the CTI Route Point number, which is where calls arrive on the switch.
- DNIS is included with calls arriving at an Aspect ACD. It stores the DNIS number associated with the trunk line the call came in on.

An interswitch routing eFlow examines these parameters to determine whether a call has been transferred from a different switch. For example, for an Aspect ACD call, the eFlow uses an If step to see if the number in the DNIS parameter falls within the range of DNIS numbers associated with trunks used to transfer calls. For Cisco CallManager, the If step would see if the ROUTE\_NUMBER value matches the current switch's CTI Route Point number.

### **Correlation Keys**

A correlation key is a unique value assigned to a task, which can be used to store data related to the task in a repository database.

Correlation keys must first be defined in the Uniphi System Administrator. There you assign a range of correlation key values to a set ID. You can define two types of correlation keys: rolling and non-rolling.

- With rolling keys, Uniphi allocates keys in sequence within the given range. When it reaches the end of the range, Uniphi cycles back to the beginning of the range, deallocating and reusing numbers as necessary.
- Non-rolling keys remain reserved until they are deallocated (with a Deallocate Correlation Key step) or until a timeout value (configured in the Administrator) is exceeded. If Uniphi receives an allocate correlation key request and no keys are available, the request is denied.

When an eFlow executes an Allocate Correlation Key step, it requests a key assignment from a specified set ID. Uniphi assigns the lowest available key in the sequence. The eFlow can use this key to store data in a repository.

In interswitch routing, a correlation key assigns an entry point for the destination switch. Before you create an interswitch eFlow, you must set up correlation key ranges in the Uniphi Administrator (Systems folder) for each switch adapter:

- For an Aspect Adapter, create a set of keys for each group of trunks used to connect to the Aspect ACD. Assign to those keys a range of numbers that match the DNIS (Dialed Number Identification Service) of the trunks.
- For a TAPI Adapter, create a set of keys that correspond to CTI Route Points used to connect to the Cisco CallManager.

When an interswitch eFlow uses an Allocate Correlation Key step to get a correlation key, it is actually procuring the number of an available entry point to the remote switch. So the number of correlation keys you set up in Uniphi Administrator reflects the maximum number of simultaneous interswitch calls the system can handle.

An interswitch eFlow uses a correlation key to store the selected agent ID in the data repository, along with the DNIS number or Route Point needed to connect to the destination server.

You can use either rolling or non-rolling correlation keys with interswitch routing. Rolling keys require very little computing resources in comparison to non-rolling keys and offer the most efficient method of allocation. However, it may be necessary to use non-rolling keys under certain circumstances. For example, the key range for DNIS numbers is often very small. With smaller key ranges, you want to be sure Uniphi has time to retrieve data stored with a key before it is recycled.

### Interswitch Example for Aspect Adapter

In this example situation, Uniphi is used as an interswitch router for two Aspect ACDs. Agents from both ACDs are defined in the Uniphi System Administrator, with a Media ID parameter identifying the switch on which the agent is physically located. A set of correlation keys is defined in the Administrator, under the set ID "Aspect2Aspect." The range of keys defined (5000-5080) matches the range of DNIS numbers associated with trunks used to transfer calls between the call centers.

#### Sample Interswitch eFlow

The interswitch eFlow shown here (Figure 38) is intended to field route requests from the both ACDs. It first determines if the presented call is new or if it has been transferred from the other ACD.

 If the call is a new one, a Select Agent step searches for a Uniphi agent to handle the call. If the selected agent is remote, the eFlow requests a correlation key for the call, then uses that key to store the selected agent ID in the repository database. Finally, it sets output parameters that tell the originating ACD to use a predetermined CCT to route the call to the other ACD.

If the call is a transferred one, the eFlow looks in the repository database for the agent ID correlated with the call and stores it in the MB\_AGENT\_ID output parameter so the routing subsystem can offer the call to the agent.



Figure 38. Interswitch Routing eFlow

The steps in this eFlow perform the following functions:

- 1. **Ignore Exceptions.** This On Exception step tells the eFlow to ignore any exceptions that occur. In a more comprehensive eFlow, this step might jump to a subflow that handles specific types of exceptions.
- 2. **TransferredCall?** This IF step checks the DNIS input parameter associated with the call. If the DNIS value is within a range that is reserved for

interswitch calls, the eFlow takes the True branch. For all other numbers, the eFlow jumps to the New Call input label and the eFlow attempts to find an agent for the call.

- **3. GetCallData.** This Get Repository Data step looks in the repository database for the agent ID stored with the call's correlation key and stores that value as a string in the StoredAgt eFlow variable. If no agent ID is found, the step branches to the No Agent step.
  - *Note:* Because the allocated correlation key is a DNIS number that is used to transfer the call, the DNIS input parameter value and the correlation key value are the same.
- 4. **SetOutputParameters.** This Set Multiple Variables step sets the ROUTE\_TYPE parameter to 2 (indicating that an agent has been found for the call) and the MB\_AGENT\_ID parameter to the retrieved agent ID (converting the ID from a string to a long value).
- 5. **NoAgent.** This Set Variable step sets the MB\_AGENT\_ID output parameter to 0, indicating that no agent ID was found.

If the call is new (not transferred), the following steps are taken:

- 6. **SelectAgent.** This step searches for an available agent in Queue 103 using criteria set up for the queue in the Uniphi System Administrator. If no agent is immediately available, the Queue branch is taken until either an agent is found or the step times out, at which point the eFlow returns to the SelectAgent step. If the selected agent is local (the Media ID set for the agent in the Administrator matches that of the switch that received the call), the Success branch is taken. If the selected agent is on a different switch, the Remote Agent branch is taken (see step 8).
- 7. **Wait.** This step pauses eFlow execution while the caller is queued for an agent.
  - *Note:* With RMS, it is important that the eFlow does not reach a Finish step before an agent is selected. For local routing, the Select Agent step queues the contact if no agents are available and lets the routing system handle the connection when an agent becomes available. So the connection can be made after the eFlow has finished. With interswitch routing, however, this is not possible. Use a Wait step or other loops, if necessary, to ensure sufficient time for agent selection.

If the selected agent is remote, the following steps are taken:

8. **AllocateKey.** This Allocate Correlation Key step requests that the system assign a rolling correlation key to the call, using a value within the range defined for the Aspect2Aspect key set in the Uniphi System Administrator. It stores the returned key number in the eFlow variable ALLOC\_DNIS. The

number will be used later by a CCT on the originating ACD to transfer the call to the other call center.

- 9. **StoreRepositoryData.** This Set Repository Data step stores the ID of the agent selected by the SelectAgent step in the repository database under the key allocated to the call (stored in ALLOC\_DNIS).
- 10. **SetRouteParameters.** This Set Multiple Variables step sets the output parameters as follows:
  - ROUTE\_TYPE is set to 1 to indicate that the selected agent is remote.
  - ROUTE\_POINT is set to 354, which is the number of the CCT that will be used by the originating ACD to transfer the call to the target ACD.

#### Aspect ACD Configuration

In order for an Aspect ACD to transfer calls to another call center, there must be a TIE line physically connecting the two, with at least one CCT dedicated to routing calls to the remote switch. CCTs handling inbound calls must also be altered to accommodate interswitch routing.

#### **Outbound CCTs**

When Uniphi selects a remote agent, it passes the agent ID back to the originating switch, along with the number of a CCT to use to route the call to a different switch. The CCT typically reserves a trunk associated with the TIE line, passes the necessary DNIS numbers, and connects the call.

The sample CCT in Figure 39 does just that. In this example, the trunk group named Cisco CM (3) is associated with the TIE line that connects the Aspect ACD and the Cisco Call Manager. This CCT selects a trunk in that trunk group, dials the

DNIS digits stored in ALLOC\_DNIS parameter (transferred into Variable A), and connects the call to the Cisco CallCenter.



Figure 39. Sample CCT for Transferring Calls to a Cisco CallManager

The steps in this CCT perform the following functions:

- 1. **Select1.** This Select step secures a trunk on the TIE line connecting the Aspect ACD with the target switch.
- 2. **Dial1.** If the Select1 step is successful, this Dial step dials the digits stored in Variable A. These are the DNIS digits that the interswitch eFlow procured with the Allocate Correlation Key step and stored in the repository database, paired with the Agent ID.
- 3. **Wait1.** If there are no available trunks in the selected trunk group, this Wait step plays music to the caller while the call is on hold. As soon as a trunk becomes available, the eFlow returns to the Select1 step and takes the Success branch.
- 4. **Connect1.** This Connect step connects the call to the selected trunk. The call is then transferred to the target switch.

#### Inbound CCTs

The sample CCT in Figure 40 asks the caller to press 1 to speak with someone in customer service or 2 to speak with someone in the branch office. It then attempts to find an available agent in the requested agent group. However, in a Uniphi setting, all agent TeleSets should be set to Idle, allowing Uniphi to determine agent availability. So the call is automatically queued while Uniphi searches for an available agent. When an agent is found, Uniphi responds with an API call that includes the ID of the agent to connect the call to (if local) or the number of a CCT to use to transfer the call (if the selected agent is remote). To support interswitch routing, it is essential that an inbound CCT incorporate a Receive Data step to wait for this response from the Uniphi server.



Figure 40. Sample Inbound CCT for Interswitch Routing

The steps in this CCT perform the following functions:

- 1. **Prompt1.** This step plays a welcome announcement that asks the caller to press 1 for customer service, or press 2 to speak with someone in the branch office.
- **2. Select1.** If the caller presses 1, this step looks for an available agent in Agent Group 2 (Customer Service). Because with Uniphi, all agent TeleSets are set to Idle, the call is normally queued. This initiates a RouteRequest event, which triggers an eFlow in Uniphi to search for an available agent, and the In Queue branch is taken.
- 3. **Select 2.** If the caller presses 2, this step looks for an available agent in Agent Group 3 (Branch Office). As noted above, the call is normally queued, a RouteRequest event triggers an eFlow to search for an agent, and the In Queue branch is taken.
- **4. PlayAnnouncement.** This step plays an announcement to the caller, asking them to wait for the next available agent.
- 5. ReceiveData. This step waits for a response from Uniphi before continuing. If Uniphi finds a local agent for the call, it responds with an API call (CTI\_Connect\_Call) that includes the call ID and the agent ID. If the selected agent is remote, Uniphi responds with an API call (CTI\_Send\_Call\_Information\_Response) that includes the call ID and the number of a CCT to use to transfer the call.
- 6. **Connect1.** This step connects the call either to an agent (if local) or to a CCT used to transfer the call to another call center (such as that shown in Figure 39).

### Interswitch Example for Cisco CallManager

In this example situation, Uniphi is used as an interswitch router for two call centers: a Cisco CallManager and an Aspect ACD. Agents from both ACDs are defined in the Uniphi System Administrator, with a Media ID parameter identifying the switch on which the agent is physically located. A set of correlation keys is defined in the Administrator, under the set ID "CM2ACD." The range of keys defined matches the range of DNIS numbers used to transfer calls from the CallManager to the Aspect ACD.

#### Sample Interswitch eFlow

The interswitch eFlow shown here (Figure 41) is intended to field route requests from the Cisco CallManager. It first determines if the presented call is new or if it has been transferred from the Aspect ACD.

- If the call is a new one, a Select Agent step searches for a Uniphi agent to handle the call. If the selected agent is remote, the eFlow requests a correlation key for the call, then uses that key to store the selected agent ID in the repository database. Finally, it sets output parameters that tell the adapter to transfer the call to the given Route Point.
- If the call has been transferred in from the Aspect ACD, the eFlow looks in the repository database for the agent ID correlated with the call and stores it in the MB\_AGENT\_ID output parameter so the routing subsystem can offer the call to the agent.



Figure 41. Interswitch Routing eFlow for Cisco CallManager

The steps in this eFlow perform the following functions:

- 1. **TransferredCall?** This IF step checks the ROUTE\_NUMBER input parameter associated with the call. If it is 9514 (the number of the Route Point used to access the current switch), Uniphi takes the True branch (see step 8). Otherwise, Uniphi assumes the call has been transferred from another switch and jumps to the Incoming Transferred Call portion of the eFlow.
- **2. PlayAnnouncement.** This step greets the caller with a welcome announcement.

- 3. **SelectAgent.** This Select Agent step searches for an available agent in Queue 101 using criteria set up for the queue in the Uniphi System Administrator. If no agent is immediately available, the Queue branch is taken until either an agent is found or the step times out, at which point the eFlow returns to the SelectAgent step. If the selected agent is local (the Media ID set for the agent in the Administrator matches that of the switch that received the call), the Success branch is taken and the eFlow is finished. If the selected agent is on a different switch, the Remote Agent branch is taken (see step 5).
- 4. **Wait.** This Wait step pauses eFlow execution while the caller is queued for an agent.
  - *Note:* With RMS, it is important that the eFlow does not reach a Finish step before an agent is selected. For local routing, the Select Agent step queues the contact if no agents are available and lets the routing system handle the connection when an agent becomes available. So the connection can be made after the eFlow has finished. With interswitch routing, however, this is not possible. Use a Wait step or other loops, if necessary, to ensure sufficient time for agent selection.

If the selected agent is remote, these steps are taken:

- 5. **AllocateKey.** This Allocate Correlation Key requests that the system assign a rolling correlation key to the call, using a value within the range defined for the CM2ACD key set in the Uniphi System Administrator. It stores the returned key number in the eFlow variable ALLOC\_DNIS. The number will be used later by the originating Cisco CallManager to transfer the call to an Aspect ACD.
- 6. **StoreRepositoryData.** This Set Repository Data step stores the ID of the agent selected by the SelectAgent step in the repository database under the key allocated to the call (stored in ALLOC\_DNIS).
- 7. **SetRouteParameters.** This Set Multiple Variables step sets the output parameters as follows:
  - ROUTE\_TYPE is set to 1 to indicate that the selected agent is remote.
  - ROUTE\_POINT is set to the value stored in ALLOC\_DNIS (in step 5).
     This is the DNIS number that Cisco CallManager will use to transfer the call to the target ACD.

If the call has been transferred in from another call center, these steps are taken:

**8. GetRepositoryData.** This step looks in the repository database for the agent ID stored with the call's correlation key and stores that value as a string in the StoredAgt eFlow variable. If no agent ID is found, the step branches to the No Agent step.

- 9. **SetOutputParameters.** This Set Multiple Variables step sets the ROUTE\_TYPE parameter to 2 (indicating that an agent has been found for the call) and the MB\_AGENT\_ID parameter to the retrieved agent ID (converting the ID from a string to a long value).
- 10. **NoAgent.** This Set Variable step sets the MB\_AGENT\_ID output parameter to 0, indicating that no agent ID was found.

#### Configuring Interswitch Routing for Cisco CallManager

In order for Interswitch routing to work with a Cisco CallManager, three different components much be configured: the Cisco CallManager itself, the IP-enabled Gateway that the CallManager uses to communicate with non-IP telecommunication devices (such as an Aspect ACD), and the Uniphi Administrator. This describes offers basic configuration information for all three components.

#### Cisco CallManager Configuration

If the Gateway has not already been defined for Cisco CallManager, follow these steps to add it to the Cisco CallManager database:

- 1. Add a Gateway device to the Cisco CallManager database (Device→ Gateway).
  - *Note:* If interswitch routing is between two remote Cisco CallManager clusters, instead of adding the Gateway device, go to Device→ Trunk and add the intercluster trunk used to connect the two clusters.
- 2. Create a route group and add the Gateway device created in step 1 (Route Plan $\rightarrow$  Route Group).
- 3. Create a route list and add the route group created in step 2 (Route Plan $\rightarrow$  Route List).
- 4. Create a route pattern for a route list created in step 3 (Route Plan→ Route Pattern). Table 3 shows an example. Make sure the route pattern matches that used with the ROUTE\_POINT eflow variable (see Interswitch Output Parameters on page 89).
- 5. Create a CTI Route Point for each interswitch connection (Device→ CTI Route Point).

For details on Cisco CallManager configuration, refer to the Cisco CallManager Administration Guide.

Table 3. Sample Route Pattern

Route Pattern	Discard Digits	Comments
9.XXXX	Pre Dot Trailing	Discard a digit before '.' and accept any 4 digits. 91001 is represented as 1001.
8XXX	None	A four-digit pattern starting with 8.
XXXX	None	Accept any four digits.

#### Unlphi Administrator Configuration

For interswitch routing to work with a Cisco CallManager, the following configuration must be done in the Uniphi System Administrator:

1. Go to System→ Adapters, find a TAPI Voice Cisco Adapter, and add all CTI Route Points created for interswitch routing (step 5 of the Cisco CallManager Configuration) in the 'Route Point' field. In the following example:

Route Point = 1234, [1000\_1010]

[1000\_1010] is a range of CTI Route Points for the interswitch routed calls.

1234 is a CTI Route Point for a regular call (not for the interswitch routing calls).

2. Go to System  $\rightarrow$  Correlation Key, and create a new correlation key set (either rolling or non-rolling). Enter the start and end numbers to define a range of keys within the set.

For example, correlation keys for calls flowing from the Cisco CallManager to the Aspect ACD could be defined as:

CM2ACD '1', '1', '1', 2000, 2010

where 2000-2010 is the correlation key range representing the Aspect ACD DNIS numbers.

Correlation keys for calls flowing from the Aspect ACD to the Cisco CallManager Correlation Keys could be defined as:

ACD2CM '1', '1', '1', 1000, 1010

where 1000-1010 is the correlation key range representing Cisco CallManager CTI Route Points (step 5 of the Cisco Call Manager Configuration).

See Correlation Keys on page 90 for more details on using correlation keys in interswitch routing.

#### Gateway Configuration

Cisco CallManager supports multiple gateways, each with its own configuration settings. The following sample configuration code is for the Cisco Gateway AS5000 series.

```
dial-peer voice 1 voip
destination-pattern .T
voice-class codec 10
session target ipv4:169.136.79.250
dtmf-relay h245-alphanumeric
ip precedence 5
no vad
dial-peer voice 2 pots
preference 1
destination-pattern .T
port 0:0
forward-digits all
```

This sample code presents two gateway dial peers that reflect two different call types:

- The first dial peer (voip) is used to route a call from the Gateway to the Cisco CallManager. (169.136.79.250). Destination pattern '.T' says to forward all digits to Cisco CallManager.
- The second dial peer (voice) is used to connect a call from the Gateway to the PSTN switch on port 0:0 (one of the physical tie lines). Destination pattern '.T' says to forward all digits to physical line.

## Index

## A

adding steps, 22 Advanced Routing, 74 configuration in Administrator, 74 agent groups, 80 agent pools, 74 application, business, 14 Architect opening, 11 step palette, 12 watch area, 13 window, 11 arranging steps, 29 Aspect eFlow System, 7, 8 configuration, 14, 17 Auto Arrange, 30 average runtime, 42

## B

business application, 14

## С

calculation of statistics, 42 caption, step, 23 comments, adding to an eFlow, 27 configuration, 14, 17 connecting steps, 23 creating eFlows, 17 variables, 25

### D

data source, 14 debugging an eFlow, 13, 36 deleting an eFlow, 39

### Ε

eBusiness Architect, see Architect eFlow statstics, 40 eFlow Engine, 7 eFlow System, 7, 8 configuration, 14, 17 1

eFlows comments in, 27 creating, 17 debugging, 13, 36 definition of, 9 deleting, 39 input/output labels, 27 managing, 15, 39 opening, 20 properties, 39 push/pull, 69 regenerating, 30 rules for executing, 38 runtime statistics, 15, 39 sample, 10, 69 Pull eFlow, 72 Push eFlow, 70 Select Agent Group, 80 Selecting an Agent by Extension, 77 Simple RMS, 75 Skill-Based Routing, 81 Task Escalation, 72 Two-Tiered Agent Search, 76 saving, 31 variables, 25 Error Count, 42 error count, 42 error messages, 13 event, 21 RouteRequest, 21 rules, 15, 38 Exception Count, 42 exception count, 42 execution count, 42

# Η

Help, online, 14

## 

input labels, 27 interswitch routing, 87 to 102

### L

labels, 27 LDAP server, 7

### Μ

maximum runtime, 42 messages, error, 13

## 0

online Help, 14 opening an eFlow, 20 Architect, 11 output labels, 27

### Ρ

palette, step, 12 parameters, routing, 74 pools, agent, 74 properties eFlow, 39 step, 24 push/pull eFlows, 69

## Q

queues, RMS, 74

## R

regenerating an eFlow, 30 repository, 14 Resource Matching Service (RMS), 74 sample eFlow, 75 RouteRequest event, 21 routing parameters, 74 rules, 75 rules, 38 event, 15 routing, 75 runtime, 42 runtime statistics, 15, 39

## S

saving an eFlow, 31 locally, 32 on the server, 31 Select Agent step, 75 selecting steps, 28 skill-based routing, 81 starting a new eFlow, 18 statistics displaying, 40 statistics, runtime, 15, 39 steps adding, 22 arranging, 29 captions of, 23 connecting, 23 palette, 12 properties, 24 reference, 47 selecting, 28 width of, 23 subflows, 15, 35

## Т

task escalation, 72 total runtime, 42 two-tiered agent search, 76

## U

Uniphi Administrator, 14, 17

# V

variables, 13, 25 viewing runtime statistics, 15, 39

## W

watch area, 13 workspace, 11

Aspect Uniphi Suite v6.0: eFlow Developer Guide