Oracle® Telephony Manager

Concepts and Procedures

Release 11*i*

November 2000

Part No. A86106-02

ORACLE®

Oracle Telephony Manager Concepts and Procedures, Release 11*i*

Part No. A86106-02

Contents

Implementing Oracle Telephony Manager

Administering Oracle Telephony Manager

Send Us Your Comments

Oracle Telephony Manager Concepts and Procedures, Release 11*i*

Part No. A86106-02

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this document. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most?

If you find any errors or have any other suggestions for improvement, please indicate the document title and part number, and the chapter, section, and page number (if available). You can send comments to us via the postal service.

Oracle Corporation
CRM Content Development Manager
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

If you would like a reply, please give your name, address, telephone number, and (optionally) electronic mail address.

If you have problems with the software, please contact your local Oracle Support Services.

Preface

Welcome to the Oracle Customer Relationship Management, Release 11*i*, suite of applications.

This Concepts and Procedures provides information and instructions to help you work effectively with Oracle Telephony Manager.

This preface explains how Concepts and Procedures is organized and introduces other sources of information that can help you.

Intended Audience

This guide is aimed at the following users:

- Technical Service Representatives (TSR)
- Customer Service Representatives (CSR)
- System Administrators (SA), Database Administrators (DBA), and others with similar responsibility
- Call Center Managers and Supervisors

This guide assumes you have the following prerequisites:

1. Understanding of computer-telephony integration (CTI)
2. Understanding of call center technology
3. Understanding of the company business processes
4. Understanding of Oracle databases

Structure

This manual contains the following sections:

“Understanding Oracle Telephony Manager" provides overviews of the application and its components, explanations of key concepts, features, and functions, as well as the application’s relationships to other Oracle or third-party applications.

“Using the Soft Phone” provides process-oriented, task-based procedures for monitoring and controlling a teleset from the desktop.

“Implementing Oracle Telephony Manager” provides general descriptions of the setup and configuration tasks required to implement the application successfully.

“Administering Oracle Telephony Manager" provides task-based procedures for required for ongoing system maintenance and includes information on administration tools and utilities.

Related Documents

For more information, see the following manuals:

- *Installing Oracle Call Center Connectors*
- *Oracle Call Center Applications Setup*
- *Oracle Telephony Manager Implementation Guide*
- *Oracle Telephony Manager Technical Reference Manual*

Understanding Oracle Telephony Manager

This topic group provides overviews of Oracle Telephony Manager and its components, explanations of key concepts, features, and functions, as well as the application's relationships to other Oracle or third-party applications.

Oracle Telephony Manager is the platform for all Oracle call center and telephony applications. Oracle Telephony Manager translates the proprietary ACD / PBX protocol of telephony communications into computer-compatible language, and routes those calls to the designated telephones and computers used by call center agents.

Each call center requires its own Oracle Telephony Manager, which can service up to 500 agents per call center.

What's New in Oracle Telephony Manager

In Release 11*i*, major changes from previous versions of Oracle Telephony Manager include the following:

- Oracle Telephony Manager now consists of a number of server applications, each of which handles a particular function. For example, the Inbound Telephony Server application monitors inbound call route points, the Routing Server application handles routing logic, and the Oracle Telephony Manager application handles media queuing and media control.
- You now add agents and agent groups by using the Foundation Resource Manager instead of using Oracle Telephony Manager. For more information, see the Foundation Resource Manager instructions.
- Agent skills are now configured and assigned by your company's human resources department instead of by call center administrators.

Monitor Server Status with the Server Monitor

The Server Monitor is a server application that monitors in real time other call center and the telephony server applications Oracle Telephony Manager, Oracle Telephony Manager Controller, eMail Center, Inbound Telephony Server, and Routing. The Server Monitor has the following functions:

- Real-time status monitoring that detects whether servers are up or down
- Remote server launching that can start or stop any server
- Fault tolerance that enables monitored servers to take appropriate action when they receive notice that another monitored server has gone down, and reconnect transparently when the monitored server comes back up

Using the server monitor command line, you can start any monitored server remotely.

Route Calls According to Defined Rules with Active Mode

In active mode, Oracle Telephony Manager routes calls according to defined rules. Active mode is the default mode of operation. In active mode, Oracle Telephony Manager uses *skill-based routing*, a dynamic call routing intelligence that delivers inbound calls to an agent who is appropriately skilled to meet the needs of the caller.

When inbound calls arrive at the switch, the switch issues a routing request. Oracle Telephony Manager monitors the routing request, applies skill-based routing rules, and identifies suitably skilled agents.

Meanwhile, a representation of the call waits in the virtual queue within Oracle Telephony Manager. When a suitable agent becomes available, Oracle Telephony Manager responds to the routing request and instructs the switch to deliver the call to the agent's extension via Universal Work Queue, where more specific pre-defined rules may apply in directing the call.

Bypass Routing Rules with Passive Mode

If necessary, you can configure Oracle Telephony Manager to operate in passive mode to bypass the routing rules.

Passive mode is typically used when a call center prefers traditional ACD routing features and/or the statistics and reporting tools of a switch vendor. In passive mode, the ACD/PBX system completely handles the routing and queuing of

inbound calls. Oracle Telephony Manager does not monitor or respond to routing requests from the switch, and skill-based routing is essentially inactive.

As calls arrive at the agents' extensions, Oracle Telephony Manager monitors the calls and captures any associated data (for example, ANI, DNIS) for screen pops.

For Nortel Meridian only, when Oracle Telephony Manager is in passive mode, the ACD/PBX system routes inbound calls and delivers them to the ACD DN, also known as the *Automatic Call Distribution key*, on the teleset. When Oracle Telephony Manager is in active mode, skill-based routing directs inbound calls to the directory number (DN), also known as the *Single Call Ringing key*, on a teleset.

Support Large Call Centers with Distributed Configuration

Distributed configuration can support call centers with more than 1000 agents, and has better scalability than standalone configuration. A distributed Oracle Telephony Manager system consists of one media queue server application, that is, Oracle Telephony Manager, and one or more media controller server applications, that is, Oracle Telephony Media Controller. Currently, Oracle supports one Oracle Telephony Media Controller per Oracle Telephony Manager.

Support Small Call Centers with Standalone Configuration

Standalone Oracle Telephony Manager is a single server application that handles both media queuing and media control. Standalone configuration is suitable for call centers with fewer than 300 agents, and is easier to maintain than distributed configurations.

Control Media Functionality with Oracle Telephony Media Controller

Oracle Telephony Media Controller is software that bridges other systems or software with the underlying media hardware, such as PBX. For example, the ability to monitor and control a teleset from software, such as the soft phone, is telephony media control.

Handle Inbound Items with Media Queue

Media Queue is the queuing and distribution component for inbound media items. Media Queue stores inbound items such as inbound calls or inbound email in a queue, and integrates with the routing screen to send the items to a set of agents for

handling. Media Queue provides an API to other screens, such as Universal Work Queue, for querying and manipulating items in the queue.

Using the Soft Phone

This topic group provides process-oriented, task-based procedures for using the soft phone to perform essential business tasks.

The soft phone is a telephone user interface on the call center agent's monitor screen. Agents use the soft phone just as they would use the buttons of a physical call center telephone.

For instructions on soft phone functions, see the following topics:

Using the Soft Phone Line Buttons

The soft phone line buttons light up to indicate line status according to the following color code.

Line Button Color Code

Line Button Color	Definition
Gray	Selected line

Green	Currently active line
Yellow	Ringing line

Placing a Call

To place a call with the soft phone, enter the number using the key pad or type into the entry box at the top, and then click **Dial**.

Dialing Digits (DTMF Tones)

To dial DTMF tones during a call with the soft phone, enter the digits using the key pad or type into the entry box at the top, and then click **Dial**.

Forwarding Incoming Calls

To forward all incoming calls to another party with the soft phone, enter the party's extension and then click **Forward**. The **Forward** button should be green at this point.

To stop forwarding calls with the soft phone, click **Forward** again. The **Forward** button should return to its default color.

Deflecting Incoming Calls

To deflect a ringing call to another party with the soft phone, enter the party's extension and then click **Dial**. This feature is supported for selected middlewares/switches only.

Answering a Call

To answer a call with the soft phone, click on the (blinking yellow) line button with the ringing call.

Putting a Call on Hold

To place a call on hold with the soft phone, click **Hold**. The line button with the held call should be blinking green.

To retrieve a held call with the soft phone, click **Hold** again or click the line button with the held call.

Transferring a Call

Steps

1. To transfer a call with the soft phone on a *non-Aspect* system, use the following steps:
   a. With the caller on the line, dial the party to whom you wish to transfer the call.
   b. Click **Transfer**.
   c. When the consulted party answers and accepts the call, click **Transfer** again.
2. To transfer a call with the soft phone on an *Aspect* system, use the following steps:
   a. Place the caller on hold by clicking **Hold**.
   b. Select another line.
   c. Dial the party to whom you wish to transfer the call.
   d. When the consulted party answers and accepts the call, click **Transfer**.

Conferencing a Call

Steps

1. To conference a call with the soft phone on a non-Aspect system, use the following steps.
   a. With the caller on the line, dial the party to whom you wish to add to the conference call.
   b. Click **Conference**.
   c. When the consulted party answers and accepts the call, click **Conference** again.
2. To conference a call with the soft phone on an Aspect system, use the following steps.
   a. Place the caller on hold by clicking **Hold**.

b. Select another line.

c. Dial the party whom you wish to the add to conference call.

d. When the consulted party answers and accepts the call, click **Conference**.

e. To disconnect from the conference call, click **Release**.

Hanging Up a Call

To hang up a call, click **Release**.

Indicating That You Are Not Ready to Receive Calls

When you have completed a call and need time to finish work associated with that call, choose the Wrap Up button to make your soft phone temporarily unavailable to receive incoming calls.

When you are logged into *Universal Work Queue* and the soft phone is in WRAPUP state, you can continue to work with your Oracle application without receiving calls.

Implementing Oracle Telephony Manager

For information about setup and configuration tasks required to implement Oracle Telephony Manager, refer to the *Implementing Oracle Telephony Manager* guide.

Administering Oracle Telephony Manager

This topic group provides task-based procedures for required for ongoing system maintenance and includes information on administration tools and utilities.

To configure major functions, see the following topics:

- Configuring Call Routing
- Configuring Static Groups
- Configuring Dynamic Groups
- Configuring Classifications
- Defining Route Priorities

Configuring Call Routing

Configuring call routing in Oracle Telephony Manager requires performing several configuration procedures sequentially. It also require understanding of the rule-based routing concepts and the business requirements. The number and kind of procedures depends upon whether routing to groups is static or dynamic. *Static routing* is based on data stored in cache. *Dynamic routing* is based on PL / SQL queries. Static routing is faster, but dynamic routing is more flexible.

To configure call routing, open the Routing Administration screen.

For instructions on the first routing configuration procedure, see Configuring Static Groups.

Configuring Static Groups

The first call routing configuration procedure is configuring static groups. A *static group* requires that you manually edit information about individual group members.

To configure static groups, open the Routing Server Administration screen and choose **Static Groups**.

The Define Groups window opens.

When configuring static groups, use the following guidelines:

- The Group Number and Group Name must be unique.
- If you want to prevent an agent from being shared with other groups, check the Exclusive Flag box to indicate that the agent belongs exclusively to this group.
- You cannot create member records without choosing a Parent Group.

To proceed to the next call routing configuration procedure, see Configuring Dynamic Groups.

Configuring Dynamic Groups

In a *dynamic group*, the database automatically updates information about individual group members.

To configure dynamic groups, in the Routing Server Administration screen, click **Dynamic Groups**.

The Dynamic Groups window opens.

When configuring dynamic groups, use the following guidelines:

- The group Number and group Name must be unique.
- A description is optional.
- In **SQL Statement**, enter the SQL statement that generates the routing rule.
- To generate the routing rule, choose **Check Syntax**. An error message notifies you if the SQL statement is invalid. If the SQL statement is valid, no message appears.

To proceed to the next call routing configuration procedure, see Configuring the Route Definition.

Configuring the Route Definition

When configuring the route definition, use the following guidelines:

- The Route Definition Name is an arbitrary, descriptive name of the route.
- The Route Type can be static or dynamic.
  - If the Route Type is dynamic, then a drop-down list of Default Destinations becomes available, and the Procedure Parameters table opens in the Route Destinations tab.

- If the Route Type is static, the Destinations table opens in the Route Destination tab.

- If you select Non-Application Database, you need to enter:
  - Database Driver
  - Database URL

To proceed to the next call routing configuration procedure, see Configuring Route Destinations.

Configuring Route Destinations

A route destination is the agent group to which a call is routed.

When configuring route destinations, use the following guidelines:

- The appearance of and available fields in the Route Destinations tab depends upon whether the chosen Route Type option is Static or Dynamic.
  - If the Route Type is Static, in the Route Destinations tab the Destinations table is visible.
  - If the Route Type is Dynamic, in the Route Destinations tab the Dynamic Destinations table and the Procedure Parameters table are visible.
- In the Dynamic Destination Procedure Name field, enter the procedure (Package.Procedure) that is to be used to derive the destination.
- If the route server cannot determine agents from the defined destinations, then the server defaults to the default destination defined for the route.

To proceed to the next call routing configuration procedure:

- If the Route Type is *static*, proceed to Configuring Route Rules.

  OR

- If the Route Type is *dynamic*, proceed to Configuring Procedure Parameters.

Configuring Procedure Parameters

The Procedure Parameters table is visible *only* if the Route Definition table Route Type is Dynamic. (If the Route Type is static, skip this procedure and proceed to Configuring Route Rules.)

Include values such as in the following examples:

- Parameter: CUSTOMER_ID:RESOURCE_ID. Separate the parameters with a colon.
- Value: 312
- Datatype: VARCHAR or INTEGER
- Direction: IN or OUT
- Sequence is the sequence of the parameter, a generated number.

To proceed to the next call routing configuration procedure, see Defining Route Rules.

Defining Route Rules

Route rules define the parameters that can be expected in an incoming call.

Choose the Key value from the drop-down list.

Enter the Value to correspond with the Key. For example, if the Key is ANI, the Value could be a telephone number. If the Key is DNIS or CUSTID, the Value could be any value.

To proceed to the next call routing configuration procedure, see Configuring Classifications.

Configuring Classifications

Configure classifications by opening the Classification Administration screen.

Classifications specify how to identify and route particular types of calls that are associated with the selected Route ID.

Classifications are of two types: *literal* or *database procedure*. A literal classification is a string. A database procedure classification is a stored procedure. The following conditions apply.

- If the classification is literal, you can define a classification.
- If the classification is a database procedure, give it a procedure name such as GET_CLASSIFICATION.

Note: Because there is no way to verify the stored procedure that you use, you need to be certain that you enter and call the correct procedure string.

To proceed to the next call routing configuration procedure, see Configuring Classifications Rules.

Configuring Classification Rules

Classification rules can determine the route that a call must take, or a classification rule can designate a call as a classification for reports or for screen pops.

Every classification must have rules that define the conditions under which the classification occurs. When a call meets all these conditions, the classification is the result and the call is routed accordingly.

In an example scenario, if a request to route ANI of 800 is a classification for Gold Card customer, then the resulting route destination could be the priority group.

When configuring classification rules, use the following guidelines:

- The Key that you choose from the drop-down list must be the same Key value that is entered in the Route Rules tab.
- A classification can have multiple rules associated with it. An individual route can have multiple destinations, which can be either static or dynamic.
- Every route has a sequence of a Key and a Value whose relationship is determined by one of the available Operations (=, !=, >, >=, <, <=). The following table lists examples of classification rules.

Sample Classification Rules

Key	Operation	Value
CUSTID	>=	100000
DNIS	=	8005555555
ANI	=	5551234

- If an incoming routing request results in a new classification, the new classification is added to the route request as one of the parameters. The server tries to find a new route based on this route request.

Note: If you configure a classification rule that results in a new rule, you need to have a route that is defined for that new rule.

For example, if the classification rules in the preceding table result in adding the new classification rule CLASSIFICATION=GOLD to the classification request, then a new route is determined based on the new rule.

To proceed to the next call routing configuration procedure, see Configuring Classification Parameters.

Configuring Classification Parameters

You only need to configure classification parameters if the classification type is a database procedure. In the Classification Administration window, the Classification Parameters tab is visible *only* if the selected Classification Type is Database Procedure. Configuring classification parameters is the last procedure in configuring call routing.

The configurations for Classification Parameters are the same as for Procedure Parameters in the Route Destinations tab.

Defining Route Priorities

You can define and change the priority of route definitions by using the Routing Priority Administration screen.

Changing the value of a route priority affects the values of other route priorities. If you increase the value of a priority, then the priorities of all the routes with a value equal to or greater than the original value decrease by one. If you decrease the value of a priority, then the value of all the routes with a value equal to or less than the original value increase by one. For example, if you decrease 6 to 3, then 3 increases to 4, 4 increases to 5, and so on. If you increase 3 to 6, then 4 decreases to 3, 3 decreases to 2, and so on.

Updating Group Member and Administrator Records

Define and edit group member and group administrator records in the Define Resources Groups window.

To administer group member or group administrator records, use the following procedure:

Prerequisites

The group must already have been created.

Steps

1. Open the Routing Administration screen and click **Manage Static Groups**.

   The Define Resources Groups window opens.

2. Choose either the **Members** or **Admins** tab.
3. From the pull-down lists, choose the **Category** and **Number**.
4. In the **Start Date** and **End Date** fields, enter the dates beginning and ending dates of the member or agent's assignment to this group.
5. From the menu, choose **Save**.

Moving Agents to Other Groups

To move a member of an agent group to a different group, use the following procedure:

Prerequisites

The groups must already have been created.

Steps

Note: When you move agents from one group to another group, the agents are automatically deleted from the previous group.

1. Open the Routing Administration screen and click **Manage Static Groups**.

   The Define Resources Groups window opens.

2. In the Members tab, from the Category and Name pull-down lists, choose the group member.
3. Choose **Move Member**.

Administering Call Routing with Oracle Workflow

With Oracle Workflow Builder 2.5, you can use drag-and-drop routing function templates to implement and administer the Call Routing screen.

To create and configure a route path using the Oracle Workflow Builder-based Call Routing screen, you need to have a thorough understanding of Oracle Workflow Builder.

Accessing the Call Route Item Type

To open the Oracle Workflow Builder Call Route item type, use the following procedure.

Prerequisites

None.

Steps

1. Launch Oracle Workflow Builder.
2. Open the database connection.
3. From the Show Item Types window, choose **Call Route** > **Show**.

Understanding Workflow Builder Item Types

All Oracle Workflow Builder processes are associated with an item type. An item type can include one or more processes. When you save your work in Oracle Workflow Builder, you actually save everything associated with the item type that is currently selected. You can save an item type to a database or a flat file, and similarly load an item type into Oracle Workflow Builder from a database or a flat file. Opening an item type automatically retrieves all the attributes, messages, lookups, notifications, functions, and processes associated with that item type.

Applying a Pre-Defined Route

Oracle Workflow Builder includes routing process examples that you can apply as your call center routing process or you can use as templates to build your own routing process.

To access a pre-defined routing processes, use the following procedure.

Prerequisite

None.

Steps

1. From the navigator tree, click **Call Route Item type**.
2. Expand the Processes folder to see a list of available pre-defined processes. Oracle Telephony Manager includes pre-defined example routes.
3. Double-click on the process name to view.

Running a Pre-Defined Routing Process

To run a pre-defined routing process with Oracle Workflow Builder, use the following procedure.

Prerequisites

None.

Steps

1. From the Attributes folder, choose The Selected Process.

   Warning: Do not modify any attribute except 'The Selected Process' attribute. Any other modification may render the Call Routing screen unusable.

   The Oracle Workflow Navigator Control Properties window appears.
2. In the Default Value field, enter the name of the process that you want to run.
3. Choose **OK**.

Modifying a Process

The Process window in Oracle Workflow Builder graphically represents the activities (icons) and transitions (arrows) for a particular process. Notification, function, and process activities make up the nodes of a process.

If a process contains a process activity in its diagram, then that process activity is known as a *subprocess*. To display information on the subprocess in a separate

Process window, double-click on the subprocess activity node in the parent Process window.

Each activity is a node, a logical step that contributes toward the completion of a process. Nodes contain stored procedures that operate on parameters. The GetAgents node routes information to a database table, invokes a stored procedure and returns the results. The database may return up to ten parameters, each parameter being an agent.

WARNING: Before saving a process to the database, select Verify from the File menu to validate the process. Do not save un-validated processes to the database. The entire Call Route item type will be rendered an error and all call routing functionality will be disabled.

You can drag and drop activities from the navigator tree into the Process window. Define transitions between activities by drawing arrows from one activity to the next using the secondary mouse button.

Designating Start and End Activities

Each Oracle Workflow Builder process must have a Start activity that identifies the beginning point of the process. You may designate any node from which it is logical to begin the process as a Start activity. An End activity should return a result that represents the result of the completion of the process.

Prerequisite

None.

Steps

1. Right click the icon that you wish to change.
2. Choose **Properties**.
3. Using the Start/End pull-down list, change the step in the process.
4. Indicate if the current node is a start or end activity in your process by choosing START or END, respectively. The default is NORMAL, which means neither. You may have multiple START and END nodes in your process.

A Start activity is marked (Start) and has a small green arrow in its activity icon. An End activity is marked (End) and has a red arrow in its activity icon.

5. For an END node, if the overall process activity has a result type associated with it, you must also select a value for the final process result. The list of values for the final process result derive from the lookup type defined as the process activity's result type.

Adding a New Node to a Process

From the Oracle Workflow Builder Process window, create a new node by dragging and dropping a function or process activity from the navigator tree into the Process window. Use functions to fine-tune the route. The activity you drag must belong to the same data store as the process to which you drag it.

If you use the right mouse button menu New Activity in the Process window to create a new node, the property pages for the node appear. Select the name of the item type and activity in the Activities property page. If you create a node by dragging and dropping an activity from the navigator tree into the process window, then double-click on the node to display the property pages so you can further specify the details of the node.

If the node is a function activity and the function activity has activity attributes, you can assign values to those activity attributes by choosing the Attribute Values tab to display the Attribute Values property page.

If the node is a process activity, then a small subprocess overlay icon appears over the upper right corner of process activity icon. The subprocess overlay icon identifies the node as a subprocess within the process diagram.

Creating a New Process Route

To create a new process route with Oracle Workflow Builder, use the following procedure.

Prerequisite

None.

Steps

1. Select the Processes folder and right mouse click to create a new process.

2. When the Oracle Workflow Navigator Control Properties dialog box appears, select the Activities tab and:
   a. Enter a unique **Internal Name**.
   b. Enter the **Display Name**.
   c. Enter a **Description** of your new route.
   d. Leave the Result Type as None.
3. From Standard Functions, drag the Start activity to the new Process window.
4. From Call Route Functions, drag the Get Agents activity to the window.
5. From Call Routing Filters, CCT Environmental Conditions, Customer Initialization Phase, Customer Product Initialization, Product Initialization Phase, and Standard Item Types, drag and drop the activities you want to use in routing the calls to the appropriate agent.

Administering Call Routing Filters

Use Oracle Workflow Builder Call Routing Filters to route calls only to those agents who meet the filter criteria.

Call Routing Filters

Call Routing Filter	Description
Competency - Language Filter	Ensures that only those agents who are competent in the language selected by the customer answer the call. Requires that a language selection be made at the IVRU.
Competency - Knowledge Filter	Ensures that only those agents who are competent in the area of expertise selected by the customer answer the call. Requires that an area of expertise selection be made at the IVRU.
Competency - Product Filter	Used to ensure that only those agents who are competent in the product selected by the customer answer the call. Requires that a product selection be made at the IVRU.
Customer Product Filter	This filter is used to ensure that only those agents who are assigned to handle the particular 'Customer Product' combination answer the call. Requires that the 'Derive Customer Product ID' sub-process be completed before this activity in the routing workflow process.

Call Routing Filters (Cont.)

Call Routing Filter	Description
Customer ID Filter	This filter is used to ensure that only those agents who are assigned to handle this particular customer answer the call. Requires that the 'Derive Customer ID' sub-process be completed before this activity in the routing workflow process.
Problem Code Filter	This filter is used to ensure that only those agents who are assigned to handle the particular 'Problem Code' answer the call. Requires that a problem code selection be made at the IVRU.
Product ID Filter	This filter is used to ensure that only those agents who are assigned to handle the particular product answer the call. Requires that the 'Derive Product ID' sub-process be completed before this activity in the routing workflow process.
Request Owner Filter	This filter is used to ensure that the call is routed to the agent who owns the 'Service Request', when the customer is calling about the particular 'Service Request'. Requires that a service request number selection be made at the IVRU.
System Type Filter	This filter is used to ensure that only those agents who are assigned to handle the particular 'System Type' answer the call. Requires that a serial number be input from the IVRU.
Telesales Agent Account Filter	This filter is used to retrieve the agents for a customer based on the account ID.
Telesales Agent Opportunity Filter	This filter is used to retrieve the agents for a given sales opportunity.
Telesales Agent Territory Filter	This filter is used to retrieve the agents for a customer based on their territory.
Agents from Dynamic Group Name	This filter is used to retrieve the agents returned based on the dynamic group name.
Agents from Dynamic Group Number	This filter is used to retrieve the agents based on the dynamic group number.
Agents from Static Group Name	This filter returns agents based on a static group name.
Agents from Static Group Number	This filter returns agents based on a static group number.
Agents not in Dynamic Group Name	This filter returns agents not in a dynamic group name.

Call Routing Filters (Cont.)

Call Routing Filter	Description
Agents not in Dynamic Group Number	This filter returns agents not in a dynamic group number.
Agents not in Static Group Name	This filter returns agents not in a static group name.
Agents not in Static Group Number	This filter returns agents not in a static group number.
All Logged in Agents	This filter returns a list of all logged in agents.

Administering Call Center Environmental Conditions

Call Center Environmental Conditions refers to the routing rules based on the date and time of the call.

Setting Time and Date Rules

In Oracle Workflow Builder, for each of these attributes, the relevant value is specified after you drag the node onto the process diagram.

To set time and date rules, use the following procedure.

Prerequisites

None.

Steps

1. In the process diagram, double click on the node icon to open the node properties sheet.

   The property page for the node appears.

2. Choose the Attribute Values tab.

   The Attribute Values property page for the node appears.

3. Choose the applicable attribute or attributes and enter them in the **Value** field.

4. Choose **Apply** to save your changes.

   OR

   Choose **OK** to save your changes and close the Property page.

Time and Date Rules

Rule	Description
Between Two Dates?	If the current date falls between the start and end dates specified, this rule returns a Yes, otherwise it returns a No.
During Business Hours?	If the call is received during business hours, this rule returns a Yes, otherwise it returns a No.
Get Day of the Month?	This rule returns the current day of the month as a number. You can use this rule to handle special, but not contiguous, days of the month differently.
Get Day of the Week?	This rule returns the current day of the week. You can use this rule to handle weekends or certain days of the week differently.
Is It After Specified Date?	If the current date is after the specified date, this rule returns a Yes, otherwise it returns a No.
Is It Before Specified Date?	If the current date is before the specified date, this rule returns a Yes, otherwise it returns a No.
Is It After Specified Time?	If the current time of day is after the specified time of day, this rule returns a Yes, otherwise it returns a No.
Is It Before Specified Time?	If the current time of day is before the specified time of day, this rule returns a Yes, otherwise it returns a No.

Deriving Rules for the Customer Initialization Phase

The Oracle Workflow Builder Customer initialization phase derives Customer ID information from the caller's responses to the IVR. The phase is represented by the Derive Customer ID sub-process, which you drag onto the process diagram.

You can drag the following rules for derivation of Customer ID onto the Derive Customer ID sub-process. You can also use these rules directly in the Routing process diagram.

Customer Initialization Phase Rules

Rule	Description
Customer ID Exists?	If the customer ID is already known, this rule returns a Yes, otherwise it returns a No. Using this as the first rule in the 'Derive Customer ID' sub-process, will allow for a quick exit from the sub-process without applying any derivation rules if the Customer ID is already known.

Customer Initialization Phase Rules (Cont.)

Rule	Description
Can Get Customer ID from Invoice Num?	If the customer ID can be derived from the Invoice Num, this rule derives the Customer ID and returns a Yes, otherwise it returns a No.
Can Get Customer ID from Order Num?	If the customer ID can be derived from the Order Num, this rule derives the Customer ID and returns a Yes, otherwise it returns a No.
Can Get Customer ID from PO Num?	If the customer ID can be derived from the PO Num, this rule derives the Customer ID and returns a Yes, otherwise it returns a No.
Can Get Customer ID from Request Num?	If the customer ID can be derived from the Request Num, this rule derives the Customer ID and returns a Yes, otherwise it returns a No.
Can Get Customer ID from RMA Num?	If the customer ID can be derived from the RMA Num, this rule derives the Customer ID and returns a Yes, otherwise it returns a No.
Can Get Customer ID from Serial Num?	If the customer ID can be derived from the Serial Num, this rule derives the Customer ID and returns a Yes, otherwise it returns a No.
Can Get Customer ID from System Name?	If the customer ID can be derived from the System Name, this rule derives the Customer ID and returns a Yes, otherwise it returns a No.

Deriving Rules for the Customer Product Initialization Phase

The Oracle Workflow Builder Customer Product initialization phase derives the Customer Product ID information from caller's responses to the IVRU. The phase is represented by the Derive Customer Product ID sub-process, which you drag onto the process diagram.

You can drag the following rules for derivation of Customer Product ID onto the Derive Customer Product ID sub-process. You can also use these rules directly in the Routing process diagram.

Customer Product Initialization Phase Rules

Rule	Description
Customer Product ID Exists?	If the customer product ID is already known, this rule returns a Yes, otherwise it returns a No. Using this as the first rule in the Derive Customer Product ID sub-process, will allow for a quick exit from the sub-process without applying any derivation rules if the Customer Product ID is already known.
Can Get Customer Product ID from Reference Num?	If the customer product ID can be derived from the Reference Number, this rule derives the Customer Product ID and returns a Yes, otherwise it returns a No.
Can Get Customer Product ID from Request Num?	If the customer product ID can be derived from the Request Number, this rule derives the Customer Product ID and returns a Yes, otherwise it returns a No.
Can Get Customer Product ID from Serial Num?	If the customer product ID can be derived from the Serial Number, this rule derives the Customer Product ID and returns a Yes, otherwise it returns a No.

Deriving Rules for the Product Initialization Phase

The Oracle Workflow Builder Product initialization phase derives the Product ID information from the caller's responses to the IVR. The phase is represented by the Derive Product ID sub-process, which you drag onto the process diagram.

You can drag the following rules for derivation of Product ID onto the Derive Product ID sub-process. You can also use these rules directly in the Routing process diagram.

Product Initialization Phase Rules

Rule	Description
Product ID Exists?	If the product ID is already known, this rule returns a Yes, otherwise it returns a No. Using this as the first rule in the 'Derive Product ID' sub-process, will allow for a quick exit from the sub-process without applying any derivation rules if the Product ID is already known.
Can Get Product ID from Reference Num?	If the product ID can be derived from the Reference Number, this rule derives the Product ID and returns a Yes, otherwise it returns a No.
Can Get Product ID from Request Num?	If the product ID can be derived from the Request Number, this rule derives the Product ID and returns a Yes, otherwise it returns a No.

Product Initialization Phase Rules (Cont.)

Rule	Description
Can Get Product ID from Serial Num?	If the product ID can be derived from the Serial Number, this rule derives the Product ID and returns a Yes, otherwise it returns a No.

Deriving Rules for TeleSales Routing

The Oracle Workflow Builder telesales routing functions derives the Customer ID information from the caller's responses to the IVR. The phase is represented by the Derive Telesales Customer ID sub-process, which you drag onto the process diagram.

You can drag the following rules for derivation of Customer ID onto the Derive Telesales Customer ID sub-process. You can also use these rules directly in the Routing process diagram.

Telesales Routing Rules

Rule	Description
Customer ID from Address ID?	If the customer ID can be derived from the Address ID, this rule derives the Customer ID and returns a Yes, otherwise it returns a No.
Customer ID from ANI?	If the customer ID can be derived from the ANI, this rule derives the Customer ID and returns a Yes, otherwise it returns a No.
Customer ID Exists?	If the Customer ID is already known, this rule returns a Yes, otherwise it returns a No. Using this as the first rule in the 'Derive Telesales Customer ID' sub-process, will allow for a quick exit from the sub-process without applying any derivation rules if the Product ID is already known.

Deriving Rules for Banking Routing

The Oracle Workflow Builder banking routing functions derives the group information from the caller's responses to the IVR.

You can also use these rule functions directly in the Routing process diagram.

Banking Routing Rules

Rule	Description
Group from Bank Branch	If the group can be derived from the Bank Branch, this rule derives the Agent Ids based on the group.
Group from Bank ID	If the group can be derived from the Bank Id, this rule derives the Agent Ids based on the group.
Group from Profitability	If the group can be derived from profitability, this rule derives the Agent Ids based on the group.

Starting Servers

When you need monitoring, Oracle suggests that you start the Server Monitor first. When the Server Monitor is in use, you can start the Inbound Telephony Server before starting Oracle Telephony Manager.

When the Server Monitor is *not* in use, Oracle Telephony Manager must be running before you can start the Inbound Telephony Server. If you start the server monitor at any time after startup, the monitored servers will automatically connect to it.

Monitored servers detect when the server monitor starts and by default connect to the server monitor within five minutes. To change this connect default value specify the `sm_reconnect_interval` parameter in the SERVER.INI file and the SMI.INI file.

Starting the Server Monitor

On Windows NT, Oracle recommends that you start the Server Monitor as the service `OracleServerMonitor_ServerName`. To start the server monitor, from the home directory run the batch file SM.BAT.

On UNIX, to start the server monitor, run the script file SM.SH.

Restarting the Server Monitor

If the Server Monitor goes down and you restart it, all clients of the Server Monitor automatically re-register with the Server Monitor and other servers. The server launcher also automatically reconnects when the Server Monitor restarts.

Starting the Server Launcher

The server launcher performs remote startup of the server monitor. The server launcher is installed automatically in the installation of any call center server. The server launcher must run on every machine that a) runs any call center and telephony server and b) requires remote startup. When the server monitor notifies the server launcher, the server launcher starts an NT service on Windows NT and starts a background process on UNIX.

Starting the Server Monitor Command Line

To access the Server Monitor commands, you need to start another command line.

On Windows NT, run the file SMCMD.BAT.

On UNIX, run the file SMCMD.SH.

Administering Server Monitor Commands

Use the following command line commands to administer the Server Monitor.

Server Monitor Commands

Command	Definition
`HELP`	Display the server monitor commands.
`SHUTDOWN`	Shutdown the server monitor and server monitor command line tool.
`STATUS`	Display server monitor status information.
`GC`	Initiate server monitor Garbage Collect.
`RSTART(serverName)`	Start a remote server.
`RSTOP(serverName)`	Stop a remote server.
`RSTATUS(serverName)`	Show the status of a remote server.
`RSTATUS`	Display the status of all servers that are monitored by the server monitor.
`RLIST`	Display a list of servers installed on different hosts.

Server Monitor Commands (Cont.)

Command	Definition
STOPCMDLINE	Stop the command line interface.

Auto-Restarting Servers with the Server Monitor

You can configure your servers to be re-started automatically on a crash. Currently only the Inbound Telephony Server, the Oracle Telephony Manager, the Oracle Telephony Media Controller(s), the Email Center server, the Oracle Routing Server(s) connect to the server monitor and can avail of this fault tolerance functionality. The Oracle Server Launcher must be running on every node that has any of the aforementioned servers connected to the server monitor. *By default Server Monitor is configured to restart all servers automatically on a crash.* To disable auto-restart for all servers set the parameter *auto_restart_disabled* to true in the sm.ini file. To disable auto- restart for a specific server set the parameter auto_restart_ disabled to true in the servers init file (svr.ini). Example configurations for init files follow:

Configuration 1

```
auto_restart_disabled=true in sm.ini
```

Result:No server will be restarted automatically.

Configuration 2

```
auto_restart_disabled=false in sm.ini
auto_restart_disabled=true in ors.ini (the Oracle Routing Servers init
file), auto_restart_disabled=false in otm.ini (the Oracle Telephony
Managers init file)
```

Result: If the Oracle Routing Server crashes, it will not be restarted automatically

Restarting a Crashed Server Application

Before you restart a crashed server application on the same server, you need to wait the length of time equal to the server refresh rate, which is currently one minute. If you restart sooner, you will receive the error message `Server by name "ServerName" is already running`. If you shut down the Windows NT

service or, on Solaris, invoke the “shutdown” command off the command line interface, you do not need to wait before restarting.

Configuring Oracle Telephony Manager to Run in Standalone Mode

To configure Oracle Telephony Manager to run in standalone mode, use the following guidelines.

- Define an Oracle Telephony Manager server definition in the Server Admin screen.
- Set the STANDALONE parameter to `true` in the Server Admin screen.

Configuring Oracle Telephony Manager to Run in Distributed Mode

To configure Oracle Telephony Manager to run in distributed mode, use the following guidelines.

- Define an Oracle Telephony Manager server definition in the Server Admin screen.
- Delete or set the STANDALONE parameter to `false` in the Server Admin screen.

Configuring Media Controller to Run in Distributed Mode

To configure Oracle Telephony Media Controller to run in distributed mode, use the following guidelines.

- Define an Oracle Telephony Media Controller server definition in the Server Admin screen, and put it in the same group as Oracle Telephony Manager.
- Set the parameter OTM_SERVER_NAME to the server name of Oracle Telephony Manager.

Connecting Oracle Telephony Manager to Routing Servers

You can configure up to three routing servers per Oracle Telephony Manager. A load-balancing algorithm distributes the calls among the routing servers.

To configure Oracle Telephony Manager to connect to routing servers, configure the parameter ROUTING_SERVER_1, ROUTING_SERVER_2, ROUTING_SERVER_3 with the appropriate routing server names.

Connecting Inbound Telephony Server to Oracle Telephony Manager

Inbound Telephony Server is a server process which monitors and forwards inbound telephony and web callback requests to Telephony Manager Server. The Inbound Telephony Server is required when running OTM in active mode, routing inbound telephony and/or web callback requests.

To configure Inbound Telephony Server to connect to Oracle Telephony Manager, configure the parameter `OTM_SERVER_NAME` with the appropriate Telephony Manager Server name.