Genesys Info Mart 8.0

User's Guide

About Genesys

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

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

Notice

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

Your Responsibility for Your System's Security

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

Trademarks

Genesys, the Genesys logo, and T-Server are registered trademarks of Genesys Telecommunications Laboratories, Inc. All other trademarks and trade names referred to in this document are the property of other companies. The Crystal monospace font is used by permission of Software Renovation Corporation, www.SoftwareRenovation.com.

Technical Support from VARs

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

Technical Support from Genesys

If you have purchased support directly from Genesys, please contact Genesys Technical Support at the regional numbers provided on page 9. For complete contact information and procedures, refer to the ***Genesys Technical Support Guide***.

Ordering and Licensing Information

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

Released by

Genesys Telecommunications Laboratories, Inc. www.genesyslab.com

Document Version: 80gim_us_02-2011_v8.0.101.00

Table of Contents

Preface

Welcome to the *Genesys Info Mart 8.0 User's Guide*. This document explains how to use data that is stored by Genesys Info Mart for contact-center historical reporting. The document shows which Dimension tables are associated with each Fact table, describes validated interaction flows that are used by Genesys Info Mart, and explains how these interactions are represented in the Genesys Info Mart database tables.

In brief, you will find the following information in this guide:

- An overview of table data
- Descriptions of how data that is related to interaction-handling attempts, interaction resources, interactions, mediation segments, and outbound-campaign contact attempts is populated
- Validated voice-interaction flows
- Validated multimedia-interaction flows
- Explanation of how to use the voice-of-data aspect of data lineage
- How dates and times of day are represented

The information is intended for end-users of Genesys Info Mart and is valid only for the Genesys 8.0 version of the software release.

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

This preface contains the following sections:

For information about related resources and about the conventions that are used in this document, see the supplementary material starting on page 155.

About Genesys Info Mart

Genesys Info Mart produces a data mart that you can use for contact center historical reporting.

Genesys Info Mart includes a server component, administration graphical user interface (GUI), and database. The Genesys Info Mart server runs a set of predefined jobs to:

- Extract data that has been gathered by Interaction Concentrator from data sources such as Configuration Server, T-Server, Interaction Server, and Outbound Contact Server. Genesys Info Mart stores this low-level interaction data, which is consolidated from Interaction Concentrator databases (Interaction Databases [IDBs]), in the Info Mart database.
- Transform the low-level interaction data and load it into a dimensional model (or star schemas) in the Info Mart database.

Genesys Info Mart can also be configured to host an aggregation engine that aggregates or re-aggregates the data, and populates Aggregate tables in the Info Mart database.

You query the Fact and Dimension tables in the dimensional model, using Structured Query Language (SQL), to obtain results that enable you to examine the data in detail, identify patterns, and predict trends for your organization.

Intended Audience

This guide is primarily intended for business users who want to query the data and for business-application developers who want to develop business-intelligence applications that query the data. The guide assumes that you have a solid understanding of database-management systems and structured query languages (such as SQL). Familiarity with CTI (computer-telephony integration) concepts, processes, terminology, and applications would also be helpful as would a basic understanding of the Genesys Framework—its architecture and functions.

Making Comments on This Document

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

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

Representative or Genesys Technical Support if you have suggestions about the product itself.

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

Contacting Genesys Technical Support

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

Region	Telephone	E-Mail
North America and Latin America	+888-369-5555 (toll-free) +506-674-6767	support@genesyslab.com
Europe, Middle East, and Africa	+44-(0)-1276-45-7002	support@genesyslab.co.uk
Asia Pacific	+61-7-3368-6868	support@genesyslab.com.au
Malaysia	1-800-814-472 (toll-free) +61-7-3368-6868 (International)	support@genesyslab.com.au
India	000 800 100 7136 (toll-free) +61-7-3368-6868 (International)	support@genesyslab.com.au
Japan	+81-3-6361-8950	support@genesyslab.co.jp
Before contacting technical support, refer to the *Genesys Technical Support Guide* for complete contact information and procedures.		

Document Change History

This section lists topics that are new or that have changed significantly since the first release of this document.

New in Document Version v.8.0.101.00

The document has been updated to support Genesys Info Mart release 8.0.1. The following topics have been added or significantly changed since the previous release of this document:

- References to multimedia interaction processing throughout the document have been updated as required, to reflect that Genesys Info Mart now transforms 3rd Party Media interactions.

- A note on page 33 describes new functionality with regard to new interaction subtypes that Genesys Info Mart might encounter during transformation.
- To support reporting on Outbound Contact Preview dialing:
  - Information about the `RESOURCE_GROUP_COMBINATION_KEY`, which has been added to the `CONTACT_ATTEMPT_FACT` dimension table, has been included in "Populating Contact Attempt Facts and Dimensions" on page 43.
  - A note on page 44 explains how you can use various fields in the `CONTACT_ATTEMPT_FACT` and `INTERACTION_FACT` tables to calculate data that is no longer populated in columns in the `CONTACT_ATTEMPT_FACT` table.

Chapter

1 Genesys Info Mart Overview

Genesys Info Mart uses multidimensional modeling to create a constellation of star schemas. These star schemas create a database for storing contact center data that can be retrieved using queries. Star schemas support queries that speed the retrieval of the stored data. Querying the data helps you uncover trends, chart heavy usage times, and reveal patterns in your contact center. In this way, Genesys Info Mart can help you:

- Determine how to measure the efficiency of your contact center in comparison with targeted service goals.
- Determine how best to staff your contact center.
- Understand customer preferences and problem trends.

This chapter contains the following sections:

Genesys Info Mart Data

Genesys Info Mart 8.0 extracts data from one or more Genesys Interaction Concentrator (ICON) databases and produces a data mart for contact center historical reporting. Genesys Info Mart yields data that is read-only and historical (representing some period of time).

The Info Mart database consists of the Global Interaction Database (GIDB) tables, Fact and Dimension tables (*dimensional model*), Merge Interaction

Database (MIDB) tables (used for voice interactions only), Control tables, the Staging area, and Temporary tables.

Note: GIDB provides the possibility for custom reporting or for drill-down reports from the dimensional model.

This guide focuses on the Fact and Dimension tables, as they are the primary sources of reporting data.

Note: This guide does not provide information about the Aggregate tables. For information about the Aggregate tables, see the *Reporting and Analytics Aggregates Reference Manual*.

Fact Tables

Fact tables are the large tables in the middle of a star schema. They represent business measures—for example, how long customers waited in a queue, how long and how often agents put customers on hold, or how long agents talked to customers. Fact tables are surrounded by a set of slowly changing Dimension tables. Fact tables represent a many-to-many relationship between dimensions; that is, there are many facts in a single Fact table, and they are related to many dimensions in various Dimension tables. Fact tables reference dimensions by using surrogate key columns.

Dimension Tables

Dimension tables describe the attributes of the associated Fact table. For example, the dimensions that are related to interactions might include the date and time when each interaction started, the required skills for various service types requested by customers, and the value of various customers to the business.

Data Aggregation

An aggregation engine creates Aggregation tables and aggregates data in environments in which either Genesys Interactive Insights (GI2) reports or Reporting and Analytics Aggregates (RAA) package are deployed. These Aggregate tables are documented in the *Reporting and Analytics Aggregates Reference Manual*.

Subject Areas

Table 1 describes the Genesys Info Mart subject areas. Each subject area is a star schema. For more information about the Fact and Dimension tables that are contained in each subject area, refer to the *Genesys Info Mart Reference Manual* for your RDBMS.

Table 1: Genesys Info Mart Subject Areas

Subject Area	Description
Interaction	Represents interactions from a customer-experience perspective.
Mediation Segment	Represents queue interaction activity. For voice, it represents ACD and virtual queue activity. For multimedia, it represents interaction queue, workbin, and virtual queue activity.
Place Group	Represents the membership of places among place groups.
Resource Group	Represents the membership of contact-center resources among resource groups.
Resource Session	Represents an agent-resource login session relative to a given media type (and also relative to DN-queue combination for voice media).
Resource Skill	Represents the skill resumes of agent resources.
Resource State	Represents an agent-resource state relative to a given media type (and relative to Place for voice media).
Resource State Reason	Represents an agent-resource state reason relative to a given media type (and relative to DN for voice media).
Campaign Group Session	Represents campaign-group sessions being loaded and unloaded.
Campaign Group State	Represents campaign-group sessions going through states, such as `Loaded`, `Started`, and `Unloading`.
Calling List Metric	Represents snapshots of outbound-campaign calling-list metrics.
Campaign Group to Campaign	Represents the associations between agent groups or place groups and outbound campaigns.
Calling List to Campaign	Represents the associations between outbound-campaign calling lists and campaigns.
Contact Attempt	Represents outbound-campaign contact-record attempts.
Interaction Resource	Represents all stages of interaction handling.
Interaction Resource State	Represents the states that a resource was in during interaction handling.

Table 1: Genesys Info Mart Subject Areas (Continued)

Subject Area	Description
Summarized Resource Session	Represents a contiguous period of time that an agent is logged onto any entity (switch, DN, or queue) relative to a given media type.
Summarized Resource State	Represents a summarized agent-resource state relative to a given media type.
Summarized Resource State Reason	Represents a summarized agent-resource state reason relative to a given media type.

Subject Area Example

As an example of a subject area, Figure 1 depicts the `INTERACTION_RESOURCE_FACT` table with some of its related Dimension tables. There are many data fields in each Fact and Dimension table; however, for the sake of simplicity, Figure 1 shows only a few of the fields in the `INTERACTION_RESOURCE_FACT` table and omits the fields that make up the Dimension tables. For information about all the fields that make up the facts and dimensions, refer to the *Genesys Info Mart Reference Manual* for your RDBMS.

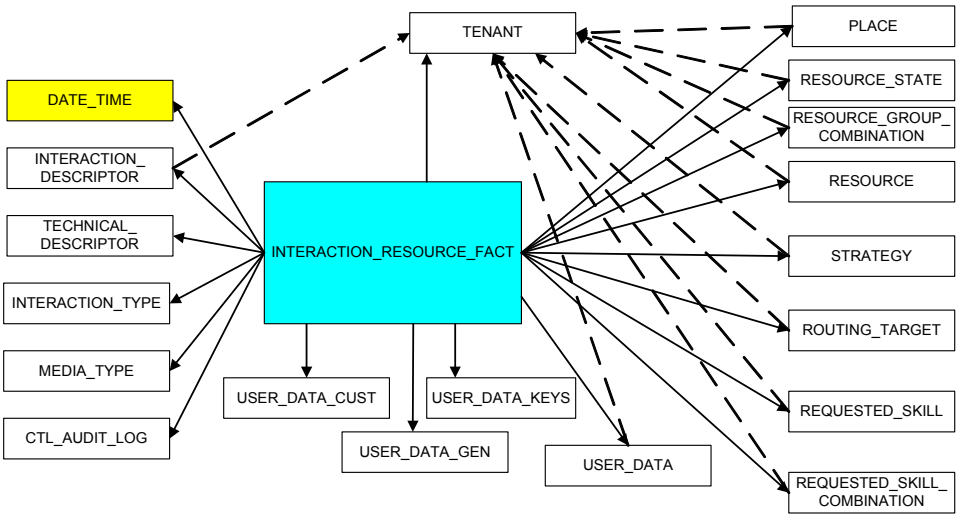

Figure 1: Sample Subject Area—Interaction Resource Fact

Bus Matrix

Figure 2 on page 16 maps the relationships between Genesys Info Mart Fact and Dimension tables in a *bus matrix*.

The bus matrix represents dimensionality of Fact tables in Genesys Info Mart as consolidated tabular views. It enables you to see the full dimensionality of each Fact table easily.

Fact table names are listed in columns in the matrix; Dimension table names are listed in rows.

The matrix excludes the `TENANT` and `DATE_TIME` dimensions, which map to all Fact tables. It also excludes the media-specific interaction and interaction resource tables, as well as the `CTL_AUDIT_LOG` table (see Chapter 2 on page 23).

For information about all the fields that make up the facts and dimensions, refer to the *Genesys Info Mart Reference Manual* for your RDBMS. For information about aggregates, see the *Reporting and Analytics Aggregates Reference Manual*.

DIMENSION TABLES / FACT TABLES	CALLING_LIST_METRIC_FACT	CALLING_LIST_TO_CAMP_FACT	CAMPAIGN_GROUP_SESSION_FACT	CAMPAIGN_GROUP_STATE_FACT	GROUP_TO_CAMPAIGN_FACT	CONTACT_ATTEMPT_FACT	INTERACTION_FACT	INTERACTION_RESOURCE_FACT	IXN_RESOURCE_STATE_FACT	MEDIATION_SEGMENT_FACT	PLACE_GROUP_FACT	RESOURCE_GROUP_FACT	RESOURCE_SKILL_FACT	SM_RES_SESSION_FACT	SM_RES_STATE_FACT	SM_RES_STATE_REASON_FACT
ATTEMPT_DISPOSITION						X										
CALL_RESULT						X										
CALLING_LIST	X	X				X										
CAMPAIGN	X	X	X	X	X	X										
CAMPAIGN_GROUP_STATE				X												
CONTACT_INFO TYPE						X										
CURRENCY							X									
DIALING MODE						X										
GROUP_			X	X	X	X					X	X				
INTERACTION_DESCRIPTOR								X								
INTERACTION_RESOURCE_STATE									X							
INTERACTION_TYPE							X	X	X	X						
MEDIA_TYPE						X	X	X	X	X				X	X	X
PLACE						X		X	X		X					
RECORD_STATUS						X										
RECORD_TYPE						X										
RECORD_FIELD_GROUP1						X										
RECORD_FIELD_GROUP2						X										
REQUESTED_SKILL								X								
REQUESTED_SKILL_COMBINATION								X								
RESOURCE_						X		X	X	X		X	X	X	X	X
RESOURCE_GROUP_COMBINATION								X		X				X	X	X
RESOURCE_STATE								X							X	X
RESOURCE_STATE_REASON																X
ROUTING_TARGET								X								
SKILL													X			
STRATEGY								X								
TECHNICAL_DESCRIPTOR								X		X						
TIME ZONE						X										

Figure 2: Bus Matrix of Fact and Dimension Tables

Chapter

2 Populating Genesys Info Mart Data

This chapter describes how Genesys Info Mart populates the data in the Genesys Info Mart database. You need this information in order to create meaningful queries for business purposes, as well as to interpret query results correctly.

This chapter contains the following sections:

For information about specific columns in Genesys Info Mart tables, see the *Genesys Info Mart Reference Manual* for your particular relational database-management system (RDBMS).

For descriptions of the sample queries provided with Genesys Info Mart, as well as for information about how you can customize the queries for your use, see the *Genesys Info Mart 7.6 SQL Queries Guide*.

Bringing Data into Info Mart: ETL

Extract, transform, and load (ETL) is performed by two main jobs, `Job_ExtractICON` and `Job_TransformGIM`.

Deployments in which Genesys Interactive Insights (GI2) or Reporting and Analytics Aggregates (RAA) is installedalso use `Job_AggregateGIM`.

`Job_ExtractICON` extracts new and changed data from Interaction Databases (IDBs), and stores it in the GIDB tables, as discussed in "Populating Low-Level Details" on page 18.

`Job_TransformGIM` transforms the data from GIDB into the dimensional-model (Fact and Dimension) tables.

`Job_AggregateGIM` calculates or recalculates metrics and stores them in the aggregate tables in the Info Mart database, based on the data that was added or changed during the last transform run.

Note: For detailed information about `Job_ExtractICON`, `Job_TransformGIM`, and `Job_AggregateGIM` and how they function, see the *Genesys Info Mart 8.0 Operations Guide*. For detailed information about the aggregation process, see the Reporting & Analytics Aggregates documentation set.

Populating Low-Level Details

The Global Interaction Database (GIDB) is an area within the Genesys Info Mart database schema in which the low-level interaction data from any number of Interaction Databases (IDBs) is consolidated for further processing.

To populate GIDB, Genesys Info Mart Server extracts data from one or more source IDBs. For voice-interaction records, the merge operation links all records that are related to the same interaction, in both single-site and multi-site deployments. The server loads all extracted (and, if applicable, merged) data into the GIDB.

The GIDB:

- Represents a subset of IDB tables, to better align the lowest level of data details in Genesys Info Mart with the Interaction Concentrator model.
- Provides low-level details about a call, party, and party history for voice and multimedia interactions in the `GIDB_G_CALL`, `GIDB_G_PARTY`, and `GIDB_G_PARTY_HISTORY` tables, respectively.
- Extracts all records that are necessary for Genesys Info Mart reporting purposes from various IDBs, to gather coherent reporting data at the lowest level of details from the entire contact center in a single data warehouse.
- Uses special fields to indicate from which IDB data was extracted.

- Stores the data as long as it is required by customers after Genesys Info Mart further processes (transforms) GIDB data.

Note: The term *voice interactions* refers to traditional telephony calls. The term *multimedia interactions* refers to interactions that are processed through the Genesys eServices/Multimedia solution, including 3rd Party Media interactions.

Genesys Info Mart Server uses the low-level details data from GIDB tables to produce data that is suitable for end-user reports and to populate the Fact and Dimension tables that compose the Info Mart dimensional model.

The *Genesys Info Mart 8.0 Reference Manual* for each supported RDBMS provides a list of GIDB tables. The meaning of the data in each row within a given GIDB table is the same as in the corresponding IDB record. For example, `GIDB_GC_PLACE` table in the Info Mart database corresponds to the `GC_PLACE` table in IDB. Refer to the *Interaction Concentrator 8.0 Physical Data Model* document for your RDBMS for information about the data stored in corresponding GIDB tables.

The DATE_TIME Dimension

The `DATE_TIME` dimension serves as a sort of calendar that establishes the dates and times for which data is populated.

All interaction-related Fact tables use only the `DATE_TIME` time dimension. No other time-dimension fields are used.

Note: Only UTC timestamps are used in the interaction-related Fact tables.

By default, a single `DATE_TIME` table is configured, but you can set up multiple tables. For example, you might need to support multiple time zones. For details on how to configure multiple `DATE_TIME` tables, see the *Genesys Info Mart 8.0 Deployment Guide*.

The `DATE_TIME` dimension is discussed in greater detail in Chapter 6, "Representing Dates and Times of Day," on page 147.

Populating Interaction Resource Data

Genesys Info Mart stores interaction resource facts in the INTERACTION_RESOURCE_FACT (IRF) table, one of the core tables that is supplied in Genesys Info Mart.

This table facilitates the creation of reports and serves as one of the primary tables from which aggregation tables are populated. (See the Reporting & Analytics Aggregates documentation set for details on aggregation tables.)

Genesys Info Mart creates interaction resource facts to represent the involvement of a contact-center handling resource in an interaction. *Handling resources* are the resources that have the greatest interest for reporting—agents, self-service IVRs, and DNs without an agent.

IRF resources also include mediation resources in which the IRF ends in mediation (such as queues, routing points, and nonself-service IVRs).

The IRF table supplies a single row within the Genesys Info Mart schema, which simplifies the SQL needed to generate reports on the resources that handle interactions within the contact center.

Each interaction resource fact represents:

- The contiguous time span of the association between the resource and the interaction.
- The particular role played by the resource (the *resource role*).
- The result of the association from the perspective of the resource (the *technical result*).

Interaction-resource facts are created for complete voice interactions and for both completed and active multimedia interactions.

The IRF table:

- Simplifies report queries by integrating conference and consult durations into the original handling resource row.
- Summarizes the total queue, route point, and IVR wait times prior to the handling resource and stores them with the handling-resource row in separate columns.
- Stores response duration per routing attempt, in addition to the initial routing sequence.
- Records the state of the resource immediately prior to involvement in the interaction, thus enabling reporting of interactions received or initiated during an AfterCaLLwork or NotReady agent state.

Genesys Info Mart uses the following additional tables to support the IRF table:

- The IXN_RESOURCE_STATE_FACT table contains all the individual states, durations and interval clips for each state the interaction-fact resource was in during the interaction.

- The INTERACTION_RESOURCE_STATE Dimension table contains the states defined for the resource that is handling the interaction.

The major factor used to determine whether or not to create a row in the table is whether the new interaction or a new attempt to handle an existing interaction has been started, or whether interaction arrived at a handling resource, or *resource of interest*.

Resources of interest in the context of interaction resource fact rows include the following:

- Agents
- IVRs (indicated as self-service IVR applications via attached data)
- Extensions and ACD positions with no Person object associated.

Note: For detailed information about the columns in the INTERACTION_RESOURCE_FACT table, see the *Genesys Info Mart Reference Manual* for your RDBMS.

Populating Interaction Resource Facts and Dimensions

The following sections describe how Genesys Info Mart populates interaction resource facts.

Each interaction resource fact row includes all prior queue, routing point, and IVR (nonself-service) counts and durations that were part of the distribution of the interaction to the resource.

Interaction-resource facts represent either the processing of interactions by handling resources (such as agents, self-service IVRs, and extensions/positions without associated agents) or unsuccessful attempts to reach such a handling resource (resulting in the interaction being abandoned in queue or abandoned in routing).

The grain of the fact is an accumulating snapshot of the contiguous participation of a contact-center handling resource in interaction processing, including time spent wrapping up the interaction. Movement of a resource from one call to another does not cause creation of a new interaction resource fact, but is accumulated in a single fact. For example, when the transferredTo resource in a transfer scenario is moved from a consult call to the original call, this movement is represented in a single fact.

However, if a handling resource is participating in parallel calls, the resource is represented by two separate facts. For example, in a consultation call scenario there are two facts for the consulting resource, one for the existing call and one for the consultation call.

Dimensions Associated with the IRF Table

- Interaction-resource fact start and end dates and times are stored as UTC timestamps (START_TS and END_TS) and as references to the DATE_TIME dimension (START_DATE_TIME_KEY and END_DATE_TIME_KEY).

  Media-neutral counts and durations are provided. These categorize the time spent on various activities, such as time spent in a queue, time spent handling the interaction, and time spent wrapping up the interaction. Because not all IRFs involve a customer directly, separate counts and durations are included to reflect the time that the customer spent waiting versus being helped.

  Note: For more information about how Genesys Info Mart represents dates and times of day, see Chapter 6 on page 147.

- The RESOURCE_ dimension indicates the routing point, queue, IVR port, or agent that either initiated or handled this resource fact.

  Note: The RESOURCE_ dimension actually has two references, RESOURCE_KEY and MEDIA_RESOURCE_KEY, which typically refer to the same resource. The following are exceptions:
  - For IVRs, RESOURCE_KEY is for the IVR Application Name and MEDIA_RESOURCE_KEY for the associated DN.
  - For Agents, RESOURCE_KEY is for the Agent, and MEDIA_RESOURCE_KEY for the associated DN.

- The PLACE dimension indicates the place at which the IRF was processed.
- The TENANT dimension identifies the tenant of the resource.
- The TECHNICAL_DESCRIPTOR dimension identifies the resource role and technical result of the IRF. For information about the resource roles and technical results for interaction resources, see Table 2 on page 26.
- The INTERACTION_DESCRIPTOR dimension identifies the customer segment (indicating the value of the customer), the type of service being requested, and the business result of the IRF.
- The STRATEGY dimension identifies the Genesys routing strategy or IVR application that processed the IRF.
- The ROUTING_TARGET, REQUESTED_SKILL, and REQUESTED_SKILL_COMBINATION dimensions indicate the Genesys Universal Routing Server's activities by identifying the target that was selected and the list of skills that were required to process the IRF.
- The CUSTOMER dimension represents the ID of the customer that is involved in the interaction.

User Data

As previously indicated, many interaction attributes are formally modeled. However, deployment-specific attributes, in the form of *user-defined attached data*, are also represented in the model.

Genesys Info Mart provides unified user-data processing from both call-related `EventUserEvents` and call-based `TEvents`, with a flexible data storage that you can configure according to the number and types of user data captured in your contact center environment. A customizable database schema enables you to treat each key-value pair (KVP) field as a fact, a dimension, or both, and store user-data KVPs in a configurable number of user data dimensions and facts that are associated with core Fact tables. Genesys Info Mart also processes the user data that arrives after call completion and updates call records accordingly.

User data can be stored as facts or dimensions. High-cardinality user data is stored as facts. Low-cardinality user data is most efficiently stored as dimensions. You can create up to 800 custom low-cardinality user data dimensions. The only limits on the quantity of high-cardinality user data that you store are performance-based. There are no absolute limits.

- *High-cardinality user data*—Data for which there can be a very large number of possible values. High-cardinality user data types are predefined, but the number of KVPs is configurable. Each interaction has no more than one value for each KVP. A Customer ID number is an example of high-cardinality user data.
- *Low-cardinality user data*—Data that has a limited range of possible values. Customer Segment, Service Type, and Service Subtype are good examples of low-cardinality user data. For example, in a `CUSTOMER` table with a column named `NEW_CUSTOMER`, this column would contain only two distinct values, `Y` or `N`, which respectively denote whether the customer was new or not. Because only two possible values are held in this column, its cardinality type is low cardinality.

High-cardinality user data requires only a single join from the IRF table. Low-cardinality user data that is stored as dimensions require two joins, one to the User Data Keys table and another to the Dimension table.

You can use the same key as both fact and dimension. Genesys provides templates for you to configure your own User Data keys.

The CTL_AUDIT_LOG Dimension Table

The new `CTL_AUDIT_LOG` Dimension table contains data for all transactions that are committed by Genesys Info Mart. This table replaces the `AUDIT_` Dimension table. Instead of service fields such as `ROW_CREATED` and `ROW_UPDATED` appearing in all tables, the `CTL_AUDIT_LOG` table contains audit information for all records.

All Fact table records now contain pointers (`CREATE_AUDIT_KEY` and `UPDATE_AUDIT_KEY`) to the relevant `CTL_AUDIT_LOG` table row.

Each row represents a logical transaction that is committed by Genesys Info Mart, identifying the ETL job involved in the transaction and including the minimum and maximum `DATE_TIME` values (which give the date-time range for the data that is committed in the transaction), and providing the processing status (an internal indicator of the kind of data that is processed).

Abandoned and Terminated Interactions

To represent every interaction in the IRF table, rows are created to represent attempts to reach a resource of interest. These rows contain data about queues, routing points, and routing queues in which the interaction has been abandoned in the distribution device by the customer, during a consultation, or during an internal call that was initiated by a resource of interest.

Abandoned Interactions

Abandoned interactions are identified as interactions in which the last resource that was involved was not a handling resource.

In such cases a row is created to represent an attempt to reach another handling resource. This interaction resource fact row contains data from all prior related mediation device segments that were involved with the attempt to reach another handling resource.

Interactions Terminated in a Mediation IVR or DN (No IVR or Agent Resource Association)

A *mediation IVR* in the context of the IRF table is an IVR resource that is not considered to be self-service because the IVR application (or a URS strategy on its behalf) did not set attached data to indicate self-service. An interaction that terminates in a mediation IVR is considered to be abandoned.

Populating Resource Roles and Technical Results

Understanding when interaction resource facts are created can help you to determine which types of interaction resources to include in, or exclude from your queries. It is important that you understand the types of resources and the resource roles, role reasons, technical results, and technical result reasons for each resource.

The following subsections describe the resource roles and technical results that are applicable for both voice and multimedia.

Resource Roles and Technical Results

This section describes the resource roles and technical results for the following types of interaction resources:

- IVR—Genesys Info Mart creates interaction resource facts each time that a self-service IVR is associated with a voice interaction.
- Agent—Genesys Info Mart creates interaction resource data each time that an agent is associated with an interaction.
- Extensions and ACD positions with no Person object associated—Genesys Info Mart creates interaction resource data each time that an extension, ACD position is associated with an interaction. The resource role and technical result for these interaction resource data are similar to those for Agent interaction resource data.

After you understand these resources and their roles, see Chapter 3 on page 63 for diagrams that depict the interaction resource facts that result from typical voice interaction flows and Chapter 4 on page 121 for diagrams that depict the interaction resource facts that result from typical multimedia interaction flows.

Resource Roles, Technical Results, and Technical Result Reasons

The resource role of the interaction resource depends on how the interaction arrives at the resource. The technical result and technical result reason of the interaction resource depend on how the interaction leaves the resource. See Table 2 for the roles, results, and reasons that are used for the IRF table.

Table 2: Resource Roles, Technical Results, and Reasons

Resource Role	Role Reason	Technical Result	Result Reason	Comment
Received	Unspecified	Customer Abandoned	Abandoned from Hold	Denotes that the resource received an inbound interaction without the benefit of prior distribution devices moving the call to it. This is typical for internal call types that are dialed directly to the resource. `Received/Conferenced` in the IRF context indicates the initiator of a conference call. Interactions that end in a nonself-service IVR have the value `CustomerAbandoned/AnsweredByOther` to indicate that the customer abandoned the call before service could be provided.
	Unspecified	Customer Abandoned	Abandoned while Queued	
	Unspecified	Customer Abandoned	Abandoned while Ringing	
	Unspecified	Redirected	Route on no Answer (RONA)	
	Unspecified	Transferred	Unspecified	
	Unspecified	Conferenced	Unspecified	

Table 2: Resource Roles, Technical Results, and Reasons (Continued)

Resource Role	Role Reason	Technical Result	Result Reason	Comment
Received Transfer	Unspecified	Customer Abandoned	Abandoned from Hold	Denotes that the IRF was created as a result of the resource being transferred to the interaction by a resource other than a non-self-service IVR port. `ReceivedTransfer/Conferenced` in the IRF context indicates the initiator of a conference call. Interactions that end in a nonself-service IVR have the value `CustomerAbandoned/AnsweredByOther` to indicate that the customer abandoned the call before service could be provided.
	Unspecified	Customer Abandoned	Abandoned while Queued	
	Unspecified	Customer Abandoned	Abandoned while Ringing	
	Unspecified	Redirected	Route on no Answer (RONA)	
	Unspecified	Redirected	Unspecified	
	Unspecified	Abandoned	Unspecified	
	Unspecified	Completed	Unspecified	
	Unspecified	Transferred	Unspecified	
	Unspecified	Conferenced	Unspecified	

Table 2: Resource Roles, Technical Results, and Reasons (Continued)

Resource Role	Role Reason	Technical Result	Result Reason	Comment
Received Consult	Unspecified	Redirected	Route on no Answer (RONA)	Denotes that the IRF was created for a resource as the result of a consultation only (the resource did not receive a transfer, or was not joined into a conference). This enables counting of consultations that are received by a resource. The `ReceivedConsult/Conferenced` row represents the unlikely event that a resource receives a consultation, consults another resource, and then then creates a conference call between the resources. This combination in the IRF context indicates the initiator of a conference call.
	Unspecified	Redirected	Unspecified	
	Unspecified	Completed	Unspecified	
	Unspecified	Transferred	Unspecified	
	Unspecified	Abandoned	Unspecified	
	Unspecified	Conferenced	Unspecified	

Table 2: Resource Roles, Technical Results, and Reasons (Continued)

Resource Role	Role Reason	Technical Result	Result Reason	Comment
Routed To	Unspecified	Customer Abandoned	Abandoned from Hold	Denotes an interaction that was delivered to the resource via a routing point. `RoutedTo/Conferenced` in the IRF context indicates the initiator of a conference call. Interactions that end in a nonself-service IVR have the value `CustomerAbandoned/AnsweredByOther` to indicate that the customer abandoned the call before service could be provided.
	Unspecified	Customer Abandoned	Abandoned while Queued	
	Unspecified	Customer Abandoned	Abandoned while Ringing	
	Unspecified	Redirected	Route on no Answer (RONA)	
	Unspecified	Redirected	Unspecified	
	Unspecified	Abandoned	Unspecified	
	Unspecified	Completed	Unspecified	
	Unspecified	Transferred	Unspecified	
	Unspecified	Conferenced	Unspecified	
Diverted To	Unspecified	Customer Abandoned	Abandoned from Hold	Denotes an interaction that was delivered to the resource via an ACD queue. `DivertedTo/Conferenced` in the IRF context indicates the initiator of a conference call. Interactions that end in a nonself-service IVR have the value `CustomerAbandoned/AnsweredByOther` to indicate that the customer abandoned the call before service could be provided.
	Unspecified	Customer Abandoned	Abandoned while Queued	
	Unspecified	Customer Abandoned	Abandoned while Ringing	
	Unspecified	Redirected	Route on no Answer (RONA)	
	Unspecified	Redirected	Unspecified	
	Unspecified	Abandoned	Unspecified	
	Unspecified	Completed	Unspecified	
	Unspecified	Transferred	Unspecified	
	Unspecified	Conferenced	Unspecified	

Table 2: Resource Roles, Technical Results, and Reasons (Continued)

Resource Role	Role Reason	Technical Result	Result Reason	Comment
In Conference	Unspecified	Customer Abandoned	Abandoned while Ringing	Denotes that the IRF was created for a resource as the result of a conference call in which the resource joined the conference. `InConference/Conferenced` in the IRF context indicates that after joining the conference, the joining resource was the initiator of a subsequent conference.
	Conference Joined	Customer Abandoned	Abandoned while Ringing	
	Unspecified	Redirected	Route on no Answer (RONA)	
	Unspecified	Redirected	Unspecified	
	Unspecified	Transferred	Unspecified	
	Unspecified	Abandoned	Unspecified	
	Conference Joined	Abandoned	Unspecified	
	Conference Joined	Completed	Unspecified	
	Unspecified	Completed	Unspecified	
	Unspecified	Conferenced	Unspecified	
Initiated	Unspecified	Conferenced	Unspecified	Denotes that the resource in the IRF row initiated either an internal interaction or an outbound interaction. `Initiated/Conferenced` in the IRF context indicates that the resource initiated a call and was the initiator of a conference call.
	Unspecified	Transferred	Unspecified	
	Unspecified	Completed	Unspecified	
	Unspecified	Abandoned	Unspecified	
	Unspecified	DestinationBusy	Unspecified	

Multimedia Stop Reason System Names

One of the reporting event attributes captured by eServices Interaction Server is the reason system name (`attr_reason_system_name`) associated with a request. The reason system name associated with a Stop Processing request is of particular significance and is captured by ICON in the `G_STOP_REASON` column of the `GM_L_USERDATA` table. There are certain Stop Processing reason names which are meaningful to Genesys Info Mart to correctly report the Technical Result:

Abandoned

In Media Server compatibility mode (described in the information on the Chat Server configuration option, `stop-abandoned-interaction`, in the eServices/Multimedia documentation), a chat interaction is stopped with a reason system name of `Abandoned` when it is abandoned by the customer. Genesys Info Mart uses this stop reason to determine if a chat interaction has been abandoned.

Sent

When an outbound-sending e-mail strategy sends an e-mail outside of the contact center, by convention, as illustrated in the Interaction Workflow Samples, the Strategy stops the outbound e-mail interaction with a reason system name of `Sent`. Genesys Info Mart relies upon this convention, and uses this stop reason to determine if an outbound e-mail was actually sent.

Normal

A stop reason of `Normal` may be used in a large variety of contexts, but there is only one scenario where its use affects Genesys Info Mart processing.

When an agent transfers a chat interaction to a Chat Transcript Queue, a Chat Transcript Strategy pulls the interaction from the queue, and decides whether or not to send an e-mail transcript of the chat interaction, based upon user data attached by Genesys Agent Desktop, and, by convention, as illustrated in the Interaction Workflow Samples, stops the chat interaction with a reason system name of `Normal`. Genesys Info Mart relies upon this convention to determine how to represent the action of the agent that transferred the chat interaction to the Chat Transcript Queue. In this case, the agent, who transferred the interaction to the Chat Transcript Queue, is not attempting to *transfer* the interaction to another resource, but instead has *completed* the chat activity, and the transfer action is to engage follow-up workflow processing.

Populating Interaction Data

Genesys Info Mart creates interaction facts to link together all facts related to a given interaction. Interaction facts represent interactions from the perspective of the customer experience.

Each interaction fact represents:

- The time span of the overall interaction
- Information that identifies the interaction parties
- Service indicators.

Interaction facts can also be linked to the user data extension tables through keys.

Interaction Fact Table

Genesys Info Mart stores both voice and non-voice interaction facts in the `INTERACTION_FACT` table.

For detailed information about the columns in the interaction Fact table, see the *Genesys Info Mart Reference Manual* for your RDBMS.

Populating Interaction Facts and Dimensions

Genesys Info Mart populates voice and multimedia interactions in the following ways:

- The `TENANT` dimension is inherited from the underlying interaction resource fact that has the lowest ordinal. This is the first resource facts that was created for the interaction, and it generally has the earliest start time. In a network routing solution, all underlying network and premise facts are considered. If premise facts exist, the `TENANT` dimension is the tenant of the first premise fact; otherwise, the `TENANT` dimension is the tenant of the first network fact.

- The INTERACTION_TYPE and MEDIA_TYPE dimensions are inherited from the underlying interaction resource fact that has the lowest ordinal. This is the first resource fact that was created for the interaction, and it generally has the earliest start time. In a network routing solution, all underlying network and premise facts are considered.

Note: Any multimedia interaction subtype that you have configured in your environment but that is new to Genesys Info Mart is automatically added to the INTERACTION_TYPE table. Once it has been added, you can choose to have Genesys Info Mart disregard that subtype for all future transformation jobs by setting the appropriate value for the IGNORE field. By default, Genesys Info Mart transforms all interactions that have the newly added subtype.

New media types are also automatically added as Genesys Info Mart encounters them. By default, interactions that are associated with new media types are transformed as offline interactions. To set them as online interactions, enter the appropriate value in the IS_ONLINE field in the MEDIA_TYPE table.

For details, see the *Genesys Info Mart 8.0 Deployment Guide*.

Populating Mediation Segments

The mediation segment Fact table describes interaction activity that involves mediation DNs, such as virtual and ACD queues, or multimedia interaction queues and workbins. The *grain* spans the time from when the interaction entered the mediation DN to the time that the interaction was abandoned in the mediation DN, cleared from the mediation DN (virtual queue only), or distributed from the mediation DN, including the time that it takes the interaction to be answered by the target resource or to be abandoned while alerting at the target resource. For voice, only completed ACD and virtual queue activity is populated; for multimedia interactions, both active and completed interaction queue, workbin, and virtual queue activity is populated.

Mediation Segment Fact Table

Genesys Info Mart stores mediation segment facts in the MEDIATION_SEGMENT_FACT table. For detailed information about the columns in this table, refer to the *Genesys Info Mart Reference Manual* for your RDBMS.

Mediation Segments and Queues

A mediation segment fact is created each time that an ACD, a virtual queue, a multimedia interaction queue, or a workbin is used during interaction processing. For voice, mediation segments are populated in Genesys Info Mart only when the mediation segment is completed. For multimedia, both active and completed mediation segments are populated.

There are also links to the associated interaction resource fact, during which time the mediation that is represented by the mediation segment fact occurred.

Each mediation segment fact represents:

- The particular role played by the queue resource. For information about the resource roles that apply to queues, see “Resource Roles for Queues” on page 36.
- The result of the association from the perspective of the queue resource to the target resource is chosen during routing. For information about the technical results and technical result reasons that apply to voice (ACD and virtual queues), refer to Table 3 on page 37. For multimedia technical results and result reasons (interaction queue, workbin, or virtual queue), refer toTable 4 on page 40.

Note: The mediation segment fact describes virtual queue usage within your routing strategies. Virtual queue activity occurs within the context of a voice or multimedia interaction.

The activity is also captured in the associated interaction resource fact, for which there will be only one row.

Configuration Options Used to Control Population of Queue Activity

The `gim-etl-populate` section of the Genesys Info Mart `Application` object, contains options that enable or disable population of the `MEDIATION_SEGMENT_FACT` table.

Note: ACD queue and virtual queue activity is always populated.

- The `populate-mm-ixnqueue-facts` option enables or disables the population of eServices/Multimedia Interaction Queue activity.
- The `populate-mm-workbin-facts` option enables or disables the population of eServices/Multimedia Interaction Workbin activity.

Populating Mediation Segment Facts and Dimensions

Genesys Info Mart populates mediation segments in the following ways:

- The start time facts represent the start time of the mediation segment (when the interaction enters the queue).

  End time facts represent the end time of the mediation segment, which is one of the following:

  - The moment at which the interaction is abandoned while in the queue.
  - The moment at which the interaction is distributed from the queue to some target resource.
  - The moment at which the interaction is cleared from the queue, such as when a routing strategy routes the interaction from a parallel queue, or when it removes the interaction from the queue as it clears the routing targets for which it was waiting.

  For more information about how Genesys Info Mart represents dates, see Chapter 6 on page 147.
- The `TENANT` dimension identifies the tenant to which the queue resource belongs.
- The `RESOURCE_` dimension identifies the mediation DN resource that is associated with the mediation segment.
- The `TECHNICAL_DESCRIPTOR` dimension identifies the resource role and technical result of the mediation segment. For information about the resource roles and technical results that are applicable to ACD and virtual queues, see "Populating Resource Roles and Technical Results" on page 36.
- The `SHORT_ABANDONED_FLAG` indicates that, while waiting to be routed from the queue, the customer abandoned the interaction before the configured threshold expired. This enables these types of interactions to be filtered from the reports.
- The `MET_THRESHOLD_FLAG` indicates that the amount of time an interaction waited to be handled by a contact center resource was within a configurable threshold from the perspective of the queue. It is measured from the time that the interaction entered the queue to the time that it was answered by a contact center resource.
- The `ANSWER_THRESHOLD` contains the configured value used to calculate the `MET_THRESHOLD_FLAG` indicator.
- The `PLACE` dimension identifies the place that is associated with the target of the routing process.
- In addition to the mediation DN resource that is associated with the mediation segment, the `RESOURCE_` dimension identifies the contact center resource that was the routing target from the mediation DN.

- MEDIATION_DURATION is the length of time that the interaction was in the ACD queue, virtual queue (as indicated by URS), or interaction queue or workbin (as indicated by Interaction Server).
- ONLINE_DURATION is the period of time that the interaction was in the ACD, virtual queue, interaction queue, or workbin before the interaction went offline.
- The INTERACTION_TYPE and MEDIA_TYPE dimensions are inherited from underlying interaction resource fact that has the lowest ordinal. This is the first resource fact that was created for the interaction and it generally has the earliest start time. In a network routing solution, all underlying network and premise resource facts are considered.
- The RESOURCE_GROUP_COMBINATION dimension records the virtual queue or queue membership in one or more groups.

For voice only:

- TARGET_IXN_RESOURCE_ID provides a link between the mediation segment fact and the interaction resource fact that was the target of the routing process that is associated with the queue. This provides the means to associate the queue with the target of the routing strategy for virtual queue reporting.

Populating Resource Roles and Technical Results

Understanding when mediation segment facts are created can help you determine which types of mediation segments to include in—or exclude from—your queries. It is important that you understand the resource roles, technical results, and technical result reasons for each resource.

Resource Roles for Queues

The resource role of the mediation segment depends on how the voice or multimedia interaction arrives at the resource.

For an ACD queue, virtual queue, multimedia interaction queue, or multimedia workbin, each row in the mediation segment Fact table has a resource_role of Received or Received_Consult.

For ACD queues, Received_Consult indicates that the interaction arrived in the ACD queue as the result of an active consultation between contact center resources and was still in consultation when the interaction was diverted by the ACD queue row.

Technical Results and Technical Result Reasons for ACD and Virtual Queues for Voice Interactions

The technical result and technical result reason of the mediation segment depend on how the voice or multimedia interaction leaves the resource, as shown in Table 3 (voice) and Table 4 on page 40 (multimedia).

Table 3: Technical Results for ACD and Virtual Queues (Voice)

Technical Result	Technical Result Reason	How the Interaction Exited the Queue
Customer Abandoned	AbandonedWhile Queued	For virtual queues, the interaction was abandoned while in the virtual queue. For ACD queues, the mediation attempt through this ACD queue was abandoned while waiting for service.
Abandoned	Unspecified	For virtual queues, either the consulation was abandoned while in the virtual queue or a consultation was retrieved while in the virtual queue. For ACD queues, the consultation mediation attempt through this ACD queue was abandoned or retrieved while waiting for service.

Table 3: Technical Results for ACD and Virtual Queues (Voice) (Continued)

Technical Result	Technical Result Reason	How the Interaction Exited the Queue
Diverted	Unspecified	For virtual queues, the virtual queue diverted the interaction to a Routing Point, ACD queue, or a target resource that is unknown. For ACD queues, the interaction was diverted by the ACD queue and the target IRF row was another mediation device such as a (RP or an ACD queue).
	AnsweredByAgent	For virtual queues, the virtual queue diverted the interaction to a target resource which was an agent, and the agent answered the interaction. For ACD queues, the interaction was diverted by the ACD queue and the target IRF was an agent who had a talk count > 0.
	AnsweredByOther	For virtual queues, the virtual queue diverted the interaction to a target resource that was not an agent, but that could answer the interaction. For ACD queues, the interaction was diverted by the ACD queue and the target IRF was a resource other than an agent who had a talk count > 0 (typically an IVR or ACD Position DN).
	AbandonedWhile Ringing	For virtual queues, the virtual queue diverted the interaction to a target resource, but the interaction was abandoned before the target could answer it. For ACD queues, the interaction was diverted by the ACD queue, and the target IRF was a handling resource (Agent, IVR or ACD position DN) who had a talk count = 0 and route on no answer (RONA) did not occur.
	Redirected	For virtual queues, the virtual queue diverted the interaction to a target resource that did not answer it; as a result, the interaction was routed to another resource. For ACD queues, the interaction was diverted by the ACD queue and the target IRF was a resource that was routed on no answer (RONA'd) or forwarded the interaction elsewhere.

Table 3: Technical Results for ACD and Virtual Queues (Voice) (Continued)

Technical Result	Technical Result Reason	How the Interaction Exited the Queue
Cleared	Unspecified	For virtual queues, indicates that the interaction was cleared from the virtual queue because no target was found. For ACD queues, indicates that the interaction was parallel queued and was not diverted from this ACD queue to another call center resource.
	StuckCall	An interaction that Interaction Concentrator (ICON) identified as a stuck call was cleared from the virtual queue. (ICON determines that an interaction is stuck in a virtual queue if ICON received an event that indicates that the interaction entered the virtual queue, but it did not receive the event that indicates that the interaction exited the virtual queue, and URS has stopped sending status updates for that interaction.) **Note:** To calculate durations from virtual queue data accurately, Genesys recommends that rows that have this technical result and reason not be used.
	RoutedFromAnotherVQ	Applies only to virtual queues. Indicates that, while the interaction was simultaneously in virtual queues, the interaction was cleared from the specified virtual queue because it was routed by another virtual queue.
	DefaultRoutedByStrategy	Applies only to virtual queues. Indicates that the interaction was cleared from the virtual queue when URS default-routed the interaction.
	DefaultRoutedBySwitch	Applies only to virtual queues. Indicates that the interaction was cleared from the virtual queue when the switch default-routed the interaction.
	Targets Cleared	Applies only to virtual queues. Indicates that the interaction was cleared from the virtual queue as the result of the URS Clear Targets function.

Technical Results and Technical Result Reasons for Virtual Queues, Interaction Queues, and Workbins for Multimedia Interactions

The technical result and technical result reason of the mediation segment depend on how the multimedia interaction leaves the resource, as shown in Table 4 on page 40.

Table 4: Technical Results for Virtual Queues (Multimedia)

Technical Result	Technical Result Reason	How the Interaction Exited the Virtual Queue
Customer Abandoned	AbandonedWhile Queued	While in the virtual queue, interaction queue, or workbin, the interaction was abandoned by the customer. **Notes:** • In Media Server compatibility mode (for example, see the Chat Server stop-abandoned-interaction configuration option), the interaction is stopped when it is abandoned by the customer. In that case, the Abandoned technical result is used. • Genesys Info Mart does not report an interaction as abandoned if the Media Server is configured not to stop the interaction when it is abandoned.

Table 4: Technical Results for Virtual Queues (Multimedia) (Continued)

Technical Result	Technical Result Reason	How the Interaction Exited the Virtual Queue
Diverted	AnsweredByAgent	The interaction was diverted from the virtual queue, interaction queue, or workbin to a target resource that was an agent, and the agent answered the interaction.
	AnsweredByOther	The interaction was diverted from the virtual queue, interaction queue, or workbin to a target resource that was a place, and the place answered the interaction, but no agent was logged in to that place.
	AbandonedWhile Ringing	The interaction was diverted from the virtual queue, interaction queue, or workbin to a target resource, but the interaction was abandoned before the target could answer it.For multimedia interactions, Genesys Info Mart only reports that an abandon occurred if the Media Server was configured to stop the interaction when it was abandoned.
	Revoked	The interaction was diverted from the virtual queue, interaction queue, or workbin to a target resource that was an agent (or a place). The agent (or place) was invited into the interaction, but the invitation was not accepted before the `delivering-timeout` that was configured in Interaction Server. As a result, the interaction is placed back into the interaction queue from which it came.
	Rejected	The interaction was diverted from the virtual queue, interaction queue, or workbin to a target resource that was an agent (or a place). The agent (or place) was invited into the interaction, but the invitation was rejected. As a result, the interaction is placed back into the interaction queue from which it came.
	RoutedToOther	The interaction was diverted from the virtual queue, interaction queue, or workbin to a target resource that was an interaction queue or workbin.

Table 4: Technical Results for Virtual Queues (Multimedia) (Continued)

Technical Result	Technical Result Reason	How the Interaction Exited the Virtual Queue
Cleared	StuckCall	An interaction that Interaction Concentrator (ICON) identified as a stuck call was cleared from the virtual queue. (ICON determines that an interaction is stuck in a virtual queue if ICON received an event that indicates that the interaction entered the virtual queue, but it did not receive the event that indicates that the interaction exited the virtual queue, and URS has stopped sending status updates for that interaction.) **Note:** To calculate durations from virtual queue data accurately, Genesys recommends that rows that have this technical result and reason not be used.
	RoutedFrom AnotherVQ	The interaction was added to this virtual queue as well as to a parallel virtual queue. It was routed from the parallel virtual queue to the target destination, so that it was cleared from this virtual queue.
	DefaultRoutedBy Strategy	The interaction was routed by URS to the default destination, as defined by the URS configuration options.
	Targets Cleared	The interaction was cleared from the virtual queue by the URS strategy `ClearTarget` function.
	PulledBackTimeout	The routing strategy was unable to route the interaction successfully before the expiration of the `routing-timeout that was` configured in Interaction Server. As a result, the routing was considered to be a failure and the interaction was taken from the routing strategy and placed back into the interaction queue from which it came.
	Stopped	The interaction was stopped while it was being handled by the routing strategy. **Note:** An example of the use of this technical result is when the interaction was stopped by the Media Server while it was in the virtual queue. However, if the interaction was stopped because it was abandoned by the customer while the Media Server was running in compatibility mode, this technical result is not used. Instead, the `Abandoned` technical result is used.

Populating Outbound Campaign Activity

The Genesys Info Mart schema contains a number of subject areas related to outbound campaign activity (see Table 1 on page 13). This section provides information about the Contact_Attempt subject area, which is the area that is focused on actual outbound campaign interactions.

Genesys Info Mart creates contact attempt facts in order to represent the attempts to reach the customer records of a calling list during the course of an outbound campaign.

Populating Contact Attempt Facts and Dimensions

Genesys Info Mart populates contact attempt facts as follows:

- The two references to the DATE_TIME dimension, in addition to the start and end timestamps, represent the start and end time, respectively, of the outbound contact attempt.
- For more information about how Genesys Info Mart represents dates and times of day, see Chapter 6 on page 147.
- The CAMPAIGN dimension identifies the outbound campaign that launched the attempt.
- The TENANT dimension identifies the tenant of the campaign.
- The GROUP_ dimension identifies the campaign group (agent group or place group) that is assigned to this campaign.
- The CALLING_LIST dimension identifies the calling list that contains the target record of the attempt.
- The RECORD_TYPE dimension identifies the type of the target record—for example, General or CampaignRescheduled.
- The RECORD_STATUS dimension identifies the status of the target record at the end of the contact attempt—for example, Updated or Cancelled.
- The CONTACT_INFO_TYPE dimension identifies the type of contact information that is provided in the target calling list record—for example, HomePhone or Mobile.
- The CALL_RESULT dimension is used to identify the final call result of the contact attempt (for example, Answer, Busy, or Wrong Party) as well as the dialer result (for example, Answer or Busy) if a dialer was used.
- The RESOURCE_ dimension identifies the resource that is associated with the first IVR port or agent that corresponds to the outbound attempt, or an agent who is previewing this record.
- The RESOURCE_GROUP_COMBINATION_KEY dimension identifies the groups of which the Agent resource was a member when the contact attempt was started. This field references the default No Group value if the agent does not belong to a group.

- The PLACE dimension identifies the place that is associated with the first IVR port or agent that corresponds to the outbound attempt.
- The DIALING_MODE dimension identifies the dialing mode that was used for the contact attempt—for example, Predictive, Progressive, or Preview. For GVP, these dialing modes are PROGRESSIVE_GVP, PREDICTIVE_GVP, and POWER_GVP, respectively.
- The MEDIA_TYPE dimension identifies the media type of the interaction that is associated with the outbound attempt—for example, Voice.
- The RECORD_FIELD_GROUP_1 and RECORD_FIELD_GROUP_2 dimensions contain custom fields from the calling list record. The values represent a snapshot that was taken at the end of the contact attempt.
- Record field facts in the CONTACT_ATTEMPT_FACT table hold custom field values from the target calling list record. The values represent the snapshot that was taken at the end of the contact attempt.
- State counts and durations summarize the amount of time that is spent on various activities.

Note: The following columns in the CONTACT_ATTEMPT_FACT table are no longer populated, although they remain in the schema:

- IXN_START_TIME
- IXN_START_TIME_KEY
- CONTACT_IXN_START_TIME
- CONTACT_WITHIN_DAILY_RANGE

To obtain the same data, use the following calculations:

- For IXN_START_TIME and CONTACT_IXN_START_TIME, make a join between CONTACT_ATTEMPT_FACT and INTERACTION_FACT on CONTACT_ATTEMPT_FACT.CALLID=INTERACTION_FACT.MEDIA_SERVER_IXN_GUID.
- For IXN_START_TIME_KEY, use INTERACTION_FACT.START_DATE_TIME_KEY.
- For CONTACT_WITHIN_DAILY_RANGE, you must also take into account the contact TIME_ZONE, which is identified by the TIME_ZONE_KEY. For assistance with this calculation, which is situation- and RDBMS-dependent, contact Genesys Technical Support.

Outbound Campaign Activity Fact Tables

Genesys Info Mart stores facts about outbound campaigns and activity in the following tables:

- Contact attempts:
  - CONTACT_ATTEMPT_FACT

- Calling lists:
  - CALLING_LIST_METRIC_FACT
  - CALLING_LIST_TO_CAMP_FACT
- Campaigns and campaign groups:
  - CALLING_LIST_TO_CAMP_FACT
  - GROUP_TO_CAMPAIGN_FACT
  - CAMPAIGN_GROUP_SESSION_FACT
  - CAMPAIGN_GROUP_STATE_FACT

For detailed information about the columns in the Outbound campaign Fact tables, refer to the *Genesys Info Mart Reference Manual* for your RDBMS.

Populating Agent Activity Data

Genesys Agent activity data for both active and completed agent states is stored in the following tables:

- Summary tables for resource sessions, states, and reasons store summarized data, which is drawn from ICON, for all media types. The tables are the following:
  - SM_RES_SESSION_FACT
  - SM_RES_STATE_FACT
  - SM_RES_STATE_REASON_FACT

  Do-Not-Disturb status for each DN (or place and media type in the case of eServices/Multimedia) can optionally be factored into the SM_RES_STATE_FACT and SM_RES_STATE_REASON_FACT tables, configurable by switch. DND is treated as a NOT_READY state with the pre-defined software reason key DND On and no reason value. The termination of the DND state is treated as a READY state.

Note: You can choose to have summarized resource states and reasons for the NotReady and AfterCallWork states not be interrupted by incalls and outcalls.

Agents states are organized in a hierarchy, so that a higher-priority state takes precedence if multiple states happen simultaneously

The default priority list (in descending order) is AfterCallWork, NOT_READY, BUSY, READY. AfterCallWork has the highest priority, so all calls made during AfterCallWork are considered a part of the AfterCallWork.

You can change the hierarchy of states in this table by adjusting the settings for the sm-resource-state-priority configuration option in the gim-etl section of the Info Mart Application object.

Note: If you customize the priority table to set BUSY above AfterCallWork in the hierarchy, but ICON is configured not to interrupt AfterCallWork, Genesys Info Mart does not display the BUSY state when it happens during uninterrupted AfterCallWork.

Obtaining Uninterrupted Voice AfterCallWork and NotReady Data

Genesys Info Mart can represent voice AfterCallWork and Not_Ready states and reasons that are sourced from ICON and have them not be interrupted by incalls or outcalls that an agent makes while in these states.

These uninterrupted states and reasons are populated in the media-type-level, ICON-based Fact tables (SM_RES_STATE_FACT, SM_RES_STATE_REASON_FACT).

To obtain this data, set the gls-enable-acw-busy configuration option, which is located in the gts section on the Annex tab of the Switch configuration object, to 0 (the default setting is 1).

The various states are organized in a hierarchy and only one state is recorded for any time period. When multiple states occur at once, the higher-priority state takes precedence over any lower-priority state. The default priority list (in descending order) is AfterCallWork, NOT_READY, BUSY, READY.

Therefore, if an agent goes into the AfterCallWork state and then makes some calls, such calls made during AfterCallWork are considered a part of the AfterCallWork.

If the priority table is changed to have BUSY take first precedence, over AfterCallWork and NOT_READY, but ICON is configured not to interrupt AfterCallWork and NOT_READY states, the BUSY state is *not* recorded when it happens during uninterrupted AfterCallWork and NOT_READY states.

Populating Summarized Resource Sessions, States and Reasons

The SM_RES_SESSION_FACT, SM_RES_STATE_FACT, and SM_RES_STATE_REASON_FACT tables incorporate all data during the period in which an agent is logged on to a particular media type, regardless of the number of DNs or queues to which the agent logs on.

The SM_RES_SESSION_FACT Table

This table provides a summary of resource sessions by agent and media type.

Note: Genesys Interactive Insights reports require you to populate this table.

The Genesys Info Mart `populate-sm-resource-session-facts` configuration option, which is in the `gim-etl-populate` section, controls whether the `SM_RES_SESSION_FACT` table is populated. The `populate-sm-[media type]-resource-activity` configuration options control which media types this table is populated with if it is enabled.

Each row of this table summarizes the login session(s) of all DNs and places that are associated with an agent relative to a given media type. The grain of the fact is an accumulating snapshot that represents the duration of the summary session.

A summary session represents the contiguous duration that an agent resource is logged on for a given media type, irrespective of the number of DNs and/or queues to which the agent resource logs on.

- For voice, a summary session starts when an agent resource first logs on to any voice DN-queue combination. The session continues, irrespective of how many other voice DNs and/or queues the agent logs on to. The session ends when the agent resource logs off all voice DNs and queues.
- For multimedia, a session is first created when the agent resource adds a media type to their login session or logs onto a DN that supports this media. The login session continues until the agent resource removes the media type from the last login session that includes this media type, or logs out of the last DN that includes this media type.

Start and end dates and times are stored as facts in the UTC time zone. Start and end date and times are also stored as a dimension reference for `DATE_TIME`. Both active and completed sessions are populated.

Note: In some multimedia scenarios, an agent can process interactions for a particular media type without logging into the media (that is, without adding the media type to a place). In this scenario, Genesys Reporting does not see agent states related to the processing of interactions for the media type that are not added to the agent's place. Therefore, to ensure correct reporting, Genesys recommends that agents take care to add a media to a place before handling interactions of this media type.

The SM_RES_STATE_FACT Table

Each row of this table describes a summarized agent resource state relative to a given media type. The grain of the fact is an accumulating snapshot that represents the duration of the summarized state.

A *summary state* represents the contiguous duration that an agent resource is logged on with a particular state for a given media type, irrespective of the number of DNs, places, and/or queues to which the agent resource logs on. The summary state is chosen from among the concurrent states of all DNs to which the agent is logged on, based on the configured state priority list. For multimedia, there are no DNs, so that the summarized state represents the state of the agent relative to the media type.

Do-Not-Disturb can optionally be factored into resource states in this table. This functionality is configurable by switch.

Note: Genesys Interactive Insights reports require you to populate this table.

This table is sourced from Interaction Concentrator. The states that are recorded are the following:

- `Unknown` (the agent is logged on, but the agent state is unknown)
- `Busy`
- `Ready`
- `NotReady`
- `AfterCallWork` (voice media only)

Whether the `NotReady` or `AfterCallWork` (voice media only) states can be interrupted by interactions that the agent initiates or receives while in these states is dependent on the configuration of the underlying ICON application.The start and end dates and times are stored as facts in UTC time zone. The start date and time are also stored as dimension references for the `DATE_TIME` dimension.

Special Considerations For Very-Short Duration States

The `SM_RES_STATE_FACT` table has one-second data granularity.

When two or more states happen within the same second, only the highest-priority state is recorded for the duration of this second. As a result, some lower-priority states with durations of less than one second may disappear completely. For example, the following scenario is possible:

> An agent goes from the `NotReady` state to the `Ready` state and, being ready, receives a call within a fraction of a second. Then, agent goes into the `Busy` state. Since the `Ready` state, by default, has lower priority, it may disappear completely if its duration is less than one second. As a result, the `INTERACTION_RESOURCE_FACT` table may display the previous summarized state of the agent as `NotReady`, although it was actually `Ready`.

This is a limitation of the summarized state model. If necessary, you can obtain the exact agent state on a specific DN just before the agent takes an interaction from the GIDB tables.

Note: The timestamps for the start and end time in summarized tables may not match times in the ICON tables. `END_TS` in summarized tables means the beginning of the second by which the state has ended.

In addition it is important to look at the priorities of the previous and the following states. If multiple states occur within one second, the state with the highest priority state covers the entire one-second interval. A look at the states before and after the second in question my uncover a lower-priority state that also occurred during that second.

The SM_RES_STATE_REASON_FACT Table

Each row of this table describes a summarized agent resource state reason and workmode relative to a given media type. The grain of the fact is an accumulating snapshot that represents the duration of the summarized state reason.

Note: You must set the Interaction Concentrator configuration option `gls-active-reason-codes` (in the `callconcentrator` section) to the mandatory value of `TRUE`. This ensures that the `SM_RES_STATE_REASON_FACT` table is consistent in situations in which the reason code state ends after the transformation of the interval in which this reason code started. If this option is *not* set to `TRUE`, the Genesys Info Mart configuration checker will log the problem and prevent any jobs from starting.

A *summary state reason* represents the contiguous duration for which an agent resource is in some state with a particular state reason for a given media type, irrespective of the number of DNs and/or queues to which the agent resource logs on. A reason code state that is written into this table should have a highest priority among all concurrent agent states. This means the same state (without reason) will occur in the `SM_RES_STATE_FACT` table.

When multiple reason codes occur simultaneously for one agent, Genesys Info Mart chooses one of them to record in the `SM_RES_STATE_REASON_FACT` table based on the following considerations:

- A reason code state that starts later overrides an earlier reason code state.
- A software reason code takes priority over hardware.
- If the keys are different, the higher-value string takes priority.

- If the keys are the same, the key with the higher string value (not the higher numeric value) takes priority (using case-insensitive alphabetical comparison).
- The DND on reason takes the lowest priority with respect to other reason keys.
- Among two identical software reason codes with identical keys the priority is given to the state with the larger case-insensitive alphabetical reason code value.

Note: Reason code values are ranked alphabetically because ICON provides no data-type information to Genesys Info Mart that would identify whether the values are alphabetic, numerical, or mixed. As a result, some codes that occur in parallel may be ranked counterintuitively (5 > 45, for example).

When a reason-code state has a lower priority than some other concurrent agent state *without* a reason, this reason code state is not recorded in the SM_RES_STATE_REASON_FACT table.

Detailed information on all of the simultaneous reason codes is available in the GIDB_G_AGENT_STATE_RC_V, GIDB_G_AGENT_STATE_RC_MM, GIDB_G_AGENT_STATE_A_V, and GIDB_G_AGENT_STATE_A_MM tables. Note that the GIDB_G_AGENT_STATE_RC_V and GIDB_G_AGENT_STATE_RC_MM tables may contain multiple records for a single interaction, differing in their ending timestamp, if a reason-code state starts in one extract interval and ends in another extract interval.

Note: Do-Not-Disturb is optionally factored into summary state reasons with the pre-defined reason code key DND On and no reason value, based on the configuration of the underlying Switch object. All reasons that are associated with the current highest priority state of the agent are recorded. Genesys Interactive Insights reports require you to populate this table.

This table is sourced from IDB. The states for which reasons are recorded are the following:

- Ready
- NotReady
- AfterCallWork (voice media only)

Whether the NotReady or AfterCallWork (voice media only) state can be interrupted by interactions that the agent initiates or receives while in these states is dependent on the configuration of the underlying ICON application.

The start and end dates and times are stored as facts in the UTC time zone. The start date and time are also stored as dimension references for the DATE_TIME dimension.

How Summarized Data Is Processed

Summarized agent data, which must be recorded by a single ICON instance for a given agent, is processed in the following ways:

- Genesys Info Mart combines information for the same agent and media type from the ICON GX_SESSION_ENDPOINT table to form summarized media type sessions.
  - For voice, Genesys Info Mart combines information for the same agent and media type from the ICON G_AGENT_STATE_HISTORY, G_AGENT_STATE_RC, GIDB_G_AGENT_STATE_A, and G_DND_HISTORY tables to create summarized states and reasons that optionally have Do-Not-Disturb status factored into them. In addition, if the agent is logged on to more than one voice DN at a time, a configurable state priority list is used to determine which DN's state is considered to be the winning state.
  - For multimedia, Genesys Info Mart combines information for the same agent and media type from the ICON G_AGENT_STATE_HISTORY, G_AGENT_STATE_RC, GIDB_G_AGENT_STATE_A, G_DND_HISTORY, and GX_SESSION_ENDPOINT tables to form summarized states and reasons that optionally have Do-Not-Disturb status factored into them.

In the rare event that there is no call activity in the contact center, agent states are updated only after some delay. You can minimize this delay by setting appropriate values for two ICON configuration options:

- use-dss-monitor—When using Genesys Info Mart, this option must always be set to on.
- dss-no-data-tout—The valid values are from 60 to 86400 seconds. The default value is 300 seconds. As a result, by default there is a five-minute (300 second) delay before Info Mart sees that the agents have no interaction states, but you can reduce the delay to 60 seconds.

Note: A smaller value may adversely affect ICON performance.

Populating Do-Not-Disturb Data

Do-Not-Disturb data is optionally factored into states and reasons in the summarized SM_RES_STATE_FACT and SM_RES_STATE_REASON_FACT tables for all media types.

Including Do-Not-Disturb Data in Summary Tables

Inclusion of Do-Not-Disturb data in the summarized SM_RES_STATE_FACT and SM_RES_STATE_REASON_FACT tables is controlled by the factor-dnd-into-sm-resource-states configuration option, which is located in

the gim-etl section under the Annex tab of each switch. The default setting is TRUE for eServices/Multimedia switches and FALSE for voice switches.

For eServices/Multimedia, Do-Not-Disturb is treated as a global NotReady for all media types to which an agent is logged on at a given place.

DND states are treated as NotReady with a reason that indicates DND on. An example, how DND state is calculated for the default state priority list (AfterCallWork, NOT_READY, BUSY, READY, UNKNOWN) is explained in Table 5. This logic might be different for a user-configured state priority list.

Users can configure state priority in the sm-resource-state-priority option, in the gim-etl-populate section of the Genesys Info Mart Application object.

Table 5: Calculating DND Status

Conditions	Resulting DND Status
DND is turned *on* and the declared state is currently Ready.	The resource is considered to be in a NotReady state with a reason that indicates DND On.
DND is turned *off* and the declared state was previously Ready.	The resource returns to Ready with whatever reasons were originally attached to the Ready request.
DND is turned *on* and the declared state is currently AfterCallWork.	The resource stays in the AfterCallWork state. If AfterCallWork ends before DND is turned back off, the resource becomes NotReady, and the reason is DND On. If DND is turned on and off during AfterCallWork, the resource state is never shown as NotReady. **Note:** AfterCallWork applies only to non-multimedia media types.
The resource is in NotReady state and DND is turned *on* or *off*.	Any NotReady reasons that are currently in effect are not interrupted. If an existing NotReady state had no reasons, a new NotReady reason state with the key DND On is added.
The resource is in Busy state and DND is turned *on*.	When the Busy state ends, the resource enters the NotReady state with DND On as the reason.

High Availability

Deploying a high availability configuration can greatly reduce the chances of data quality issues. Genesys recommends using high availability throughout the data chain, from data sources through ICON and IDBs. Genesys Info Mart is designed to take advantage of high availability configurations in order to determine and draw on the most complete and reliable data available.

When you are using Interaction Concentrator in a high availability (HA) configuration, Genesys Info Mart must then select available data in such a way that it takes a complete and correct data set with no duplications. To accomplish this, ICON stores data-session information in each IDB that confirms whether the data for the time period that is to be extracted is complete and correct. Genesys Info Mart can then take data of whichever one of the redundant IDBs has the most complete and correct data.

Note: Refer to the Genesys Info Mart Release Note to determine the version number of Interaction Concentrator 8.0 that is required to use Genesys Info Mart HA functionality.

This approach to identifying the best data for any period eliminates the need for the resource-consuming double-extraction, analysis, and deduplication processes that are used in previous releases of Genesys Info Mart.

Criteria for Choice of Better IDB

When choosing which IDB in an HA pair (or set) from which to extract data, the data extract job examines all data connection reliability information in the available IDBs, so as to minimize switchovers from one IDB to another and to minimize data loss in cases in which there are gaps in source data in all available IDBs.

To avoid data loss, Genesys Info Mart does the following:

- Searches for an IDB that has no gaps in the data-session information within the next data chunk.
- If more than one IDB exists, Genesys Info Mart extracts data from the IDB that has a longer period without any interruption.
- If there are interruptions in all IDBs, data is extracted from that IDB that has the fewest gaps in the data.

There is a set of interrelated configuration options in Interaction Concentrator that enable Genesys Info Mart to have sufficient session information for data that is missing or incorrect. For details, see the *Interaction Concentrator 8.0 Deployment Guide*.

High-Water Marks

To track the data that is to be transformed, Genesys Info Mart 8.0 uses data-session information in the form of *high-water marks* that ICON writes in each IDB. The *high-water mark* is a timestamp that indicates the beginning

point for each extract slice, enabling interaction and configuration data from HA IDBs to be correlated.

Note: Genesys Info Mart requires ICON to write session information to the IDBs, whether or not your environment is HA. For information on how to verify that ICON is writing session information, see section on determining data availability and reliability in the *Interaction Concentrator 8.0 User's Guide.*

Storing Data from Multiple Sources in One IDB

Data from multiple data sources can be stored in the same IDB. However, the all data must be of the same type. For example, you could have an IDB that has data from more than one T-Server, but not data from both T-Server and Interaction Server.

Note: Do not store voice and OCS data in the same IDB.

Identifying HA Sets of Data Sources

For Genesys Info Mart to determine which IDBs contain data from an HA set of data sources, it looks to the database access points (DAPs) that are specified in the `Connections` tab of the Genesys Info Mart `Application` object. These DAPs point to the corresponding IDBs, each of which identifies its data source (such as T-Server, Outbound Contact Server (OCS), or Interaction Server) in the `G_DSS_*_PROVIDER` table.

Although Genesys Info Mart does not communicate directly with ICON, you must add all instances of ICON that supply data to Genesys Info Mart in the `Connections` tab of the Genesys Info Mart `Application` object. This ensures that, even if an IDB or a data source is unavailable, Genesys Info Mart can still determine how many data sources there should be and whether the available IDBs contain a complete set of data for the time period in question.

To ensure that HA works correctly, you must store data from each data source (T-Server, OCS, or Interaction Server) in more than one IDB, configure DAPs for those IDBs, and add the DAPs to the `Connections` tab in the Genesys Info Mart `Application` object. Genesys Info Mart does not perform cross-checks to enforce the correct configuration.

Note: If HA is also configured at the data-source level (for example, for OCS), the HA ICONs should still be listening in to each server (primary and backup OCS). Otherwise, with both ICONs monitoring the same data source, if the data source goes down, the data in both ICONs will be compromised.

Preventing Data Quality Issues when Restarting ICON

A high availability configuration can eliminate potential multimedia data-quality issues that might arise as a result of setting the `calls-in-the-past` ICON configuration option to the required value of `true`.

If you need for any reason to restart a multimedia ICON, to install an upgrade, for example, and you do *not* have a high availability configuration, information

about previous parties and first values of user data keys might be missing or inaccurate. Use of a high availability configuration eliminates these issues and is recommended.

For additional information about potential data-quality issues for multimedia ICON, see the "Special Considerations When Restarting Multimedia ICON" section in the "Understanding Genesys Info Mart Jobs" chapter of the *Genesys Info Mart 8.0 Operations Guide*.

Handling Partially-Merged Calls

In multi-site scenarios, Genesys Info Mart must merge data from the multiple sites. If an interaction moves from site to site during the handling process, Genesys Info Mart uses linkage data to integrate the data from various T-Servers into a single interaction. This can be disrupted for a number of reasons, causing data quality issues.

There are three major cases when data from a site might be unavailable:

- The site is not monitored (partially-monitored environment).
- The site is monitored, but information is missing.
- Information is delayed.

Data Issues in a Partially-Monitored Environment

If you configure Genesys Info Mart to extract voice interaction data from topologies in which not all T-Servers or IVR Servers involved in the call flow are monitored by Interaction Concentrator, data inconsistencies can occur, such as incomplete and missing data.

Note: If you have an environment that includes unmonitored sites, such sites must be noted in the `GSYS_DNPREMOTELOCATION` table.

A partially-monitored environment can result in missing data at the start, middle, or end of an interaction. The following interaction scenarios can affect the population of interaction data within Genesys Info Mart, resulting in data inconsistencies:

- The interaction originates in an unmonitored T-Server.
- The interaction terminates in an unmonitored T-Server.
- The interaction originates in a monitored T-Server, moves to an unmonitored T-Server, and then passes on to a monitored T-Server.

Note: Any combination of the previously-mentioned deployments will have an effect on the population of voice interaction data.

Each time that the interaction moves from an unmonitored to a monitored T-Server, it appears to be a new interaction. For example, a single interaction might start on a monitored T-Server, be sent to an unmonitored T-Server, and then be sent to a monitored T-Server. This single interaction is represented in Genesys Info Mart as two interactions. When this type of interaction scenario occurs, the linkage information that ties an interaction together as it moves from T-Server to T-Server is incomplete and Interaction Concentrator cannot associate what it sees as multiple calls into a single interaction.

A partially-monitored deployment can result in data that is incorrect or missing from the following Fact tables:

- Populating Interaction Fact
  - Some interaction facts will be missing where entire calls or parties are missing in the source data. Tables that are affected by this can include:
    - Interaction Resource Fact
    - Interaction Resource State Fact
    - Mediation Segment Fact
- Populating Interaction Resource Fact—Some interaction resource facts might not reflect accurate information about `MEDIATION RESOURCES`, `CONSULT`, `CONFERENCE`, and `TRANSFER` metrics, `TECHNICAL DESCRIPTOR`, `ROUTING TARGET` and `SERVICE LEVEL` flags, such as `MET SERVICE OBJECTIVE` and `SHORT ABANDONED`. This occurs because these fields are highly dependent on other resources that are involved in the interaction which might or might not be monitored.
- Populating Interaction Resource State Fact—Some interaction resource state facts might not contain accurate information in the `STATE DESCRIPTOR` and `ROLE VALUE` for the states that are generated for interaction resource facts. This occurs because these values are populated based on interaction-type information that might change as a result of an unmonitored T-Server in the environment.
- Populating Mediation Segment Fact—Some mediation segment facts might be missing or have incorrect `TECHNICAL DESCRIPTOR` values because the ETL cannot determine why the interaction was placed in the Queue or virtual queue, or whether it was answered or abandoned after it was distributed from the queue or virtual queue.

Late Data

Late data can be a result of various issues, such as intermittent connectivity issues for an IDB. For example, you might currently have only data for the beginning of an interaction, but data from a second T-Server is anticipated to arrive ‘soon’. In this case, *soon* means that data should arrive during the timeout set in the Genesys Info Mart `extract-data-stuck-threshold` configuration option.

- `extract-data-stuck-threshold` timeout—This setting specifies the allowable delay to wait for the missing data to become available, at which point the interaction is processed normally. If the threshold expires before the data arrives, the data is treated as missing.

Note: For additional details on the `extract-data-stuck-threshold` timeout, see the *Genesys Info Mart 8.0 Deployment Guide*.

Error Handling in Case of Missing Data

For various reasons, information from a data source for a specific time range might be missing from a monitored site. In a high-availability (HA) environment, a single failure does not result in loss of data. If, in exceptional circumstances, multiple failures occur, the HA environment becomes, in effect, a non-HA environment.

Note: The following discussion of error handling assumes a non-HA environment.

There are a number of points at which data transfer and collection can be interrupted. Figure 3 on page 58 shows three failure points at which communication can be interrupted. Each point is discussed in the correspondingly-numbered paragraph that follows Figure 3.

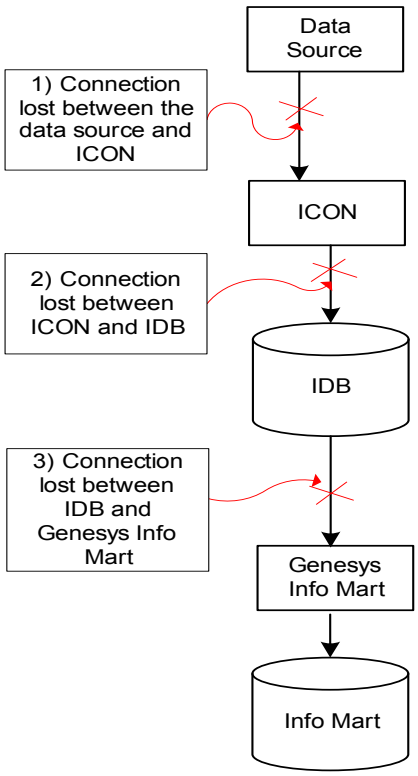

Figure 3: Possible Failure Points

1) Connection failure between ICON and the data source.

If this connection fails, data transmission is interrupted. When the connection is restored, ICON writes whatever data the data source can provide to IDB, from which Genesys Info Mart then extracts it.

When Genesys Info Mart's configuration checker runs, it identifies unavailable data sources and halts any extract job until the data source is again available. To have Genesys Info Mart extract from available data sources only, you can disable the unconnected data source by clearing the State Enabled check box in the data source's `Application` object. As a result, the data that Genesys Info Mart extracts from IDB might be delayed on account of the upstream connection failure.

Recommendation: To prevent the loss of data from a connection failure between the data source and ICON, set up your environment with HA at the data-source level.

In addition, Genesys recommends that you configure an alert for the log message (number 20110) that is generated when data from a data source does not arrive within the `extract-data-stuck-threshold` timeout period. If an alert occurs, you can then you can then investigate promptly and take appropriate action. For example, you might decide to temporarily disable the data source `Application` object in the Configuration Layer, which enables Genesys Info Mart to proceed with transformation of the remaining data.

2) Communication failure between ICON and IDB.

When the interruption to communication occurs between an ICON and the IDB it populates, Genesys Info Mart continues extracting data from any other available IDBs for the same data domain. The result depends on two factors:

- Whether the data interruption lasts so long that the extract window passes entirely beyond the point at which the data interruption occurred;

and

- Whether your Genesys Info Mart settings can accommodate backlog extraction.

If the failure lasts long enough, the extract window might pass the point at which the interruption occurred, as shown in Figure 4 on page 59.

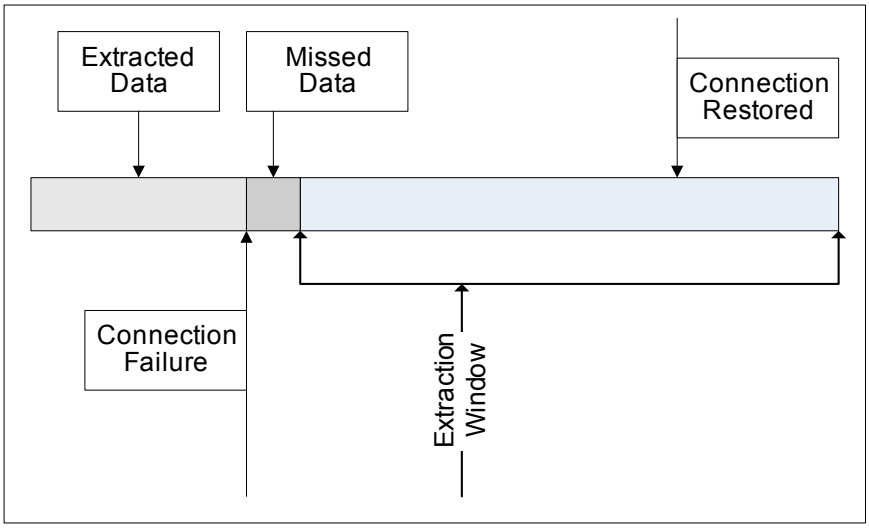

Figure 4: Data Extract Window

By adjusting extract chunk size or even halting purging, you can ensure that backlogged data is correctly extracted and transformed. The outcome is delay, but no data loss.

If the configuration options that control extract chunk size and ETL scheduling do not accommodate efficient processing of a backlog of ICON data, you might fail to extract some ICON data if the communication failure lasts long enough for the extract window to pass entirely beyond the time at which the communication failure initially occurred. Alternatively, if the configuration options that control data retention in GIDB do not allow for a backlog of unprocessed data, you might fail to transform some extracted data before it is purged.

Recommendation: To prevent the loss of data from a connection failure between ICON and IDB, set up your environment with HA at the ICON level.

3) Communication failure between Genesys Info Mart and IDB.

There are a number of outcomes in this scenario, depending on the nature of the connection interruption.

- Genesys Info Mart might not be able to access the data in IDB and therefore cannot read the session information from IDB. In this case, the extract fails.
- Genesys Info Mart tries to match the list of configured data sources against the list of data sources for the available IDB(s). If it finds no mismatch, Genesys Info Mart:
  - Logs an event that indicates the DAP connection that is missing.
  - Proceeds with extract.

Note: If Genesys Info Mart has no connection to the Info Mart database, it will not start the extract job at all.

Configuration Options for Missing-Data Behavior

For transformation, two configuration options control the handling of missing data: `error-policy-islink-dangling` and `error-policy-irf-exception`. The first option enables you to determine whether missing data is handled as an exception. If it is, the second option enables you to specify how such an exception is handled.

Loss of data from one data source in the available (accessible) IDBs can be handled as critical or noncritical, depending on the configuration of the `error-policy-irf-exception` and `error-policy-islink-dangling` options.

Note: For additional information on these options, see the description of the options in the *Genesys Info Mart 8.0 Deployment Guide* and the discussion of high availability data handling in the *Genesys Info Mart 8.0 Operations Guide*.

Missing Configuration Objects

Genesys Info Mart checks the list of known configuration objects:

- During data transformation.

If, during transformation of configuration facts data, `Job_TransformGIM` notes a missing configuration object, it records the information in the `STG_IDB_FK_VIOLATION` table.

During transformation of data types other than configuration data, Genesys Info Mart treats such missing configuration objects as late-arriving and creates placeholders for the missing objects based on the configuration object type and

its unique ID. When the missing configuration objects arrive from Configuration Server, these placeholders are populated with missing data.

Note: The unique configuration ID also prevents accidental duplication of configuration objects.

Data Maintenance and Purging

Purging enables you to remove data that is no longer necessary, such as data from interactions that were completed long ago, from the Global Interaction Database (GIDB), the Staging area, and the Genesys Info Mart Fact tables. A number of configuration options enable you to configure how long to retain data and what data to purge.

Note: For information about the configuration options that are used for purging, see the *Genesys Info Mart 8.0 Deployment Guide*.

Purging in GIDB and the dimensional model consists of removing:

- Completed and active fact data from GIDB.
- Completed and active fact data from the dimensional model, including certain Staging tables.
- Discarded operational data from discard tables.
- Outdated information from the `AUDIT_LOG` and History tables.
- For partitioned tables, partitions having only old data that is eligible for deletion.

Purging of the staging area includes both user-defined criteria for removing discarded data and automatic clearing out of internal, temporary data.

Data Retention Thresholds

GIDB and dimensional model purging is controlled by *data retention thresholds,* which define the retention windows; that is, the duration of data to keep in the Genesys Info Mart database. The purging algorithms divide dimensional model and GIDB Fact tables into two categories:

- Short-lived, such as voice calls or agent logins.
- Long lived, such as e-mail interactions.

The options that define these categories and how purging is handled are described in detail in the "Purging the Info Mart Database" section of the *Genesys Info Mart 8.0 Operations Guide*.

When you are configuring the data retention threshold for GIDB, to ensure that you allow enough time for GIDB data to be transformed, consider the following recommendations:

- The data retention threshold should be greater than the data stuck threshold.

- The data retention threshold should be greater than the merge procedure timeout.

When you are configuring the data retention threshold for the dimensional model, ensure that you allow enough time for dimensional model data to be aggregated and used in reports.

As a general rule during configuration of the data retention thresholds, allow ample time to ensure that data is not accidentally purged before it has been fully processed.

Note: For a detailed discussion of purging, including specifics of how purging is handled in various table types, see the *Genesys Info Mart 8.0 Operations Guide*.

DATE_TIME Table Maintenance

Two configuration options, `day-time-min-days-ahead` and `day-time-max-days-ahead`, enable you to specify for what period `Job_InitializeGIM` and `Job_MaintainGIM` will add records to the `DATE_TIME` table.

- Data is added if the last record that is present in the table is earlier than `current-date + day-time-min-days-ahead`.
- Records are added until `current-date + day-time-max-days-ahead`.

Note: For a more detailed discussion of the `DATE_TIME` table, see Chapter 6 on page 147.

Chapter

3 Validated Voice Interaction Flows

This chapter describes the recognized, validated voice interactions that have been tested and that are supported by Genesys Info Mart. The validated interactions are premise-based flows that involve one or more of the deployed Genesys solutions.

The call flows that are described in this chapter are intended as examples that you can modify for your environment. However, Genesys does not guarantee results for modified interaction flows.

This chapter contains the following sections:

Note: Voice interactions that are generated by other supported Genesys solutions might yield call flows in Genesys Info Mart that do not directly translate to the call flows that are described in this chapter. Voice interactions that involve Genesys solutions and are not supported by Genesys Info Mart might yield unpredictable results.

Overview

The validated call flows that are described in this chapter are organized according to the types of solution that might be deployed in your contact center. Table 6 provides an overview of the validated call flows.

Table 6: Validated Call Flows

Solution	Description
Framework only	Based on the dialed number, voice interactions that arrive at the switch are queued to an ACD queue that represents a requested skill, service type, or customer segment. Agents who are logged into the ACD queues handle the interactions. **Note:** Flows that start in a diagram under one of the other solutions can resume in another diagram under this solution (for example, if a voice interaction in Universal Routing is routed to an agent, and the agent performs a two-step transfer to another agent).
IVR in front of switch	Voice interactions arrive at an IVR that is visible to the IVR Server's virtual T-Server. The focus of the IVR application can be either self-service or simple front-end identification and segmentation. If the IVR application cannot completely handle the voice interaction, the interaction can be transferred to an ACD queue behind the switch that represents a requested skill, service type, or customer segment. Agents logged in to the ACD queues handle the interactions.
IVR behind switch	Voice interactions that arrive at the switch are queued to an ACD queue, where the ACD positions are actually IVR ports. The focus of the IVR application can be either self-service or simple front-end identification and segmentation. If the IVR application cannot completely handle the voice interaction, the interaction can be transferred to an ACD queue that represents a requested skill, service type, or customer segment. Agents who are logged in to the ACD queues handle the interactions.
Universal Routing	Voice interactions that arrive at the switch are delivered to a Routing Point. Universal Routing Server (URS) uses criteria such as ANI, DNIS, and the date and time of day to collect information and select an appropriate routing target. Basic targets are ACD queues and individual DNs; more advanced targets are agent groups, place groups, and skill expressions.

Table 6: Validated Call Flows (Continued)

Solution	Description
Universal Routing assisted by IVR behind switch	Voice interactions that arrive at the switch are queued to an ACD queue, where the ACD positions are actually IVR ports. The IVR application collects digits and information about the caller, and transfers the call to a Routing Point. Universal Routing uses the collected information to select an appropriate routing target. Basic targets are ACD queues and individual DNs. More advanced targets are agent groups, place groups, and skill expressions.
IVR in front of switch assisted by Universal Routing	Voice interactions arrive at an IVR that is visible to the IVR Server's virtual T-Server. Through a Routing Point in the IVR Server's virtual T-Server, the IVR application invokes a Universal Routing strategy. Universal Routing instructs the IVR application to play applications or collect information. Universal Routing uses the collected information to return an appropriate target. The IVR application hook-flash transfers the call to that target.
IVR behind switch assisted by Universal Routing	Voice interactions that arrive at the switch are queued to an ACD queue, where the ACD positions are actually IVR ports. Through a virtual routing point in the premise T-Server, the IVR application invokes a Universal Routing strategy. Universal Routing instructs the IVR application to play applications or collect information. Universal Routing uses the collected information to return an appropriate target. The IVR application mute transfers the call to that target.

Diagram Conventions

The call flow diagrams in this chapter use the following conventions:

- Dotted shading indicates customer wait time.
- Diagonal shading indicates customer handle time.
- The following abbreviations are used for simplicity:
  - IRF—Interaction Resource Fact
  - SS IVR—Self-service IVR (considered to be a *handling resource* or *resource of interest* with regard to IRF data collection)
  - nonSS IVR—Nonself-service IVR (considered *not* to be a *handling resource* or *resource of interest* with regard to IRF data collection)

Figure 5 on page 66 shows a legend for the call flow diagrams.

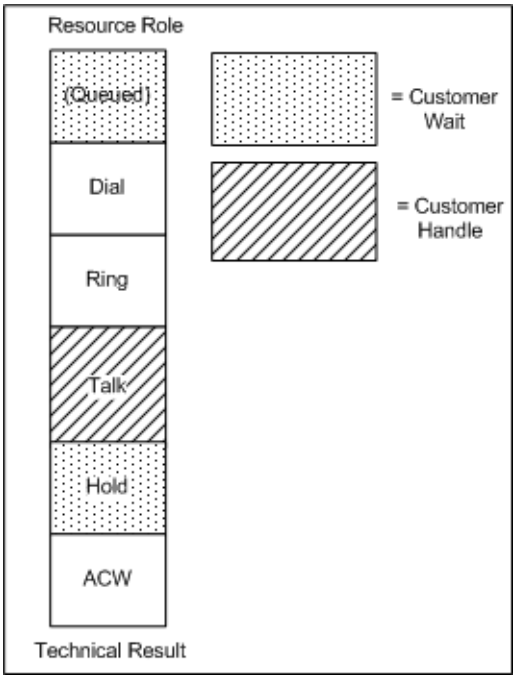

Figure 5: Call Flow Legend

To show the voice interaction flow, the diagrams in this chapter depict the media-specific states in sequence.

Figure 6 shows an example of an IRF call flow.

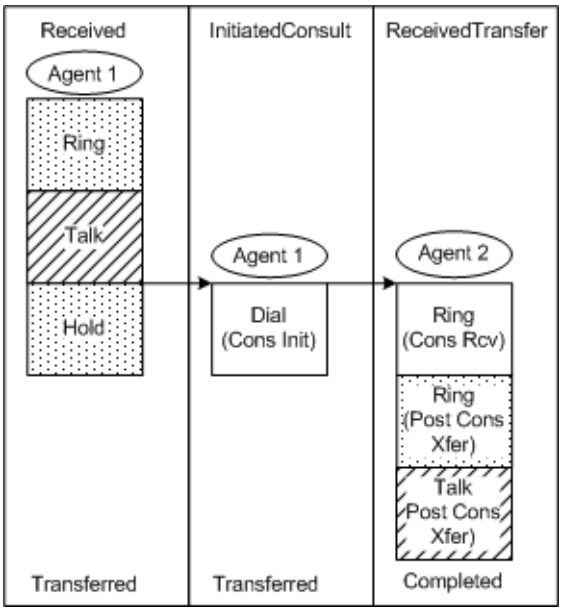

Figure 6: Sample Call Flow

Notes on the Interaction Resource Fact Diagrams

In Interaction Resource Fact (IRF) diagrams, call flows represent the resources that handle the interaction and their states.

The following list points out features that are specific to the diagrams:

- The circled resource in the diagram represents the resource.
- The rows are separated by vertical lines.
- The resources of interest are *handling resources*, which are those that have the greatest interest for reporting—agents, self-service IVRs, and DNs without an agent. *Nonhandling resources* include mediation resources such as queues, routing points, and nonself-service IVRs.
- The diagrams also show with which portion of the call each resource's state is associated (such as received consult, post-consult transfer, and post-consult conference).

Framework-Only Call Flows

Based on the dialed number, voice interactions that arrive at the switch are queued to an ACD queue that represents a requested skill, service type, or customer segment. Agents who are logged in to the ACD queues handle the interactions.

Note: Flows that start under a diagram in one of the other solutions might resume in another diagram under this solution (such as when a voice interaction in Universal Routing routes to an agent, and the agent performs a two-step transfer to another agent).

This section describes call flows for the following types of interactions:

- Inbound call flow examples
- Outbound call flow example (see page 82)
- Internal call flow examples (see page 82)

Inbound Call Flow Examples

This subsection contains several examples of inbound call flows. Each example represents a different outcome:

- An ACD queue directs the inbound call to an agent.
- The inbound call is answered directly by an agent (see page 69).
- An agent mute transfers the call to an ACD queue (see page 69).
- An agent mute transfers the call to another agent (see page 73).

- An agent consults to an ACD queue and then retrieves the call (see page 73).
- An agent consults to another agent and then retrieves the call (see page 77).
- An agent consults to an ACD queue and then transfers the call (see page 78).
- An agent consults to another agent and then transfers the call (see page 79).
- An agent consults to an ACD queue and then conferences the call (see page 80).
- An agent consults to another agent and then conferences the call (see page 81).

Inbound to Agent via ACD Queue

In this call topology, an inbound call is delivered to an agent via an ACD queue. The interaction arrives at the ACD queue, and the ACD queue diverts it to an agent.

- Figure 7 depicts the call topology.

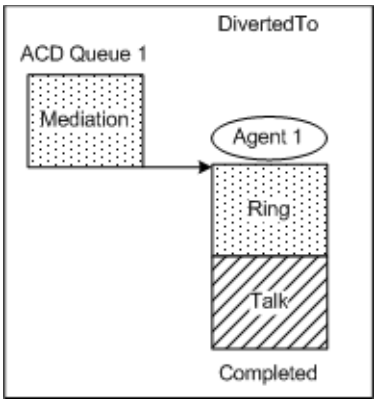

Figure 7: Inbound to Agent via ACD Queue

Inbound to Agent Directly

In this call topology, an inbound call is answered directly by an agent.

- Figure 8 depicts the call topology.

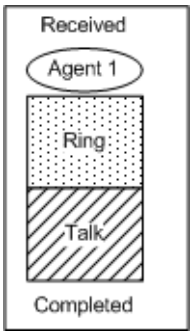

Figure 8: Inbound to Agent Directly

Mute Transfer to ACD Queue

In this call topology, an inbound call arrives at the ACD queue and is diverted to an agent. The agent then mute transfers the call to another ACD queue.

This section shows three possible outcomes of a call that is mute transferred to an ACD queue:

- The call is abandoned while it is in the second ACD queue (see "Mute Transfer to ACD Queue—Abandoned in Queue" on page 70).
- The call is abandoned while it is ringing at the second agent (see "Mute Transfer to ACD Queue—Abandoned While Ringing" on page 71).
- The call is successfully transferred to the second agent (see "Mute Transfer to ACD Queue—Completed" on page 72).

Mute Transfer to ACD Queue—Abandoned in Queue

For this outcome, the call is abandoned while it is in the second ACD queue.

- Figure 9 depicts the call topology.

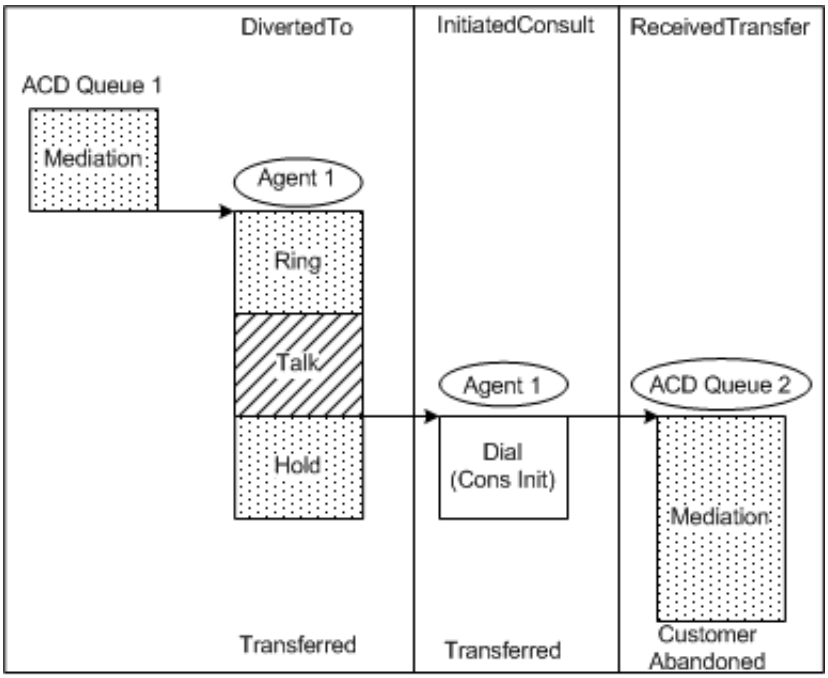

Figure 9: Transfer Abandoned While in ACD Queue

Mute Transfer to ACD Queue—Abandoned While Ringing

For this outcome, the call is diverted to the second agent, but it is abandoned while ringing.

- Figure 10 depicts the call topology.

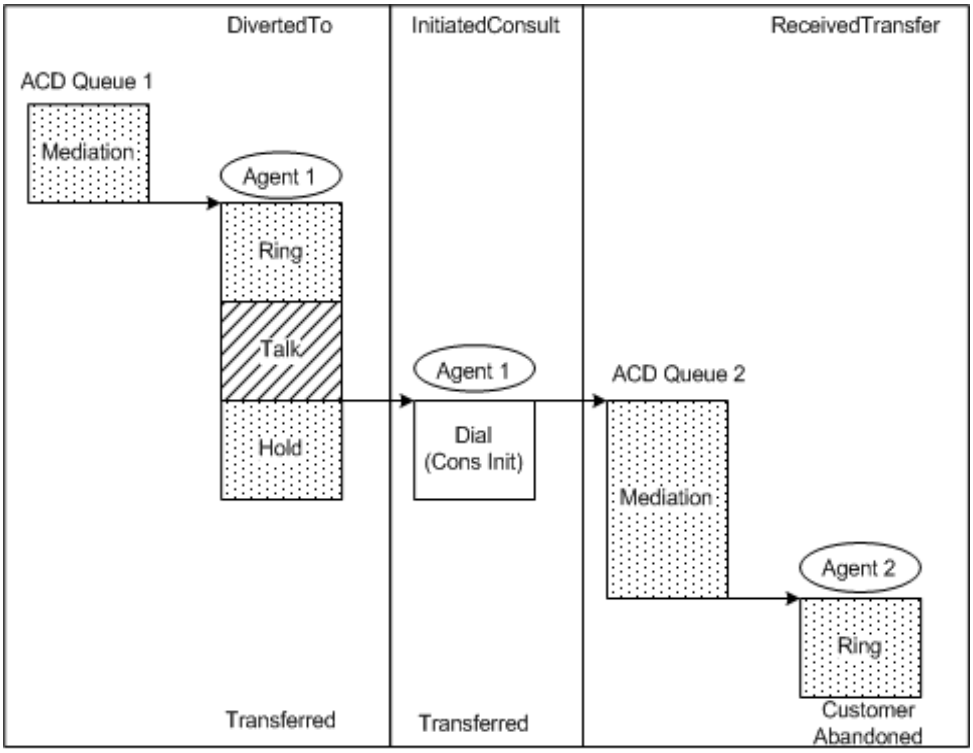

Figure 10: Transfer Abandoned While Ringing at Agent

Mute Transfer to ACD Queue—Completed

For this outcome, the call is successfully diverted to another agent.

- Figure 11 depicts the call topology.

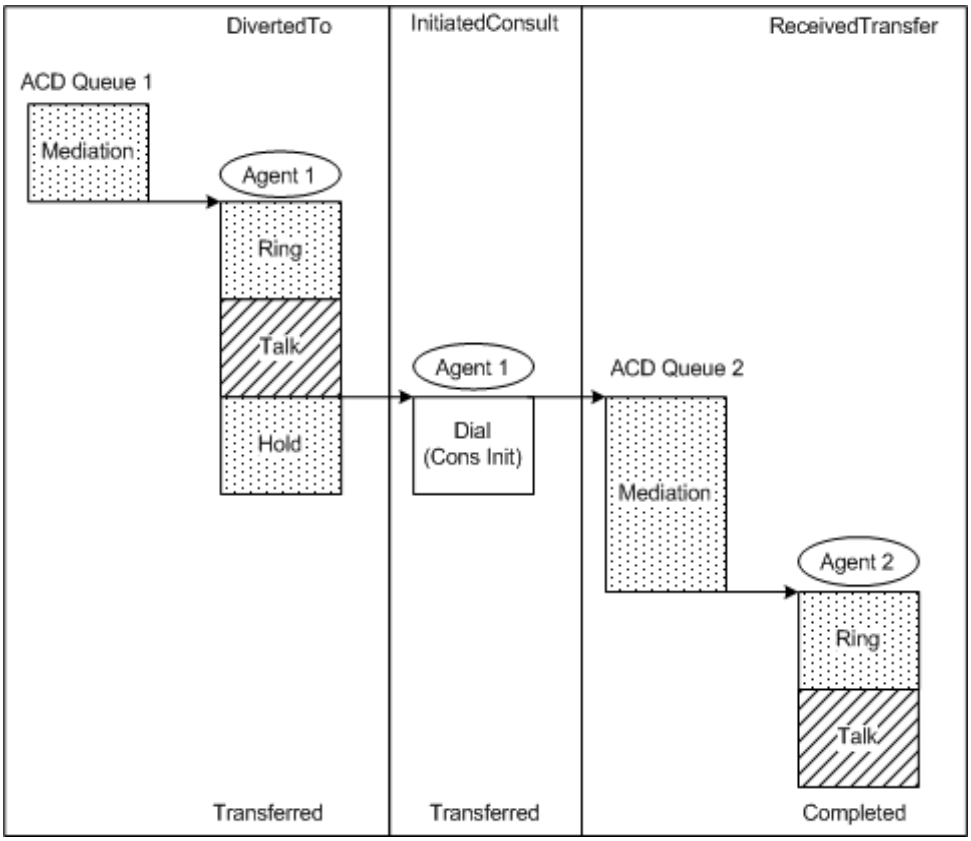

Figure 11: Transfer Completed

Mute Transfer to Agent

This call topology shows the outcome of a call that arrives at an agent, who answers the call and then mute transfers it to another agent.

- Figure 12 depicts the call topology.

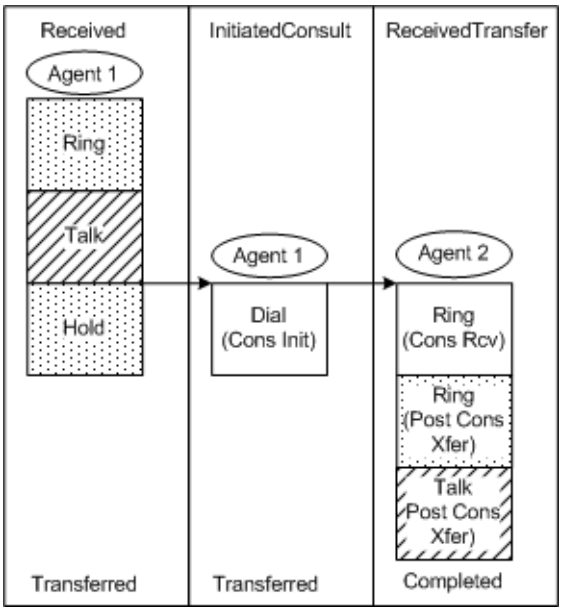

Figure 12: Agent Mute Transfers to Another Agent

Consult to Agent via ACD Queue, and Then Retrieve

In this call topology, an inbound call arrives at the ACD queue and is diverted to an agent. The agent consults to another ACD queue, and the call is diverted to another agent. The consultation ends when the first agent retrieves the call.

This section shows three possible outcomes of a call that is retrieved after a consultation has been initiated:

- The call is retrieved while it is in the second queue (see "Consult to ACD Queue—Abandoned in Queue").
- The call is retrieved while it is ringing at the second agent (see "Consult to ACD Queue—Abandoned While Ringing" on page 75).
- The call is retrieved after the consultation is completed (see "Consult to ACD Queue—Completed" on page 76).

Consult to ACD Queue—Abandoned in Queue

For this outcome, the call is retrieved while it is in the second ACD queue.

- Figure 13 depicts the call topology. Note that from the IRF perspective, the call is abandoned from the queue because no new handling resource receives the call from the queue.

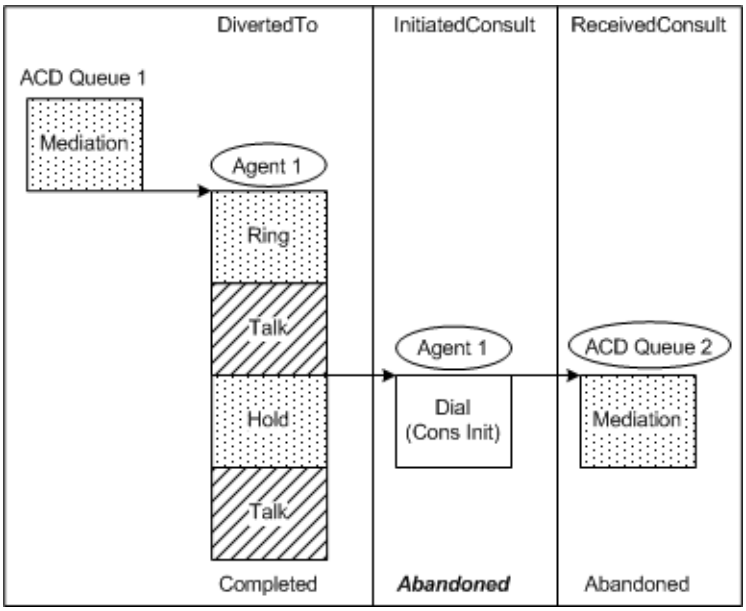

Figure 13: Call Retrieved While in ACD Queue

Consult to ACD Queue—Abandoned While Ringing

For this outcome, the call is retrieved while it is ringing at the second agent.

- Figure 14 depicts the call topology. Note that from the IRF perspective, the call is abandoned from the queue because the new handling resource, `Agent 2`, never receives the call.

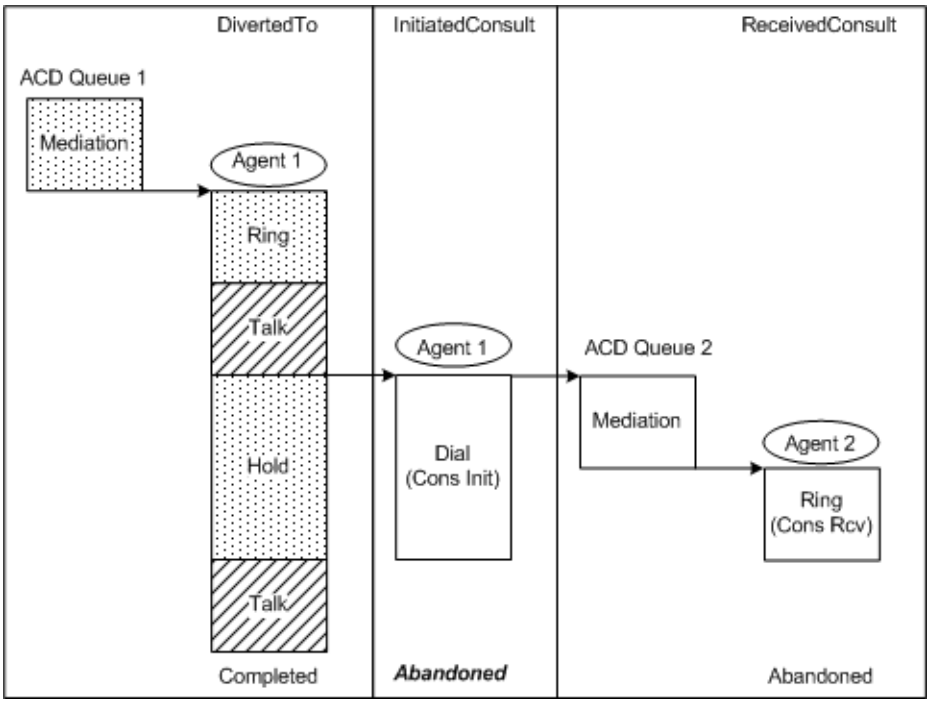

Figure 14: Call Retrieved While Ringing at Agent

Consult to ACD Queue—Completed

For this outcome, the call is retrieved after the consultation is completed.

- Figure 15 depicts the call topology.

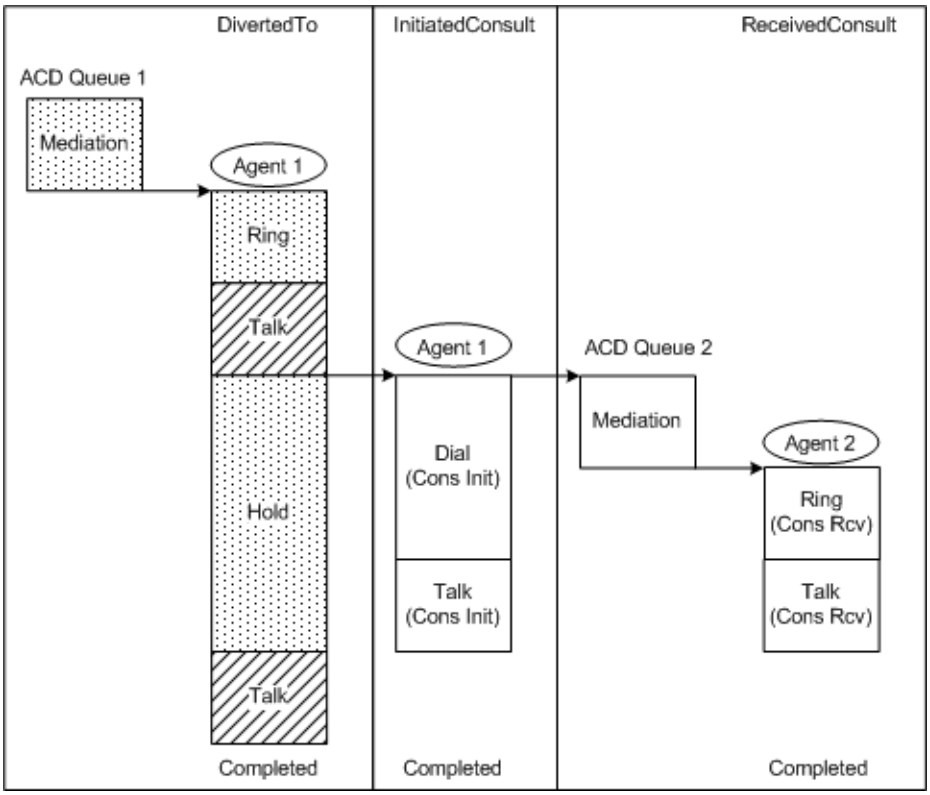

Figure 15: Consultation Completed

Consult to Agent, and Then Retrieve

This call topology shows the outcome of a call that arrives at an agent, who consults to another agent. The consultation ends when the first agent retrieves the call.

- Figure 16 depicts the call topology.

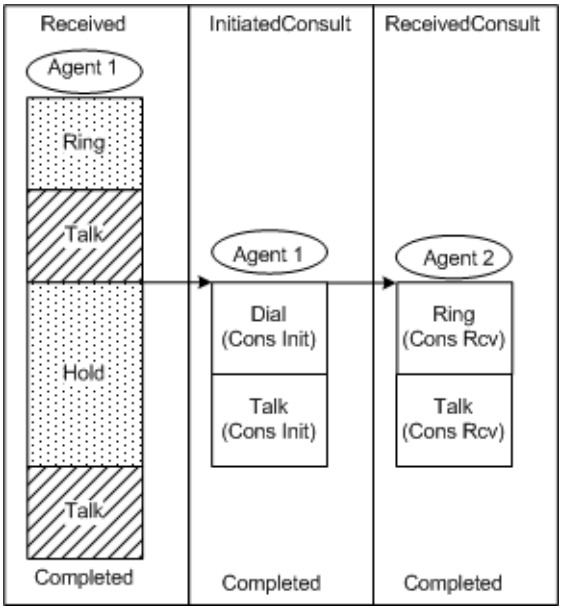

Figure 16: Consult to Agent, and Then Retrieve

Consult to Agent via ACD Queue, and Then Transfer

This call topology shows the outcome of a call that is transferred after a consultation. The call arrives at the ACD queue and is diverted to an agent. The agent consults to another ACD queue, and the call is diverted to another agent. The consultation ends when the first agent transfers the call.

- Figure 17 depicts the call topology.

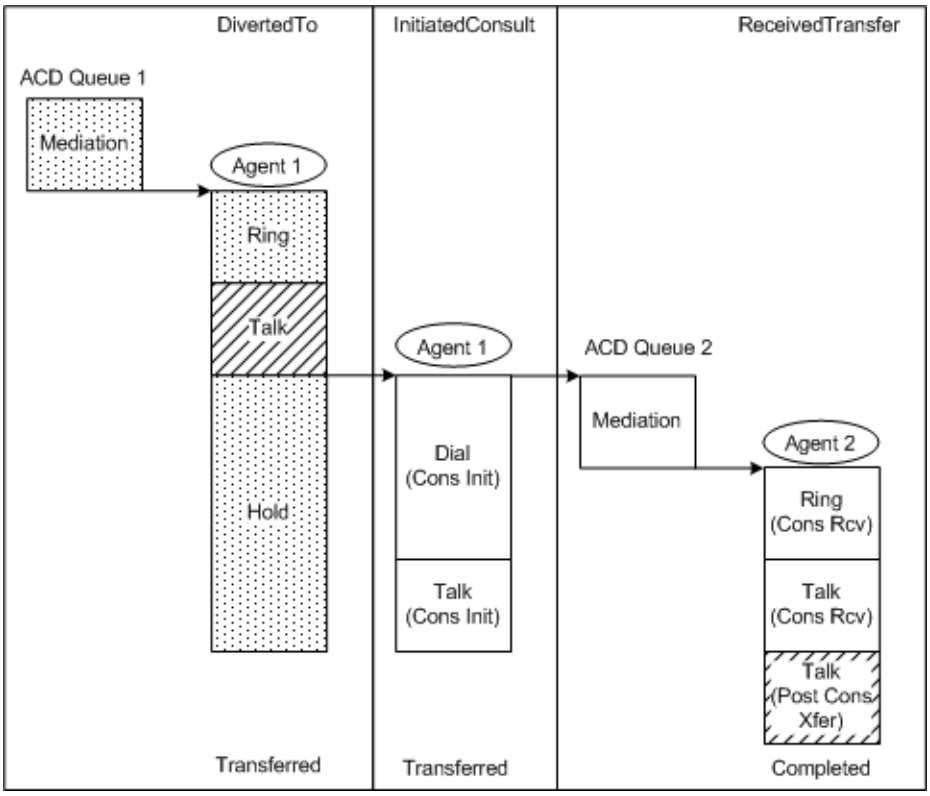

Figure 17: Consult to Agent via ACD Queue, and Then Transfer

Consult to Agent Directly, and Then Transfer

This call topology shows the outcome of a call that is transferred after a consultation. The call arrives at an agent, who consults to another agent, and then transfers the call. The consultation ends when the first agent transfers the call.

- Figure 18 depicts the Interaction Resource Fact (IRF) representation of the call topology.

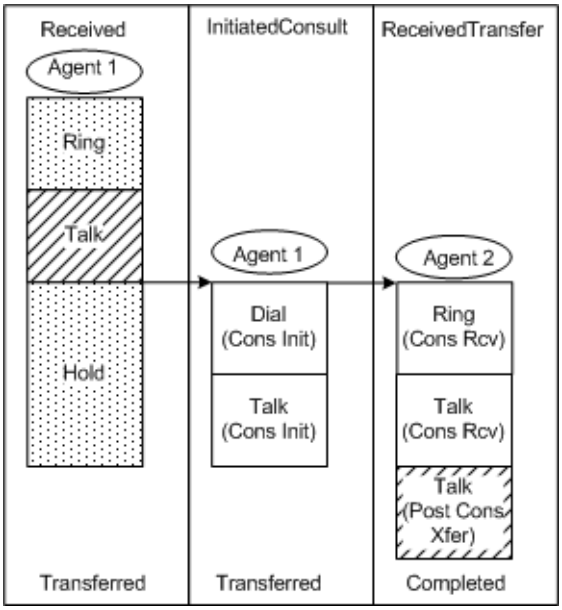

Figure 18: Consult to Agent Directly, and Then Transfer

Consult to Agent via ACD Queue, and Then Conference

This call topology shows the outcome of a call that is conferenced after a consultation. The call arrives at the ACD queue and is diverted to an agent. The agent consults to another ACD queue, and the call is diverted to another agent. The consultation ends when the first agent conferences the call.

- Figure 19 depicts the Interaction Resource Fact (IRF) representation of the call topology.

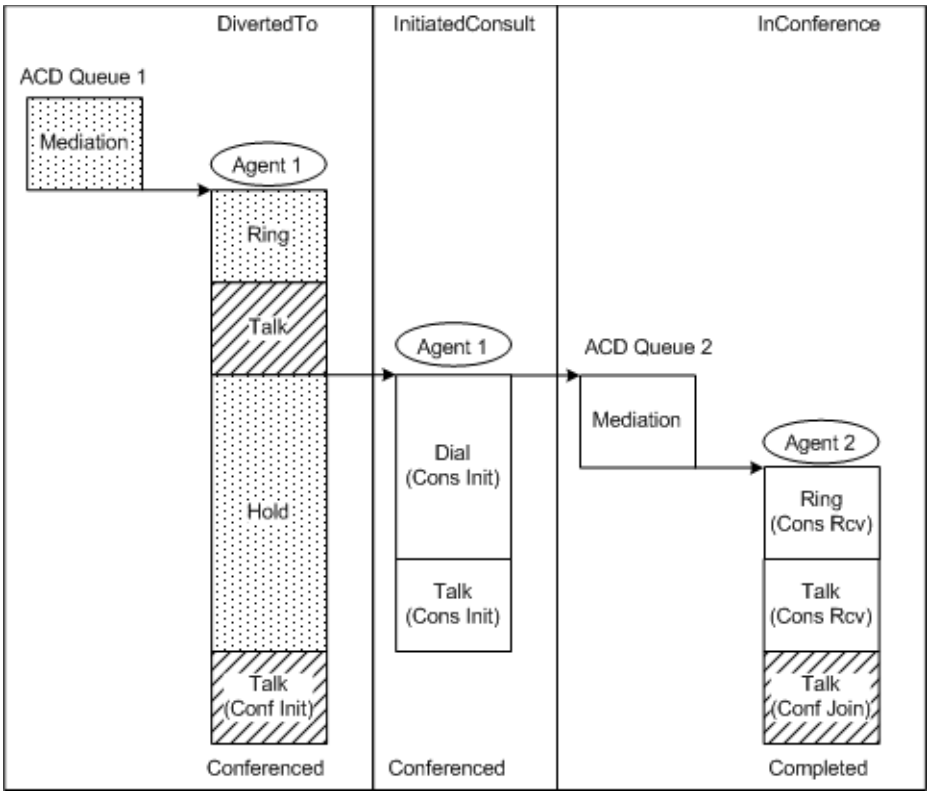

Figure 19: Consult to Agent via ACD Queue, and Then Conference

Consult to Agent Directly, and Then Conference

This call topology shows the outcome of a call that is conferenced after a consultation. The call arrives at an agent, who consults to another agent. The consultation ends when the first agent conferences the call.

- Figure 20 depicts the Interaction Resource Fact (IRF) representation of the call topology.

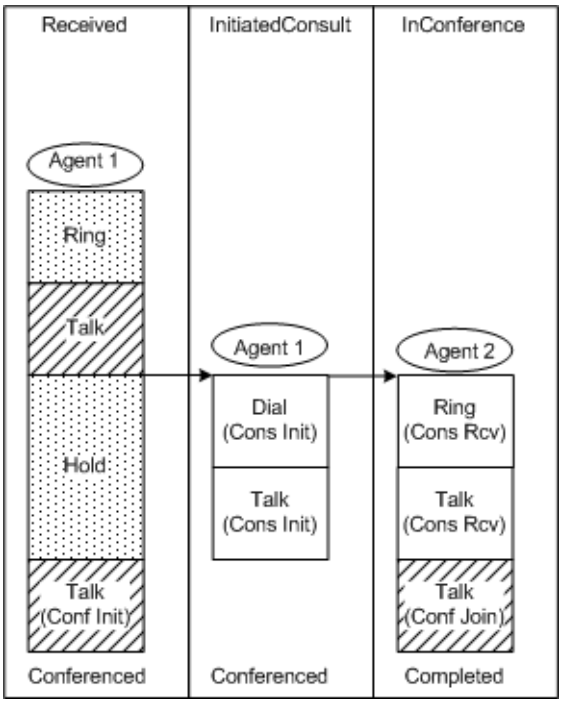

Figure 20: Consult to Agent Directly, and Then Conference

Outbound Call Flow Example

This call topology shows a call flow example of a direct outbound call. An agent dials an off-switch number. After talking with an external party, the agent hangs up.

- Figure 21 depicts the Interaction Resource Fact (IRF) representation of the call topology.

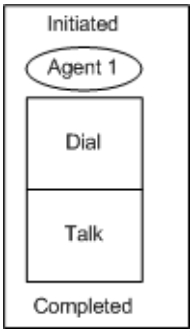

Figure 21: Agent Dials External Party

Internal Call Flow Examples

This subsection contains several examples of internal call flows. Each example represents a different outcome:

- An ACD queue directs the internal call to another agent.
- The internal call is answered directly by another agent (see page 83).
- An agent mute transfers the call to an ACD queue (see page 84).
- An agent mute transfers the call to another agent (see page 85).
- An agent consults to an ACD queue, and then retrieves the call (see page 87).
- An agent consults to another agent, and then retrieves the call (see page 89).
- An agent consults to an ACD queue, and then transfers the call (see page 91).
- An agent consults to another agent, and then transfers the call (see page 93).
- An agent consults to an ACD queue, and then conferences the call (see page 95).
- An agent consults to another agent, and then conferences the call (see page 97).

Internal to Agent via ACD Queue

This call topology shows the outcome of an internal call to an agent via an ACD queue. An agent initiates a call to the ACD queue, and the interaction is diverted to another agent.

- Figure 22 depicts the Interaction Resource Fact (IRF) representation of the call topology.

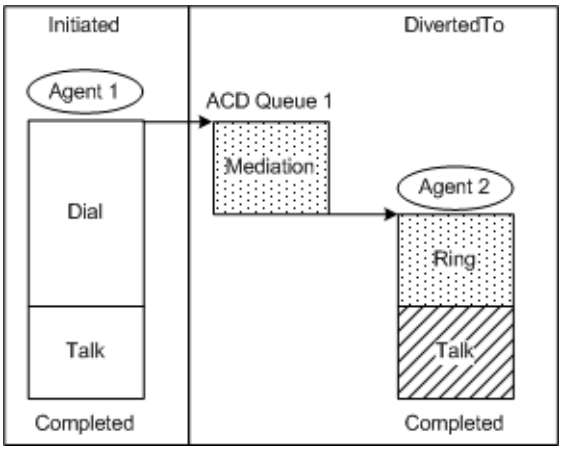

Figure 22: Internal Call to Another Agent via ACD Queue

Internal to Agent Directly

This call topology shows the outcome of a call that an agent initiates directly to another agent.

- Figure 23 depicts the Interaction Resource Fact (IRF) representation of the call topology.

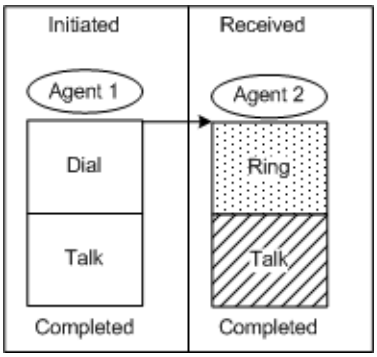

Figure 23: Internal Call from One Agent to Another

Mute Transfer to ACD Queue

In this call topology, an agent initiates a call to another agent. One of the agents then mute transfers the call to an ACD queue, and the interaction is diverted to another agent.

This section shows two possible outcomes of a call that is mute transferred to an ACD queue:

- The receiver (`Agent 2`) initiates the transfer (see "Mute Transfer to ACD Queue—Call Receiver Initiates Transfer" on page 84).
- The initiator (`Agent 1`) initiates the transfer (see "Mute Transfer to ACD Queue—Call Initiator Initiates Transfer" on page 85).

Mute Transfer to ACD Queue—Call Receiver Initiates Transfer

For this outcome, the receiving agent initiates a mute transfer to the ACD queue.

- Figure 24 depicts the Interaction Resource Fact (IRF) representation of the call topology.

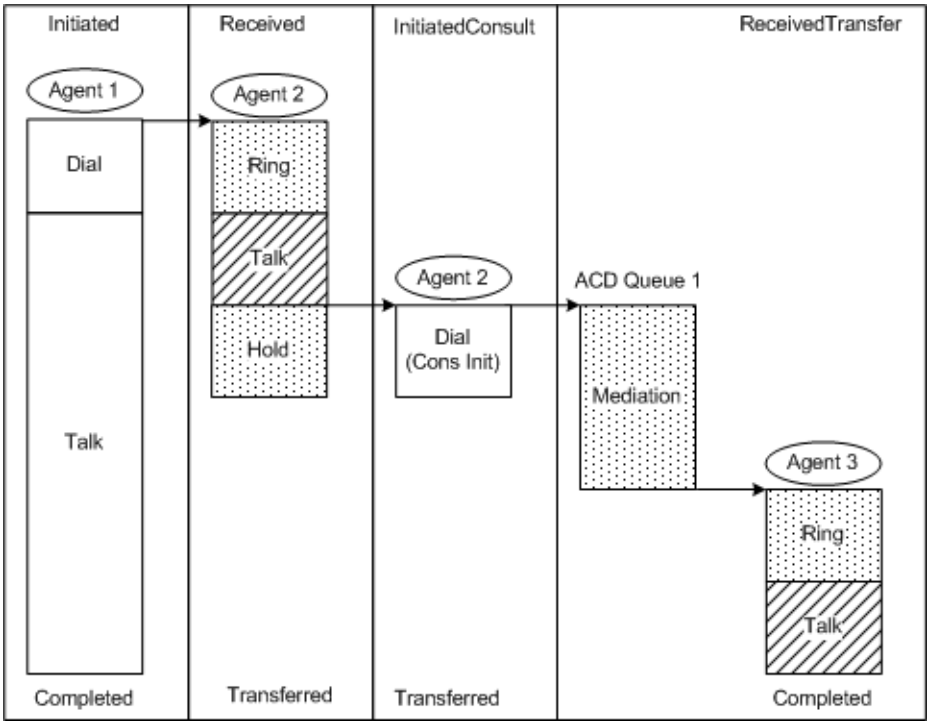

Figure 24: Receiving Agent Initiates Transfer to ACD Queue

Mute Transfer to ACD Queue—Call Initiator Initiates Transfer

For this outcome, the initiating agent initiates a mute transfer to the ACD queue.

- Figure 25 depicts the Interaction Resource Fact (IRF) representation of the call topology.

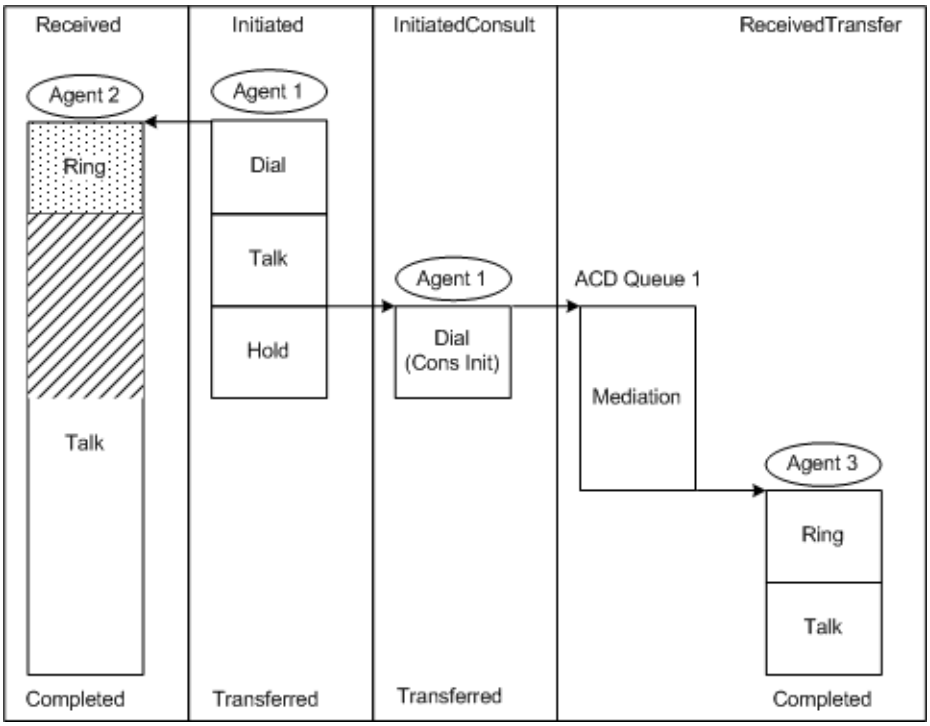

Figure 25: Initiating Agent Initiates Transfer to ACD Queue

Mute Transfer to Agent

In this call topology, an agent initiates a call to another agent. One of the agents then mute transfers the call to another agent.

This section shows two possible outcomes of a call that is mute transferred directly to an agent:

- The receiver (`Agent 2`) initiates the transfer (see "Mute Transfer to Agent—Call Receiver Initiates Transfer" on page 86).
- The initiator (`Agent 1`) initiates the transfer (see "Mute Transfer to Agent—Call Initiator Initiates Transfer" on page 87).

Mute Transfer to Agent—Call Receiver Initiates Transfer

For this outcome, the receiving agent initiates a mute transfer to another agent.

- Figure 26 depicts the Interaction Resource Fact (IRF) representation of the call topology.

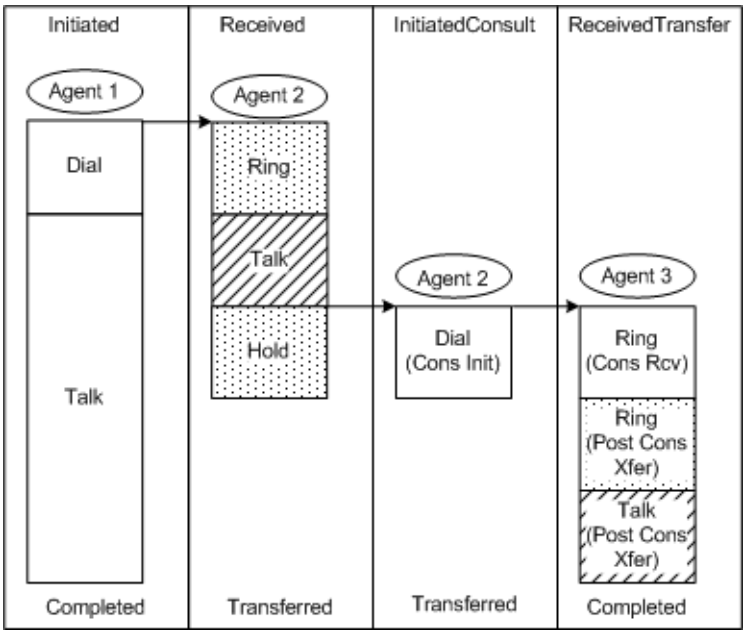

Figure 26: Receiving Agent Initiates Transfer to Another Agent

Mute Transfer to Agent—Call Initiator Initiates Transfer

For this outcome, the initiating agent initiates a mute transfer to another agent.

- Figure 27 depicts the Interaction Resource Fact (IRF) representation of the call topology.

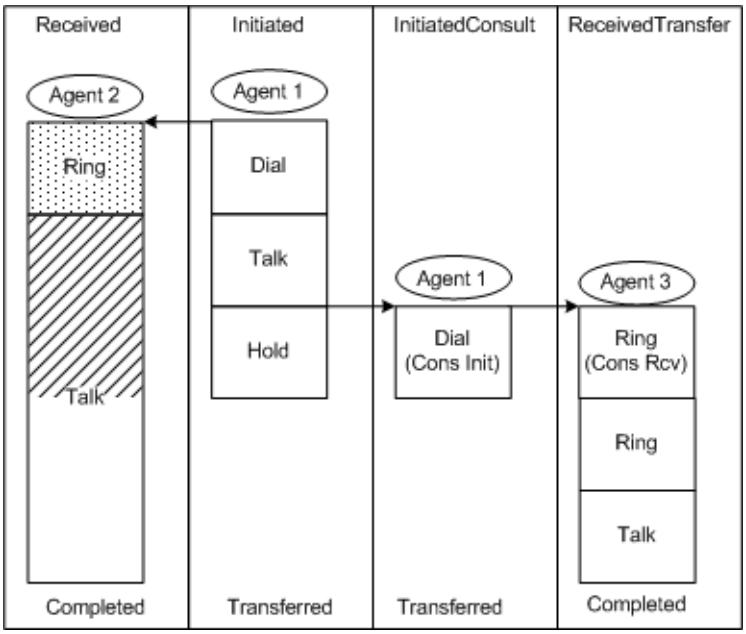

Figure 27: Initiating Agent Initiates Transfer to Another Agent

Consult to Agent via ACD Queue, and Then Retrieve

In this call topology, an agent initiates a call to another agent. One of the agents then initiates a consultation to an ACD queue, and the interaction is diverted to another agent. The consultation ends when the consulting agent retrieves the interaction.

This section shows two possible outcomes of a call that is retrieved after a consultation has been initiated:

- The receiver (`Agent 2`) initiates the consultation (see “Receiving Agent Consults to ACD Queue, and Then Retrieves” on page 88).
- The initiator (`Agent 1`) initiates the consultation (see “Initiating Agent Consults to ACD Queue, and Then Retrieves” on page 89)

Receiving Agent Consults to ACD Queue, and Then Retrieves

For this outcome, the receiving agent initiates a consultation to the ACD queue.

- Figure 28 depicts the Interaction Resource Fact (IRF) representation of the call topology.

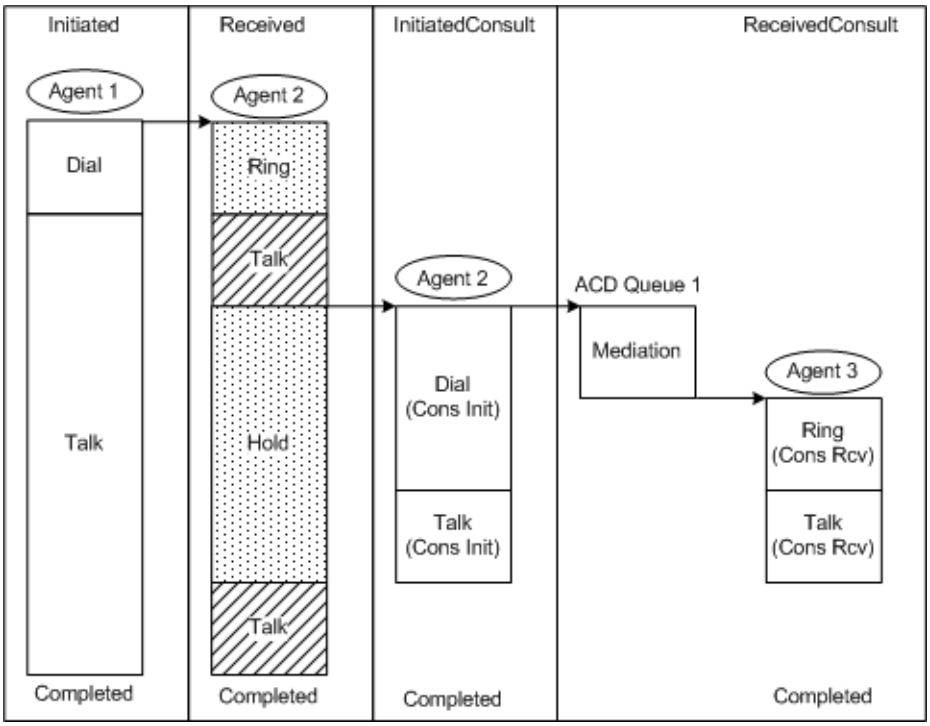

Figure 28: Receiving Agent Consults to ACD Queue Then Retrieves

Initiating Agent Consults to ACD Queue, and Then Retrieves

For this outcome, the initiating agent initiates a consultation to the ACD queue.

- Figure 29 depicts the Interaction Resource Fact (IRF) representation of the call topology.

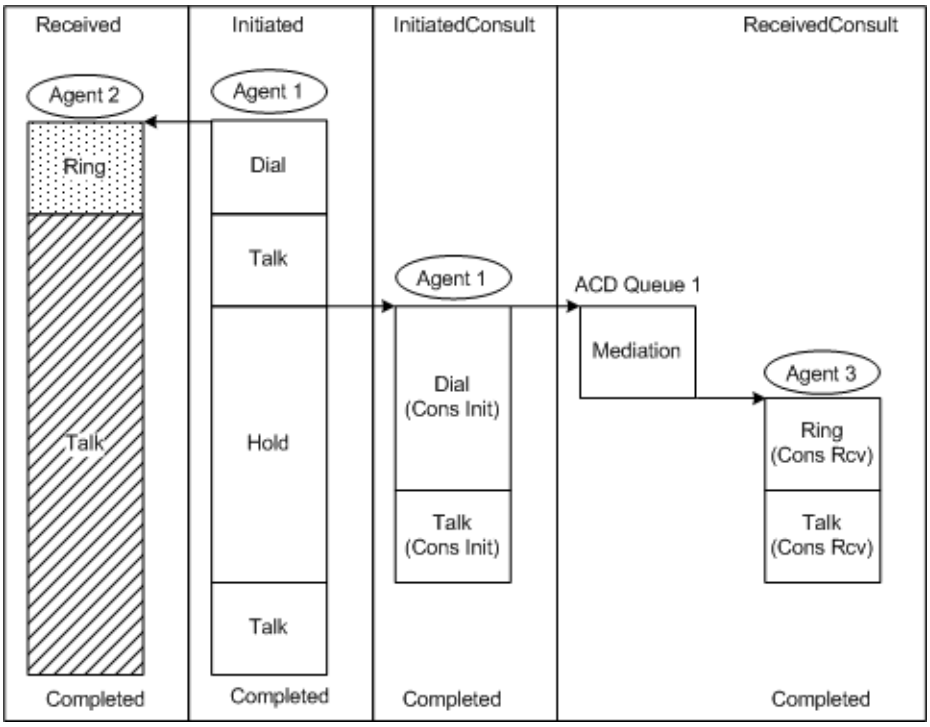

Figure 29: Initiating Agent Consults to ACD Queue Then Retrieves

Consult to Agent, and Then Retrieve

In this call topology, an agent initiates a call to another agent. One of the agents then initiates a consultation to a third agent. The consultation ends when the consulting agent retrieves the interaction.

This section shows two possible outcomes of a call that is retrieved after a consultation has been initiated:

- The receiver (`Agent 2`) initiates the consultation (see “Receiving Agent Consults to Another Agent, and Then Retrieves” on page 90).
- The initiator (`Agent 1`) initiates the consultation (see “Initiating Agent Consults to Another Agent, and Then Retrieves” on page 91)

Receiving Agent Consults to Another Agent, and Then Retrieves

For this outcome, the receiving agent initiates a consultation to another agent.

- Figure 30 depicts the Interaction Resource Fact (IRF) representation of the call topology.

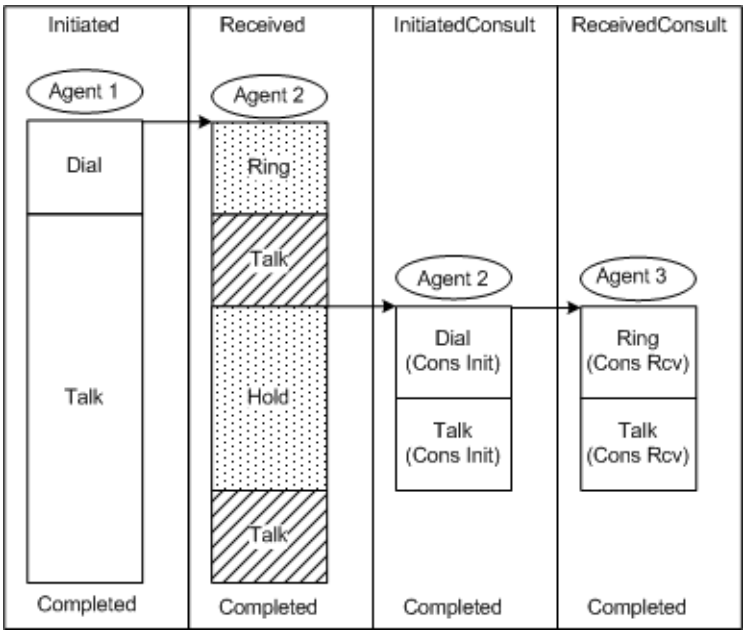

Figure 30: Receiving Agent Consults to Another Agent Then Retrieves

Initiating Agent Consults to Another Agent, and Then Retrieves

For this outcome, the initiating agent initiates a consultation to another agent.

- Figure 31 depicts the Interaction Resource Fact (IRF) representation of the call topology.

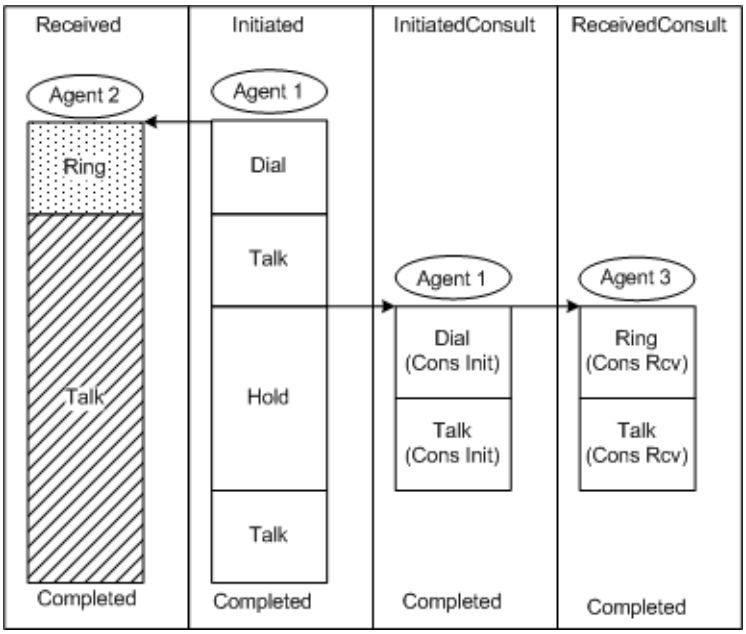

Figure 31: Initiating Agent Consults to Another Agent Then Retrieves

Consult to Agent via ACD Queue, and Then Transfer

In this call topology, an agent initiates a call to another agent. One of the agents then initiates a consultation to an ACD queue, and the interaction is diverted to another agent. The consultation ends when the consulting agent transfers the interaction.

This section shows two possible outcomes of a call that is transferred after a consultation:

- The receiver (`Agent 2`) initiates the consultation (see "Receiving Agent Consults to ACD Queue, and Then Transfers" on page 92).
- The initiator (`Agent 1`) initiates the consultation (see "Initiating Agent Consults to ACD Queue, and Then Transfers" on page 93)

Receiving Agent Consults to ACD Queue, and Then Transfers

For this outcome, the receiving agent initiates a consultation to the ACD queue.

- Figure 32 depicts the Interaction Resource Fact (IRF) representation of the call topology.

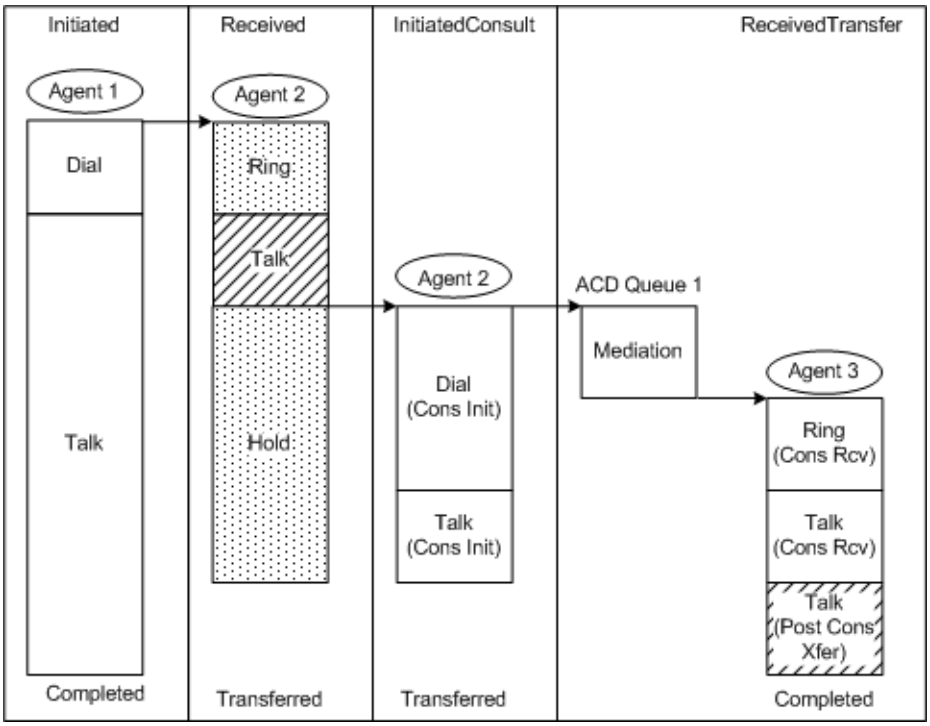

Figure 32: Receiving Agent Consults to ACD Queue Then Transfers

Initiating Agent Consults to ACD Queue, and Then Transfers

For this outcome, the initiating agent initiates a consultation to the ACD queue.

- Figure 33 depicts the Interaction Resource Fact (IRF) representation of the call topology.

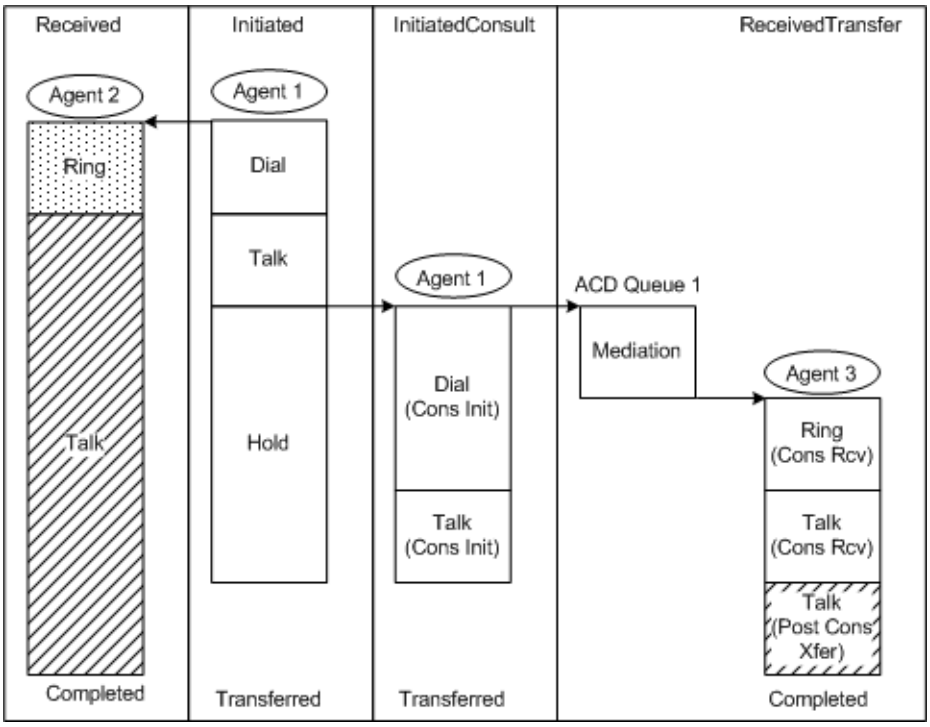

Figure 33: Initiating Agent Consults to ACD Queue Then Transfers

Consult to Agent, and Then Transfer

In this call topology, an agent initiates a call to another agent. One of the agents then initiates a consultation to a third agent. The consultation ends when the consulting agent transfers the interaction.

This section shows two possible outcomes of a call that is transferred after a consultation has been initiated:

- The receiver (`Agent 2`) initiates the consultation (see “Receiving Agent Consults to Another Agent, and Then Transfers” on page 94).
- The initiator (`Agent 1`) initiates the consultation (see “Initiating Agent Consults to Another Agent, and Then Transfers” on page 95)

Receiving Agent Consults to Another Agent, and Then Transfers

For this outcome, the receiving agent initiates a consultation to another agent.

- Figure 34 depicts the Interaction Resource Fact (IRF) representation of the call topology.

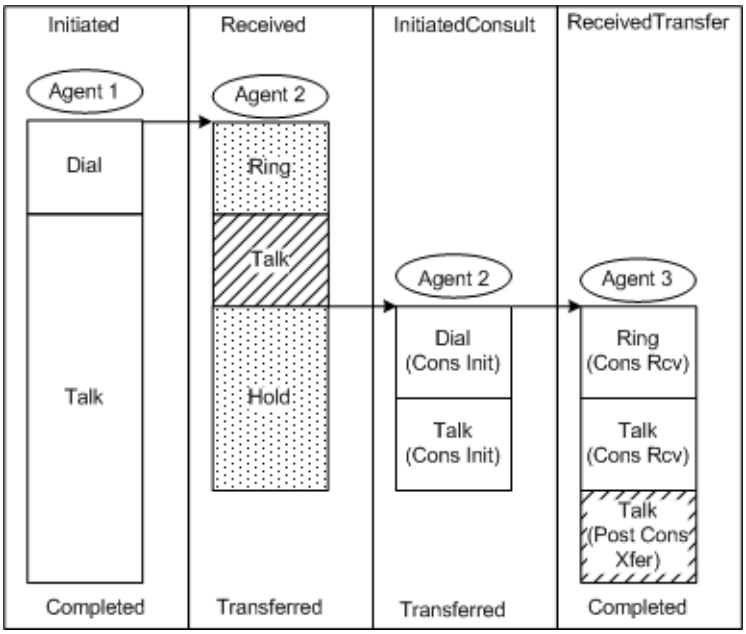

Figure 34: Receiving Agent Consults to Another Agent Then Transfers

Initiating Agent Consults to Another Agent, and Then Transfers

For this outcome, the initiating agent initiates a consultation to another agent.

- Figure 35 depicts the Interaction Resource Fact (IRF) representation of the call topology.

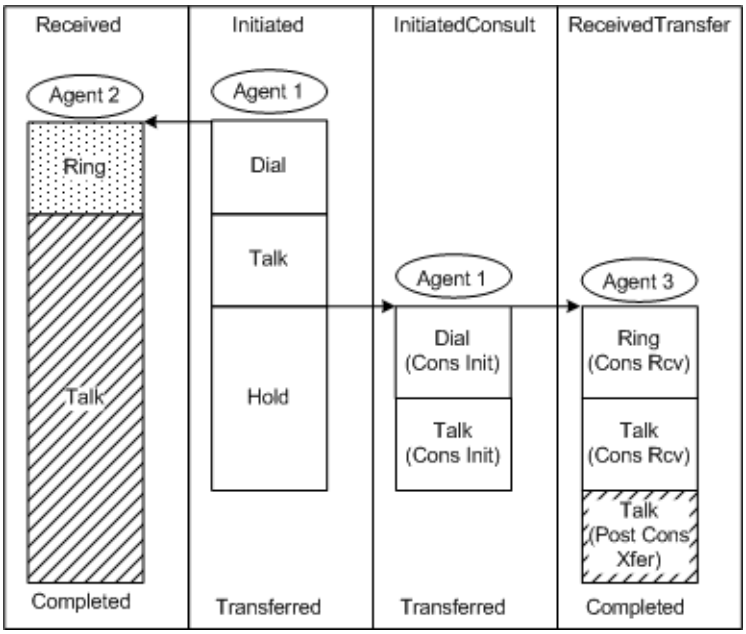

Figure 35: Initiating Agent Consults to Another Agent Then Transfers

Consult to Agent via ACD Queue, and Then Conference

In this call topology, an agent initiates a call to another agent. One agent then initiates a consultation to an ACD queue, and the interaction is diverted to a third agent. The consultation ends when the consulting agent conferences the interaction.

This section shows two possible outcomes of a call that is conferenced after a consultation:

- The receiver (`Agent 2`) initiates the consultation (see "Receiving Agent Consults to ACD Queue, and Then Conferences" on page 96).
- The initiator (`Agent 1`) initiates the consultation (see "Initiating Agent Consults to ACD Queue, and Then Conferences" on page 97).

Receiving Agent Consults to ACD Queue, and Then Conferences

For this outcome, the receiving agent initiates a consultation to the ACD queue.

- Figure 36 depicts the Interaction Resource Fact (IRF) representation of the call topology.

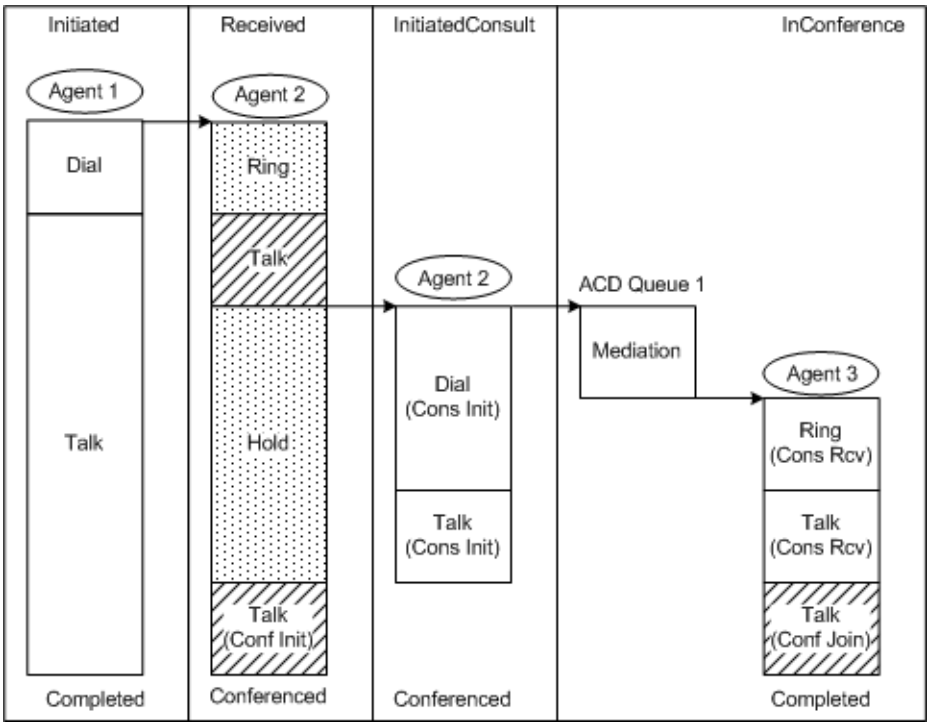

Figure 36: Receiving Agent Consults to ACD Queue Then Conferences

Initiating Agent Consults to ACD Queue, and Then Conferences

For this outcome, the initiating agent initiates a consultation to the ACD queue.

- Figure 37 depicts the Interaction Resource Fact (IRF) representation of the call topology.

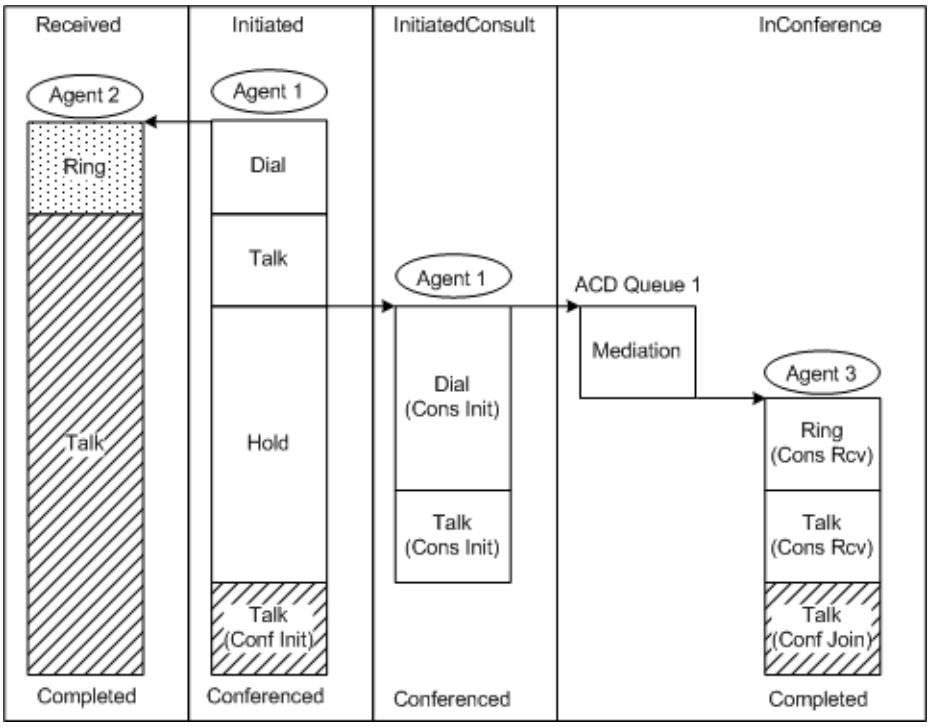

Figure 37: Initiating Agent Consults to ACD Queue Then Conferences

Consult to Agent, and Then Conference

In this call topology, an agent initiates a call to another agent. One agent then initiates a consultation to a third agent. The consultation ends when the consulting agent conferences the interaction.

This section shows two possible outcomes of a call that is conferenced after a consultation:

- The receiver (`Agent 2`) initiates the consultation (see “Receiving Agent Consults to Another Agent, and Then Conferences” on page 98).
- The initiator (`Agent 1`) initiates the consultation (see “Initiating Agent Consults to Another Agent, and Then Conferences” on page 99)

Receiving Agent Consults to Another Agent, and Then Conferences

For this outcome, the receiving agent initiates a consultation to another agent.

- Figure 38 depicts the Interaction Resource Fact (IRF) representation of the call topology.

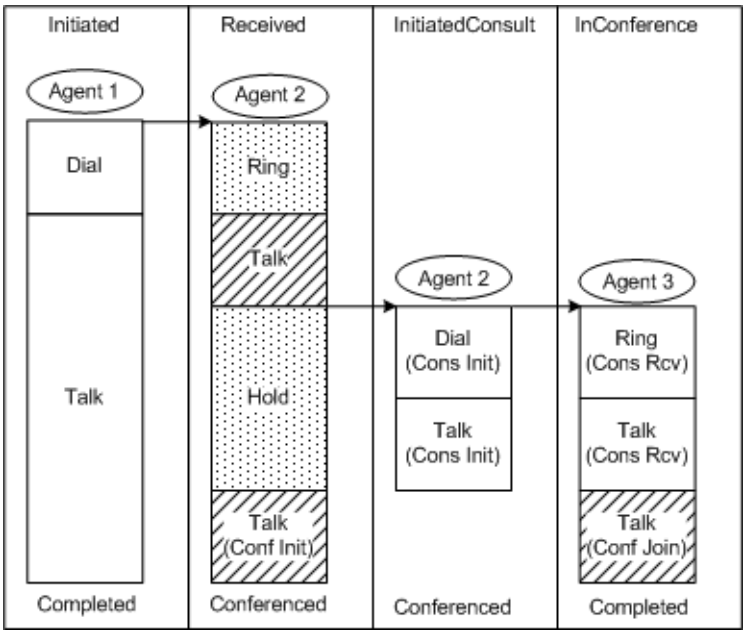

Figure 38: Receiving Agent Consults to Another Agent Then Conferences

Initiating Agent Consults to Another Agent, and Then Conferences

For this outcome, the initiating agent initiates a consultation to another agent.

- Figure 39 depicts the Interaction Resource Fact (IRF) representation of the call topology.

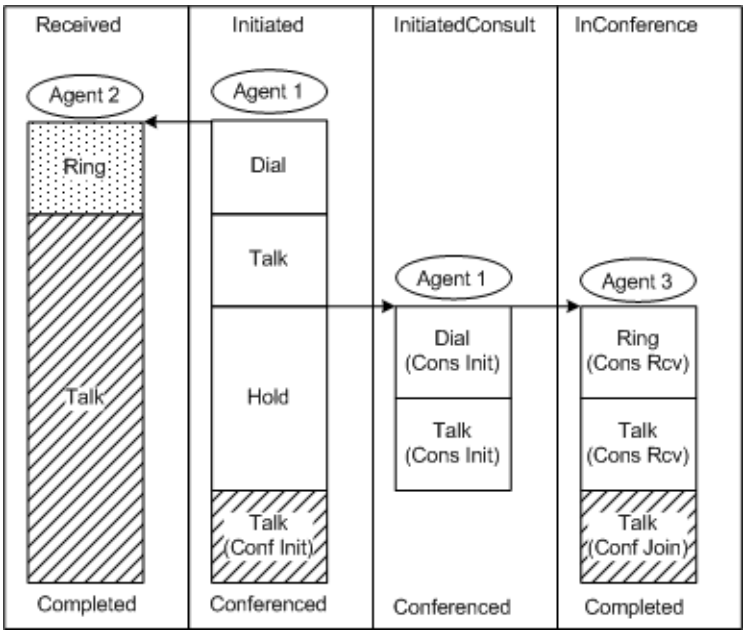

Figure 39: Initiating Agent Consults to Another Agent Then Conferences

IVR-in-Front-of-Switch Call Flows

Voice interactions arrive at an IVR that is visible to the IVR Server's virtual T-Server. Either self-service, or simply front-end identification and segmentation, can be the focus of the IVR application. If the IVR application cannot completely handle the voice interaction, the interaction can be transferred to an ACD queue behind the switch that represents a requested skill, service type, or customer segment. Agents who are logged in to the ACD queues handle the interactions.

Inbound Call Flow Examples

This subsection contains several examples of inbound call flows. Each example represents a different outcome:

- An inbound call arrives at an IVR port.
- The IVR transfers the call to an ACD queue (see page 100).
- The IVR transfers the call to an agent (see page 102).

Inbound to IVR Port

This call topology shows the outcomes of a call that arrives at an IVR port.

- Figure 40 depicts the Interaction Resource Fact (IRF) representation of the call topology when the call completes normally in the case of a *self-service* (SS) IVR (when the IVR is in its own box).
- Figure 41 depicts the Interaction Resource Fact (IRF) representation of the call topology when the call is abandoned by the customer. This is the *nonself-service* (nonSS) IVR case.

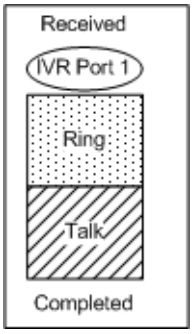

Figure 40: Inbound Interaction to IVR Port, Completed

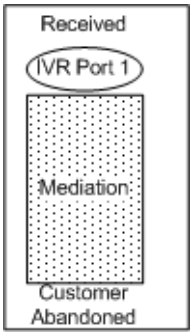

Figure 41: Inbound Interaction to IVR Port, Abandoned

IVR Transfer to ACD Queue

This call topology shows the outcome of an interaction that arrives at an IVR port, which hook-flash transfers the interaction to an ACD queue.

- Figure 42 on page 101 depicts the Interaction Resource Fact (IRF) representation of the call topology in the case of an SS IVR (when the IVR IVR is in its own box).
- Figure 43 on page 101 depicts the Interaction Resource Fact (IRF) representation of the call topology in the case of a nonSS IVR.

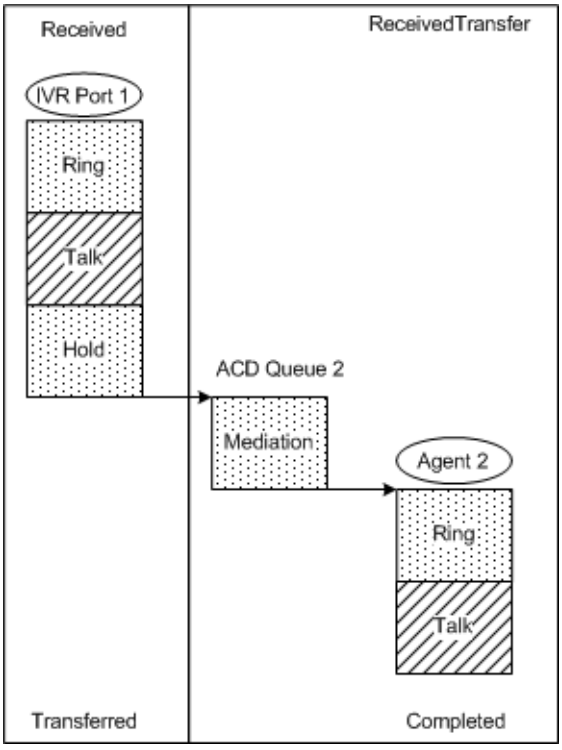

Figure 42: IVR Hook-Flash Transfer to ACD Queue (SS IVR)

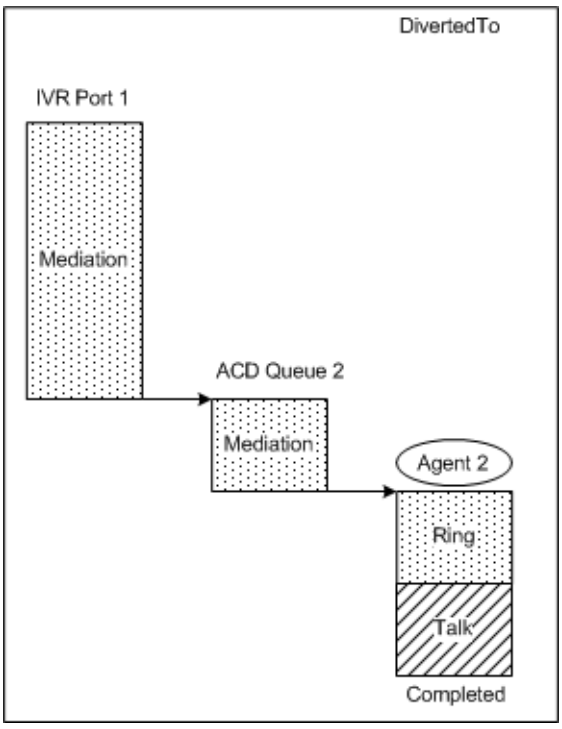

Figure 43: IVR Hook-Flash Transfer to ACD Queue (nonSS IVR)

IVR Transfer to Agent

This call topology shows the outcome of an interaction that arrives at an IVR port, which hook-flash transfers the interaction to an agent.

- Figure 44 depicts the Interaction Resource Fact (IRF) representation of the call topology in the case of an SS IVR.
- Figure 45 depicts the Interaction Resource Fact (IRF) representation of the call topology in the case of a nonSS IVR.

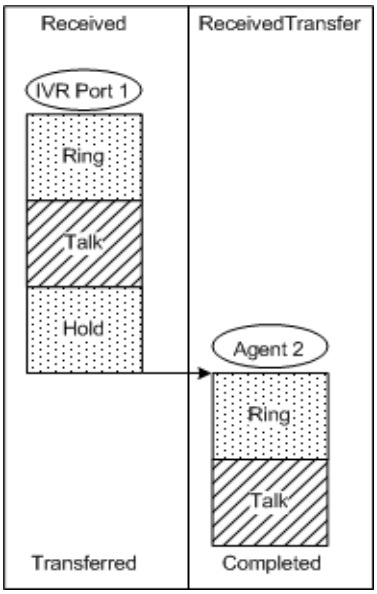

Figure 44: IVR Hook-Flash Transfer to Agent (SS IVR)

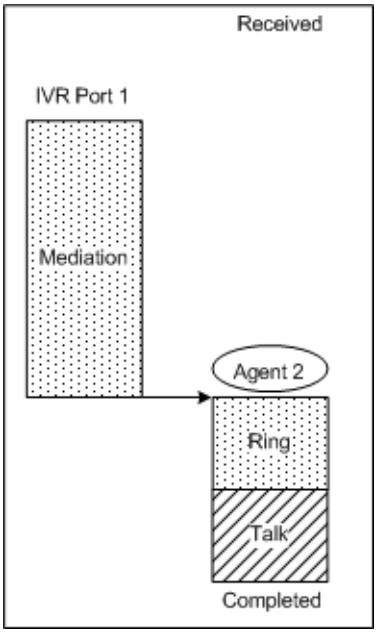

Figure 45: IVR Hook-Flash Transfer to Agent (nonSS IVR)

IVR-Behind-Switch Call Flows

Voice interactions that arrive at the switch are queued to an ACD queue, where the ACD positions are actually IVR ports. Either self-service, or simply front-end identification and segmentation, can be the focus of the IVR application. If the IVR application cannot completely handle the voice interaction, the interaction can be transferred to an ACD queue that represents a requested skill, service type, or customer segment. Agents who are logged in to the ACD queues handle the interactions.

Inbound Call Flow Examples

This subsection contains several examples of inbound call flows. Each example represents a different outcome.

- An inbound call arrives at an IVR via an ACD queue.
- An inbound call arrives directly at an IVR (see page 104).
- An IVR mute transfers the call to another ACD queue (see page 105).
- An IVR mute transfers the call to an agent (see page 107).

Inbound to IVR via ACD Queue

This call topology shows the outcome of an inbound call to an IVR via an ACD queue. The interaction arrives at the ACD queue and is diverted to an IVR port.

- Figure 46 depicts the Interaction Resource Fact (IRF) representation of the call topology when the call completes normally in the case of a *self-service* (SS) IVR (when the IVR is in its own box).
- Figure 47 on page 104 depicts the Interaction Resource Fact (IRF) representation of the call topology when the call is abandoned by the customer. This is the *nonself-service* (nonSS) IVR case.

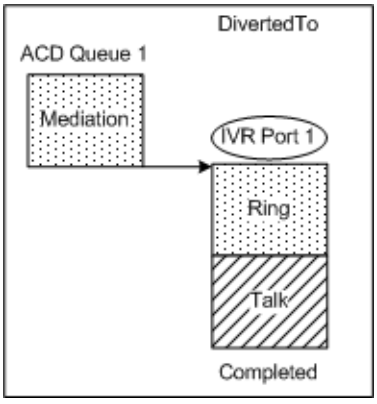

Figure 46: Inbound to IVR via ACD Queue, Completed

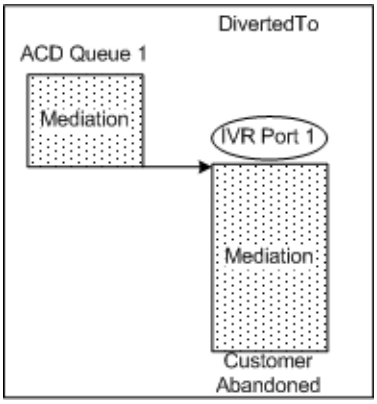

Figure 47: Inbound to IVR via ACD Queue, Abandoned

Inbound to IVR Directly

This call topology shows the outcome of a call that arrives directly at an IVR port.

- Figure 48 depicts the Interaction Resource Fact (IRF) representation of the call topology when the call completes normally. This is the SS IVR case (when the IVR is in its own box).
- Figure 49 on page 105 depicts the Interaction Resource Fact (IRF) representation of the call topology when the call is abandoned by the customer. This is the nonSS IVR case.

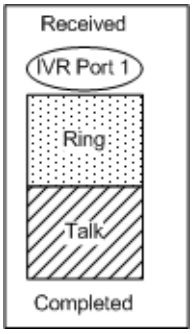

Figure 48: Inbound to IVR Directly, Completed

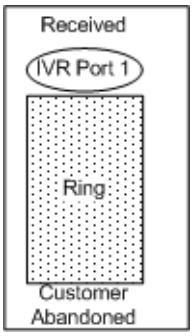

Figure 49: Inbound to IVR Directly, Abandoned

Mute Transfer to ACD Queue

This call topology shows the outcome of a call that is mute transferred to an agent via an ACD queue. The interaction arrives at an ACD queue and is diverted to an IVR port. The IVR then mute transfers the call to another ACD queue.

- Figure 50 depicts the Interaction Resource Fact (IRF) representation of the call topology in the case of an SS IVR (when the IVR is in its own box).
- Figure 51 on page 106 depicts the Interaction Resource Fact (IRF) representation of the call topology in the case of a nonSS IVR.

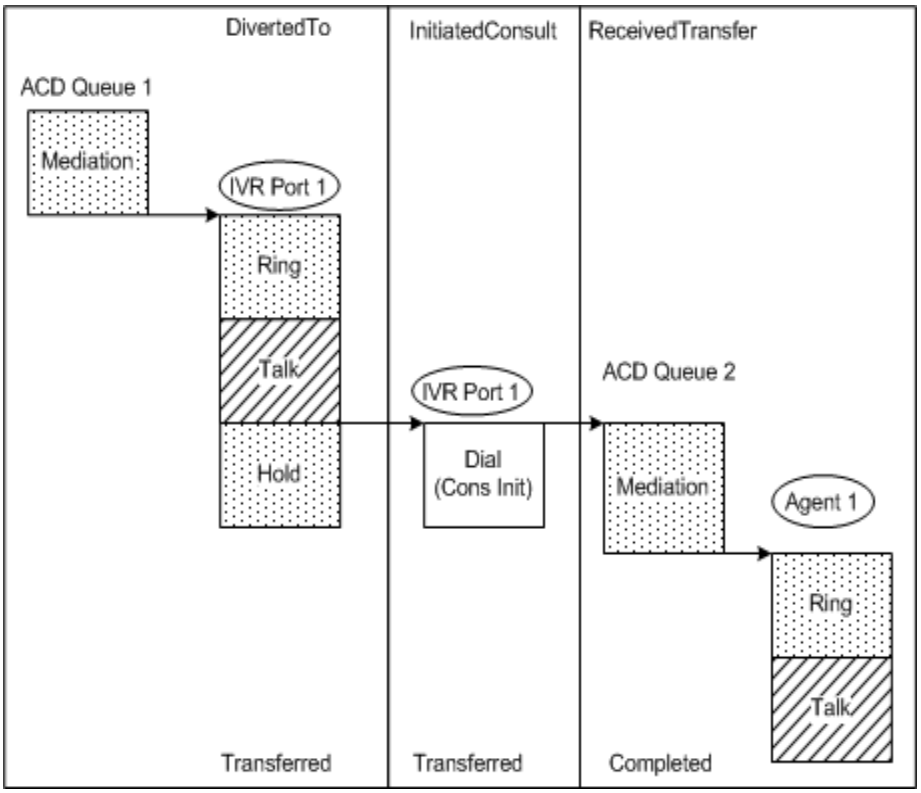

Figure 50: IVR Mute Transfers to ACD Queue (SS IVR)

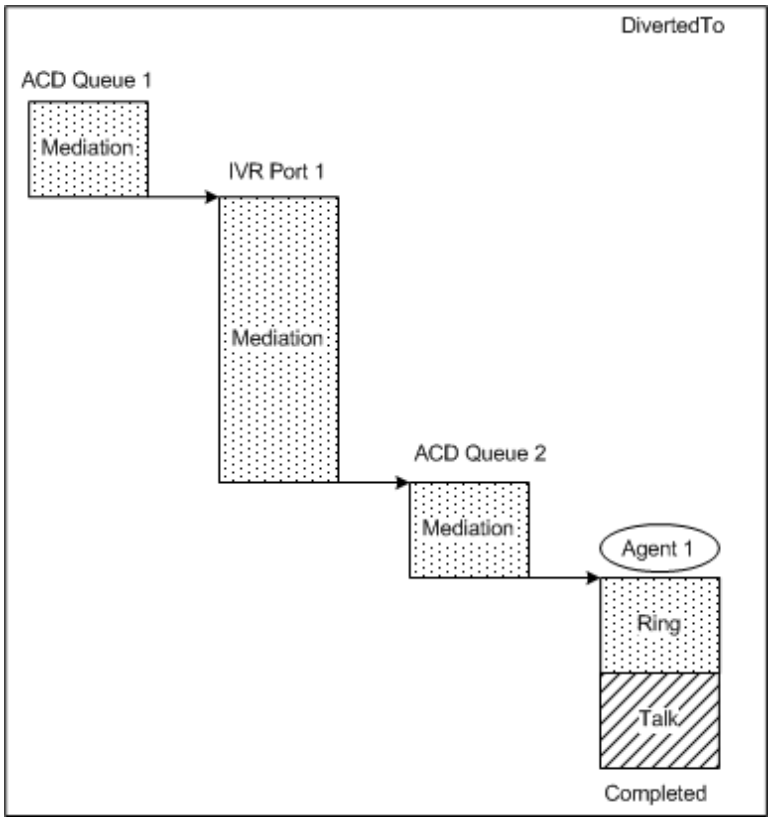

Figure 51: IVR Mute Transfers to ACD Queue (nonSS IVR)

Mute Transfer to Agent

This call topology shows the outcome of a call that is mute transferred to an agent. The interaction arrives at an ACD queue and is diverted to an IVR port. The IVR then mute transfers the call to an agent.

- Figure 52 depicts the Interaction Resource Fact (IRF) representation of the call topology in the case of an SS IVR (when the IVR is in its own box).
- Figure 53 on page 108 depicts the Interaction Resource Fact (IRF) representation of the call topology in the case of a nonSS IVR.

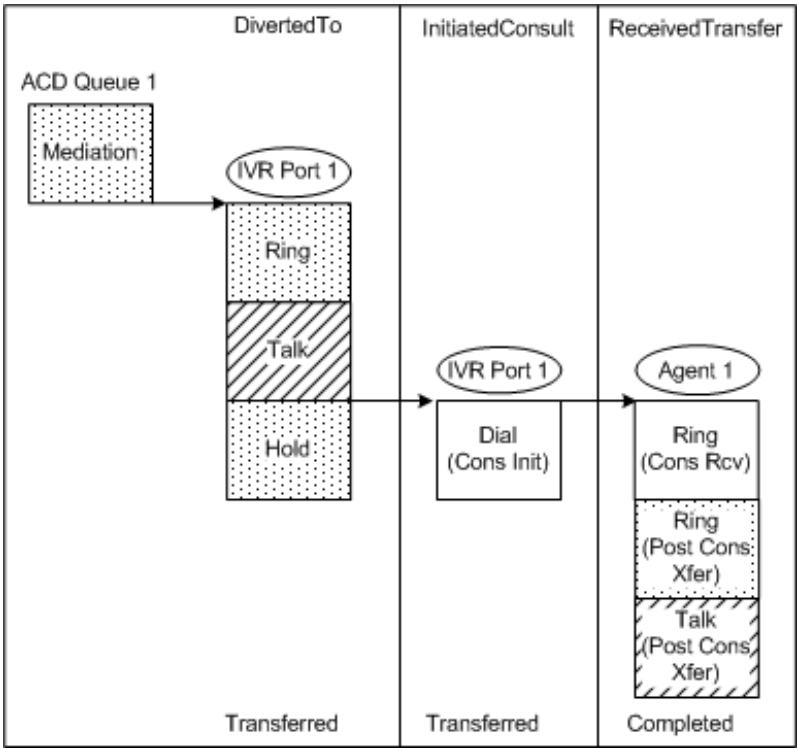

Figure 52: IVR Mute Transfers to Agent (SS IVR)

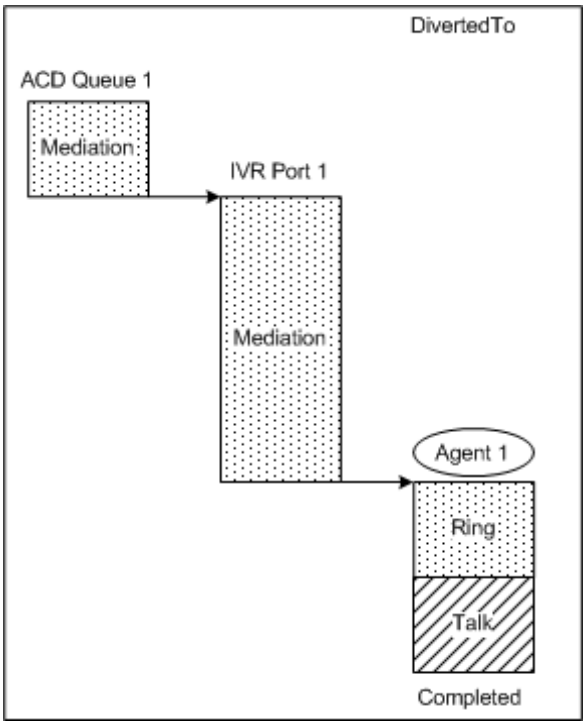

Figure 53: IVR Mute Transfers to Agent (nonSS IVR)

Universal Routing Call Flows

Voice interactions that arrive at the switch are delivered to a routing point. Universal Routing Server uses ANI, DNIS, or the date and time of day to collect information and select an appropriate routing target. Basic targets are ACD queues and individual DNs. More advanced targets are agent groups, place groups, and skill expressions.

Inbound Interactions

This subsection contains the following examples of routed call flows. Each example represents a different outcome:

- A Routing Point routes the call to an ACD queue.
- A Routing Point routes the call to an agent (see page 110).

Routing Point Routes to ACD Queue

This call topology shows the outcome of a call that is routed to an agent via an ACD queue. The call arrives at the Routing Point. The Routing Point then routes the call to an ACD queue, and the interaction is diverted to an agent.

- Figure 54 depicts the Interaction Resource Fact (IRF) representation of the call topology.

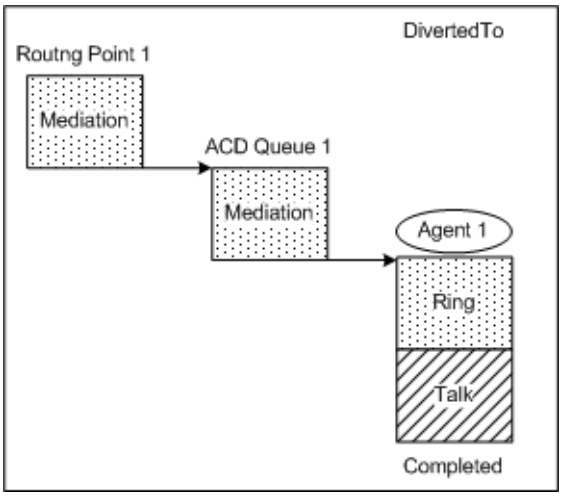

Figure 54: Routing Point Routes to ACD Queue

Note: Figure 54 applies to both network routing and enterprise routing. For network routing, `Routing Point 1` could be a service number on a network T-Server that routes the voice interaction to `ACD Queue 1` on a premise T-Server.

Routing Point Routes to Agent

This call topology shows the outcome of a call that is routed directly to an agent. The call arrives at the Routing Point. The Routing Point then routes the call to an agent.

- Figure 55 depicts the Interaction Resource Fact (IRF) representation of the call topology.

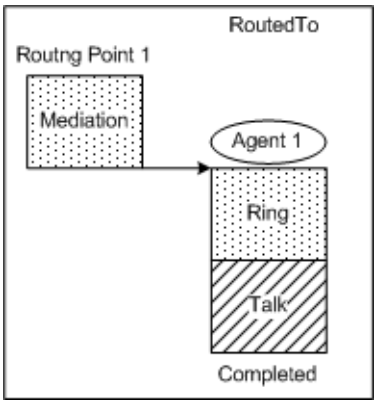

Figure 55: Routing Point Routes to Agent—IRF

Note: Figure 55 applies to both network routing and enterprise routing. For network routing, `Routing Point 1` could be a service number on a network T-Server that routes the voice interaction to `Agent 1` on a premise T-Server.

Universal Routing Assisted by IVR-Behind-Switch Call Flows

Voice interactions that arrive at the switch are queued to an ACD queue, where the ACD positions are actually IVR ports. The IVR collects digits and information about the caller and transfers the call to a Routing Point. Universal Routing uses the collected information to select an appropriate routing target. Basic targets are ACD queues and individual DNs. More advanced targets are agent groups, place groups, and skill expressions.

Inbound Call Flow Examples

This subsection contains several examples of routed call flows. Each example represents a different outcome:

- A Routing Point routes the call to an ACD queue.
- A Routing Point routes the call to an agent (see page 112).

Routing Point Routes to ACD Queue

This call topology shows the outcome of a call that is routed to an agent via an ACD queue. The call arrives at an ACD queue and is diverted to an IVR port. The IVR then transfers the call to a Routing Point, which routes the call to an ACD queue.

- Figure 56 depicts the Interaction Resource Fact (IRF) representation of the call topology in the case of a *self-service* (SS) IVR (when the IVR is in its own box).
- Figure 57 on page 112 depicts the Interaction Resource Fact (IRF) representation of the call topology in the case of a *nonself-service* (nonSS) IVR.

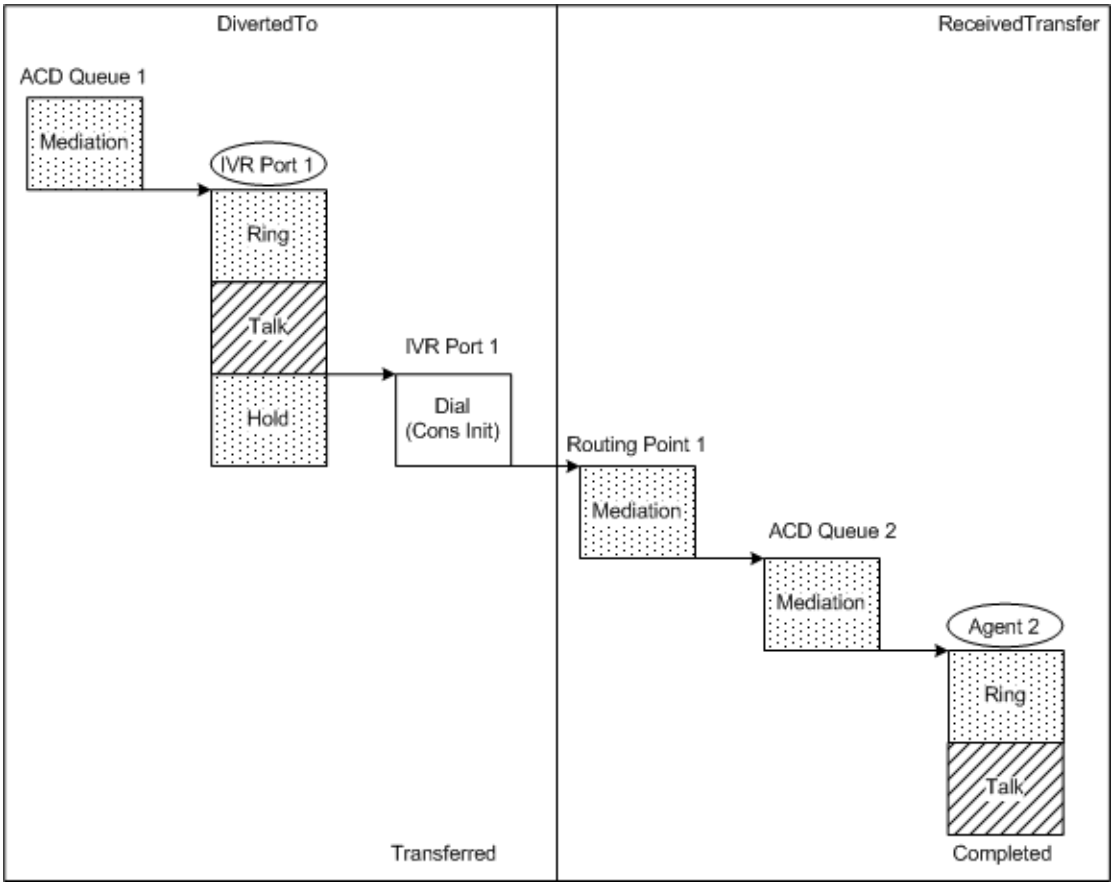

Figure 56: Routing Point Routes to ACD Queue (SS IVR)

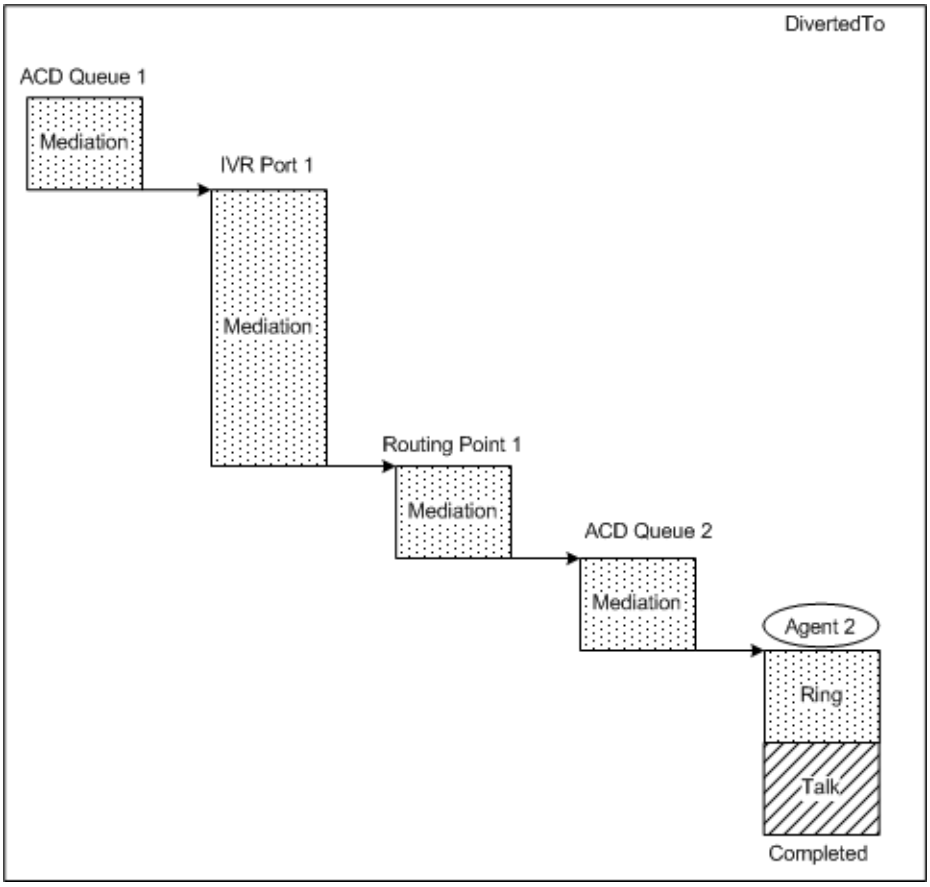

Figure 57: Routing Point Routes to ACD Queue (nonSS IVR)

Routing Point Routes to Agent

This call topology shows the outcome of a call that is routed directly to an agent. The call arrives at an ACD queue and is diverted to an IVR port. The IVR then transfers the call to a Routing Point, which routes the call to an agent.

- Figure 58 on page 113 depicts the Interaction Resource Fact (IRF) representation of the call topology in the case of an SS IVR (when the IVR is in its own box).
- Figure 59 on page 113 depicts the Interaction Resource Fact (IRF) representation of the call topology in the case of a nonSS IVR.

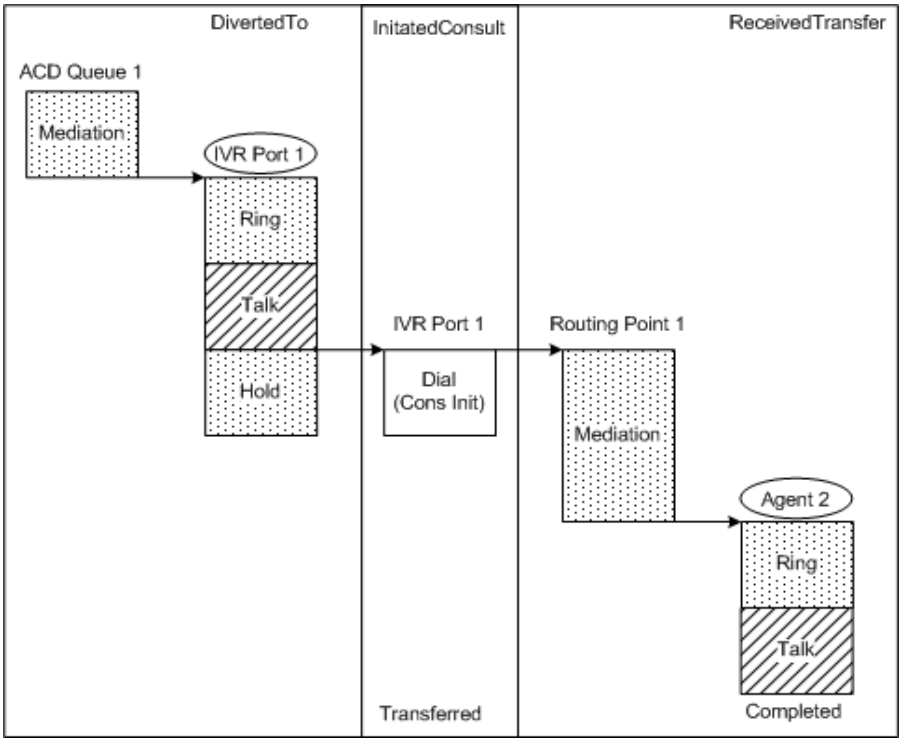

Figure 58: Routing Point Routes to Agent (SS IVR)

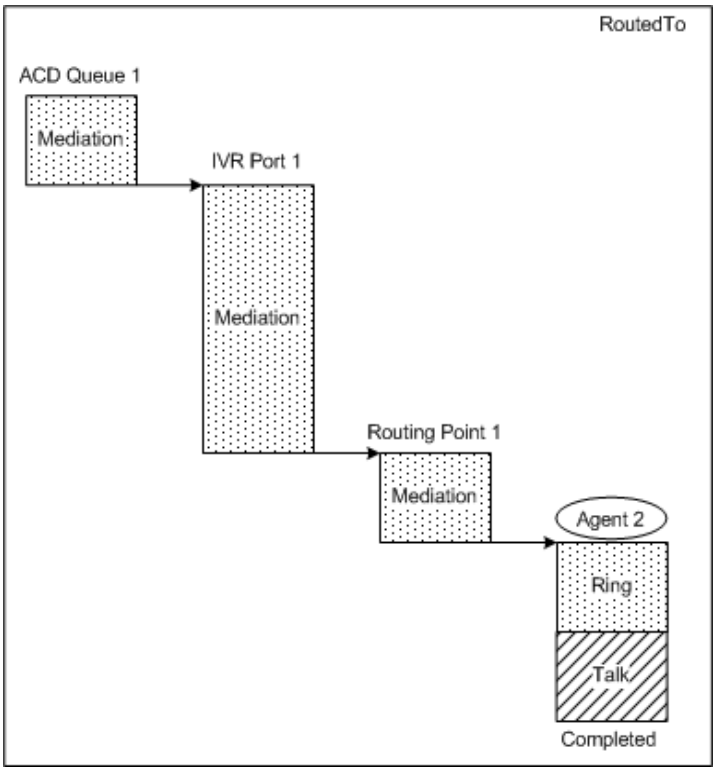

Figure 59: Routing Point Routes to Agent (nonSS IVR)

IVR-in-Front-of-Switch Assisted by Universal Routing Call Flows

Voice interactions arrive at an IVR that is visible to the IVR Server's virtual T-Server. Through a Routing Point in the IVR Server's virtual T-Server, the IVR application invokes an Universal Routing strategy. Universal Routing instructs the IVR application to play applications or collect information. Universal Routing uses the collected information to return an appropriate target. The IVR application hook-flash transfers the call to that target.

Inbound Interactions

This subsection contains several examples of inbound call flows. Each example represents a different outcome:

- The IVR transfers the call to an ACD queue.
- The IVR transfers the call to an agent (see page 115).

IVR Transfers to ACD Queue

This call topology shows the outcome of a call that an IVR transfers to an ACD queue, in accordance with routing instructions. The call arrives at an IVR port. The IVR requests routing instructions from a Routing Point, and then hook-flash transfers the call to an ACD queue.

- Figure 60 on page 115 depicts the Interaction Resource Fact (IRF) representation of the call topology.

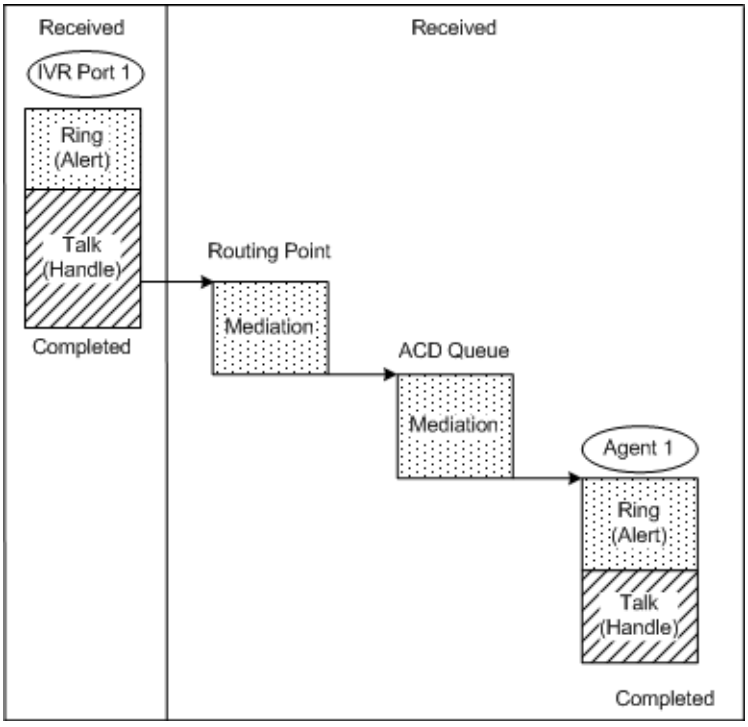

Figure 60: IVR Transfers to ACD Queue

IVR Transfers to Agent

This call topology shows the outcome of a call that an IVR transfers directly to an agent, in accordance with routing instructions. The call arrives at an IVR port. The IVR requests routing instructions from a Routing Point, and then hook-flash transfers the call to an agent.

- Figure 61 on page 116 depicts the Interaction Resource Fact (IRF) representation of the call topology.

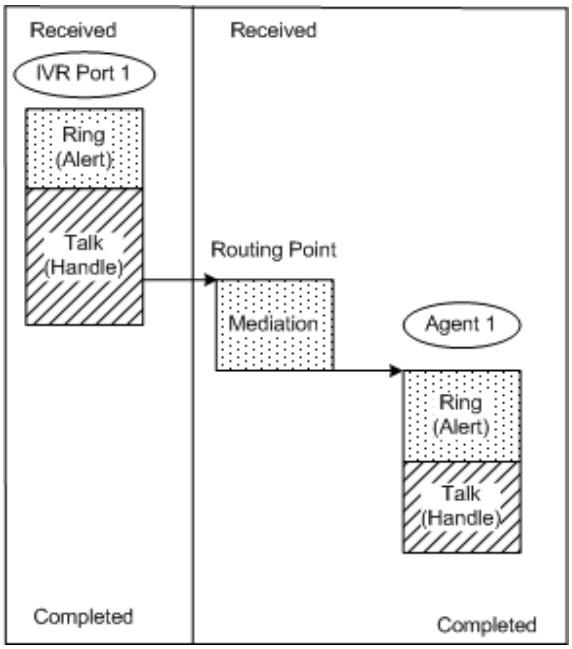

Figure 61: IVR Transfers to Agent

IVR-Behind-Switch Assisted by Universal Routing Call Flows

Voice interactions that arrive at the switch are queued to an ACD queue, in which the ACD positions are actually IVR ports. Through a virtual routing point in the premise T-Server, the IVR application invokes an Universal Routing strategy. Universal Routing instructs the IVR application to play applications or collect information and uses the collected information to return an appropriate target. The IVR application mute transfers the call to that target.

Inbound Call Flow Examples

This subsection contains the following examples of inbound call flows. Each example represents a different outcome:

- The IVR transfers the call to an ACD queue.
- The IVR transfers the call to an agent (see page 118).

IVR Transfers to ACD Queue

This call topology shows the outcome of a call that an IVR transfers to an ACD queue, in accordance with routing instructions. The call arrives at an ACD queue and is diverted to an IVR port. The IVR requests routing instructions from a virtual routing point, and then mute transfers the call to another ACD queue.

- Figure 62 depicts the Interaction Resource Fact (IRF) representation of the call topology in the case of a *self-service* (SS) IVR (when the IVR is in its own box).
- Figure 63 on page 118 depicts the Interaction Resource Fact (IRF) representation of the call topology in the case of a *nonself-service* (nonSS) IVR.

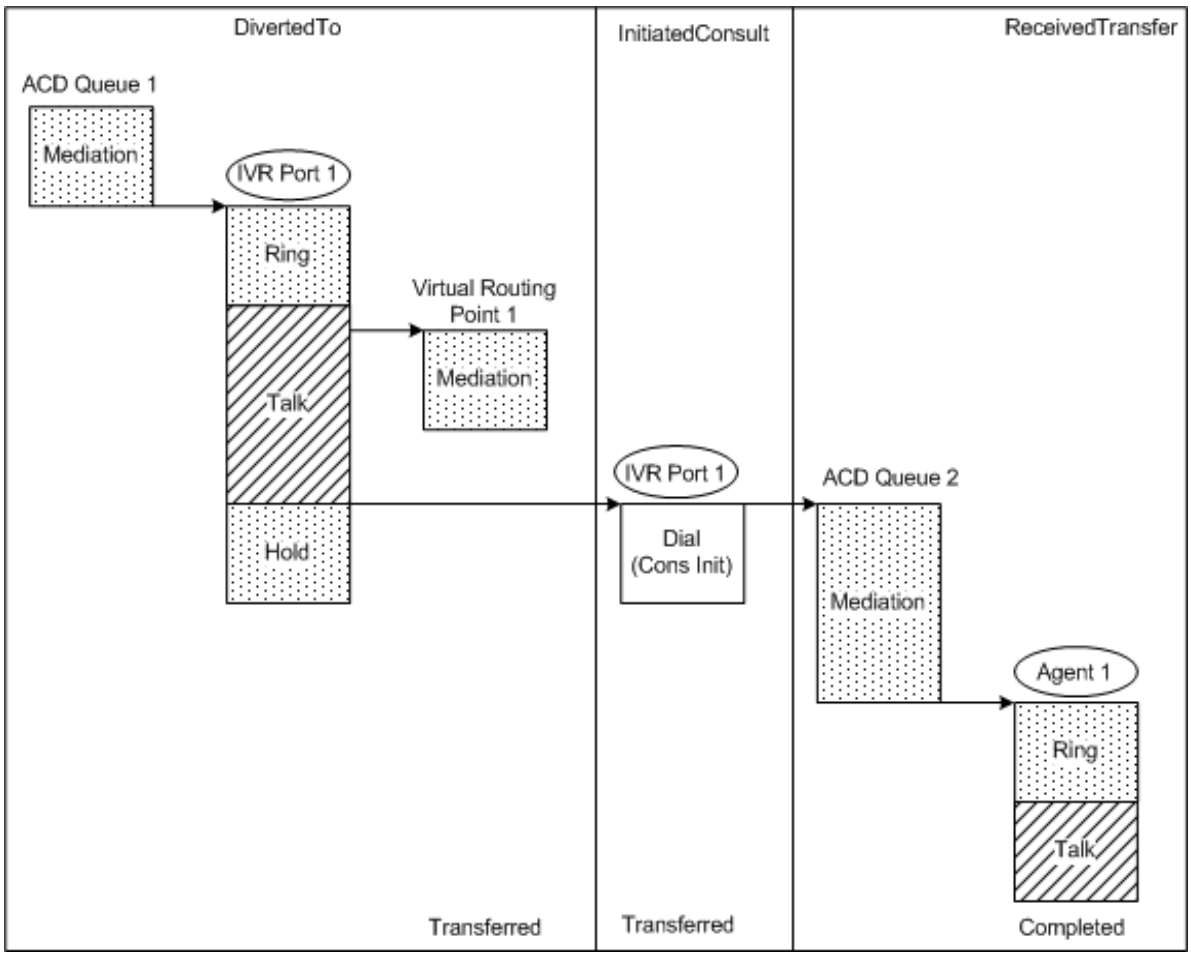

Figure 62: IVR Transfers to ACD Queue (SS IVR)

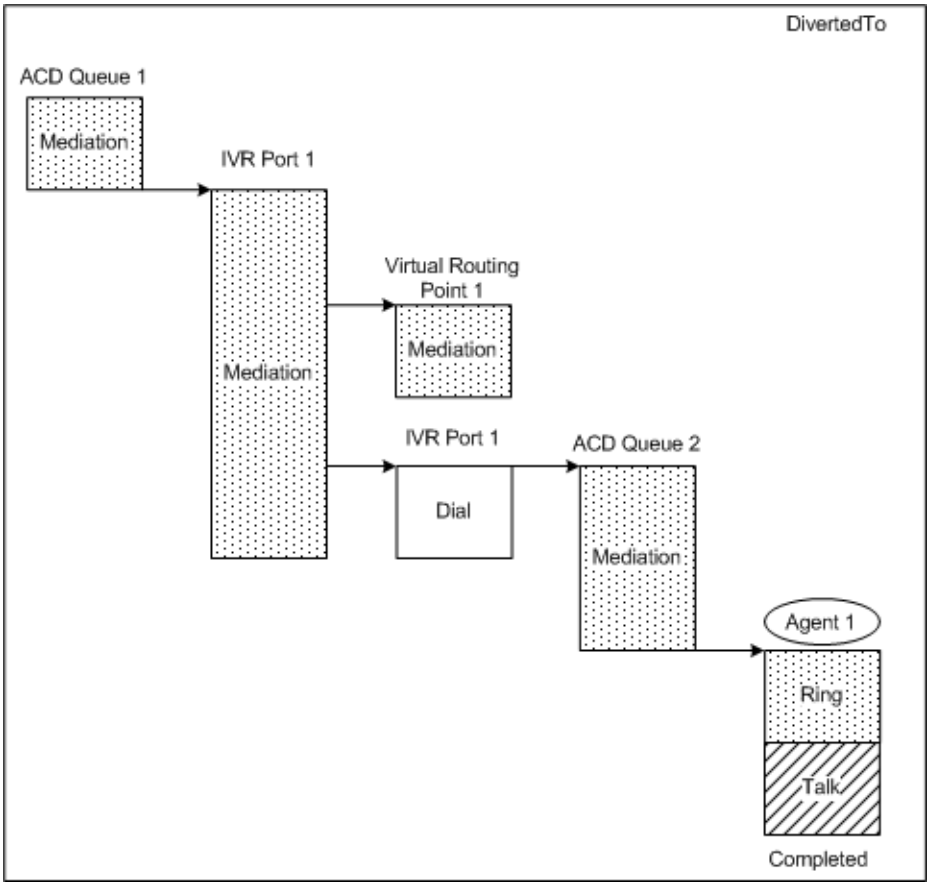

Figure 63: IVR Transfers to ACD Queue (nonSS IVR)

IVR Transfers to Agent

This call topology shows the outcome of a call that an IVR transfers directly to an agent, in accordance with routing instructions. The call arrives at an ACD queue and is diverted to an IVR port. The IVR requests routing instructions from a virtual routing point, and then mute transfers the call to an agent.

- Figure 64 on page 119 depicts the Interaction Resource Fact (IRF) representation of the call topology in the case of an SS IVR (when the IVR is in its own box).
- Figure 65 on page 120 depicts the Interaction Resource Fact (IRF) representation of the call topology in the case of a nonSS IVR.

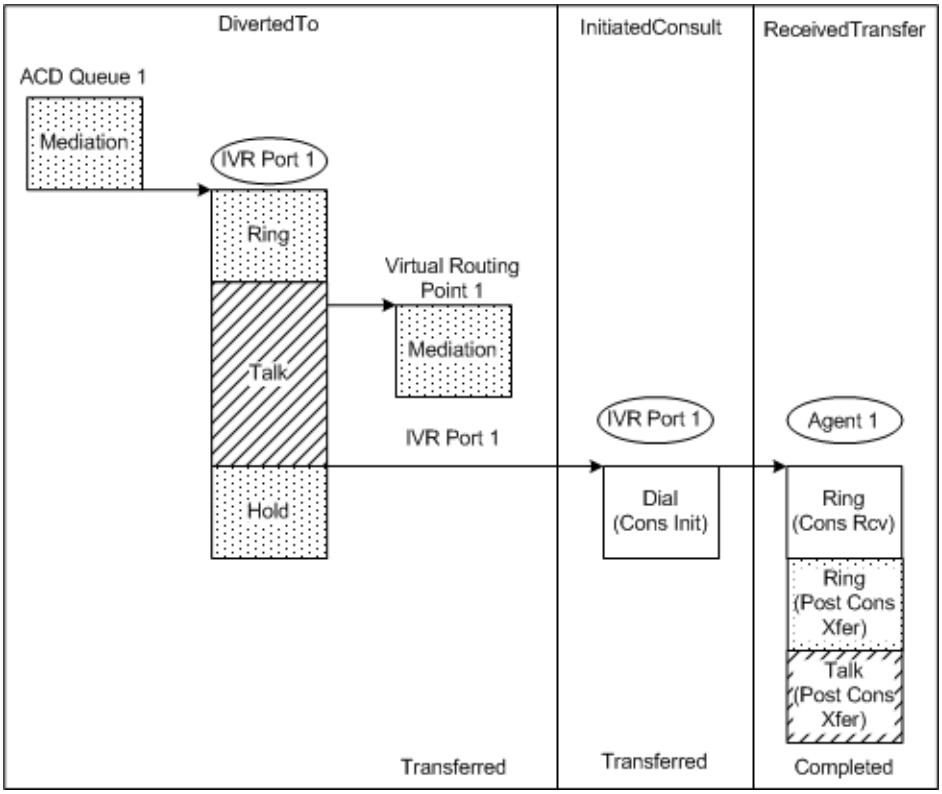

Figure 64: IVR Transfers to Agent (SS IVR)

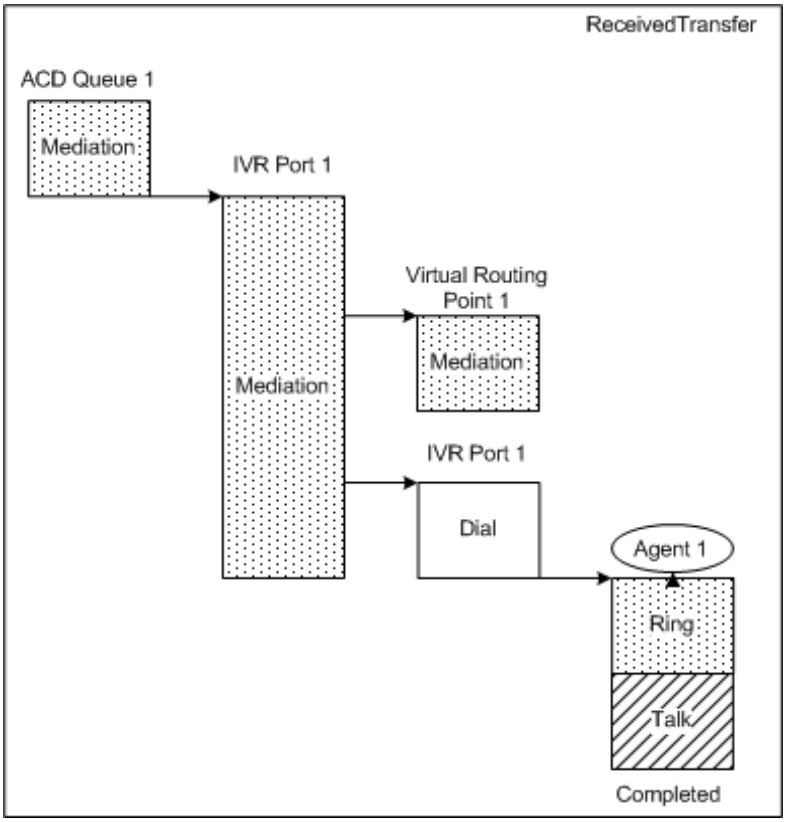

Figure 65: IVR Transfers to Agent (nonSS IVR)

Chapter

4 Validated Multimedia Interaction Flows

This chapter describes the recognized, validated multimedia interactions that have been tested, and that are supported by Genesys Info Mart.

This chapter provides detailed sections that discuss chat and e-mail. However, Genesys Info Mart supports full processing of any 3rd Party Media interactions, in addition to e-mail and chat interactions.

Use the sections that discuss e-mail as a guide to interactions that do not involve an online session with a customer (offline interactions), and the chat section as a guide to interactions that do involve an online session with a customer (online interactions).

The interaction flows described in this chapter are intended as examples that you can modify for your environment. However, Genesys does not guarantee results for modified interaction flows.

This chapter contains the following sections:

Diagram Conventions

The flow diagrams in this chapter use the same conventions as the call flow diagrams in Chapter 3 (see "Diagram Conventions" on page 65).

Additional Convention

Terms such as *Ring* and *Talk* are used generically. In multimedia interaction flows these terms indicate the *Alerting* and *Connected* conditions.

Like the voice interaction flows in Chapter 3, these diagrams are based on data from the `INTERACTION_RESOURCE_FACT` (IRF) table. The call flows represent the resources that handle interactions and the resources' states.

The following list points out features specific to the diagrams:

- The circled resource in the diagram represents the resource.

- The rows are separated by vertical lines.
- The resources of interest are handling resources, which are those having the greatest interest for reporting. Primarily, these are agents, but may also include routing strategies which become handling resources when sending an Autoresponse. Additionally, when an interaction is abandoned in an Interaction Queue or a Routing Strategy, the resource where it was abandoned is represented with an IRF entry.

E-Mail Interactions

This section contains several examples of e-mail flows. Each example represents a different outcome:

- A routing strategy routes the e-mail interaction to an agent, and the agent replies (see page 123).
- A routing strategy routes the e-mail interaction to an agent, but the agent does not accept the invitation (see page 124).
- An incoming e-mail interaction is handled by a routing strategy with an autoresponse (see page 124).
- A routing strategy routes the e-mail interaction to an agent, who transfers it directly to another agent (see page 125).
- A routing strategy routes the e-mail interaction to an agent, who unsuccessfully attempts to transfer it directly to another agent (see page 126).
- An e-mail interaction is routed to an agent, who transfers the e-mail through a Queue to another agent, who replies to the e-mail (see page 128).
- An agent unsuccessfully attempts to transfer an e-mail interaction to another agent through a queue (see page 128).
- A routing strategy routes the e-mail interaction to an agent, who consults to another agent before sending a reply (see page 129).
- A routing strategy routes the e-mail interaction to an agent, who unsuccessfully attempts to consult to another agent before sending a reply (see page 131).
- An agent saves a draft of the e-mail reply, and then later completes and sends it (see page 132).
- An agent pulls the e-mail interaction from an interaction queue or workbin (see page 133).
- Multipart reply (see page 133).
- Agent(s) create multiple replies to an inbound e-mail, one of which is stopped without being sent (see page 134).

Strategy Routes E-Mail to Agent, and Agent Replies

Figure 66 shows the outcome of an e-mail interaction that a routing strategy routes to an agent, who accepts the invitation. This interaction flow begins in Mediation, while the interaction waits in Interaction Queue 1, and while Strategy 1 attempts to find a routing target. The interaction is routed to Agent 1. Agent 1 accepts the invitation into the interaction, and creates an outbound reply e-mail, closing the original inbound e-mail. The outbound e-mail is placed into Interaction Queue 2, from which it will either be sent out of the contact center to the customer, or perhaps first to a supervisor for a quality review.

Note: If Interaction Queue 2 and Strategy 2 are an Outbound Queue and a Outbound Strategy, used simply to send the reply out of the Contact Center, then the Result for Agent 1's Reply IRF is `Completed`; otherwise it is `Transferred`.

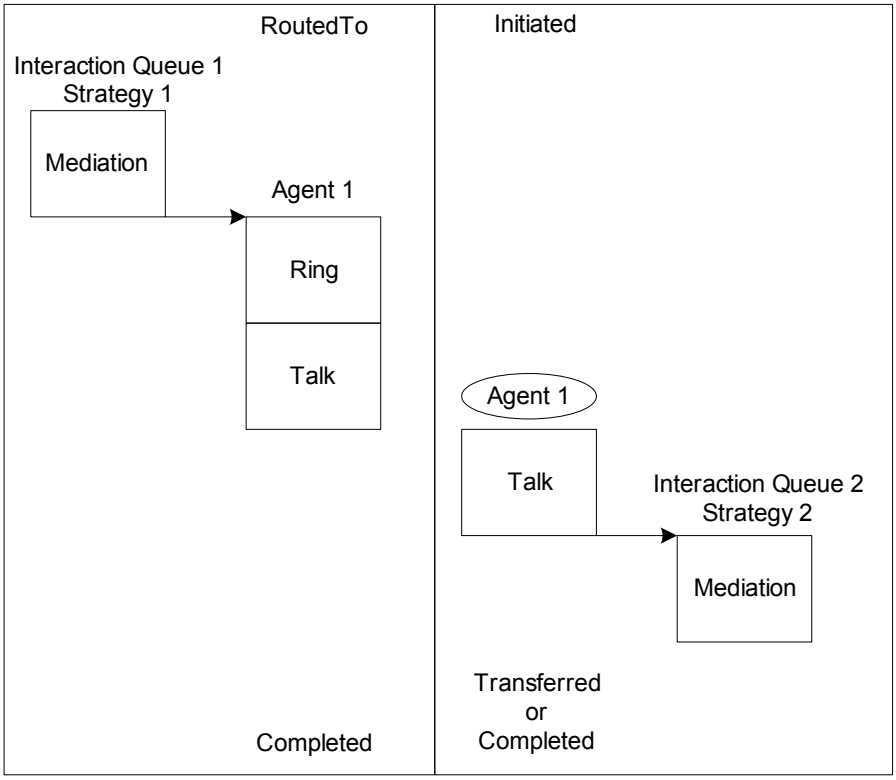

Figure 66: Strategy Routes E-Mail to Agent, and Agent Replies

Agent Invited into E-Mail Interaction, and Invitation Revoked

Figure 67 shows the outcome when an e-mail interaction is offered to an agent by a routing strategy, but the agent does not accept the invitation. This interaction flow begins in Mediation, while the interaction waits in Interaction Queue 1, and while Strategy 1 attempts to find a routing target. The interaction is routed to Agent 1. Agent 1 does *not* accept the invitation into the interaction. The e-mail interaction is returned to the interaction queue so that it can be reprocessed.

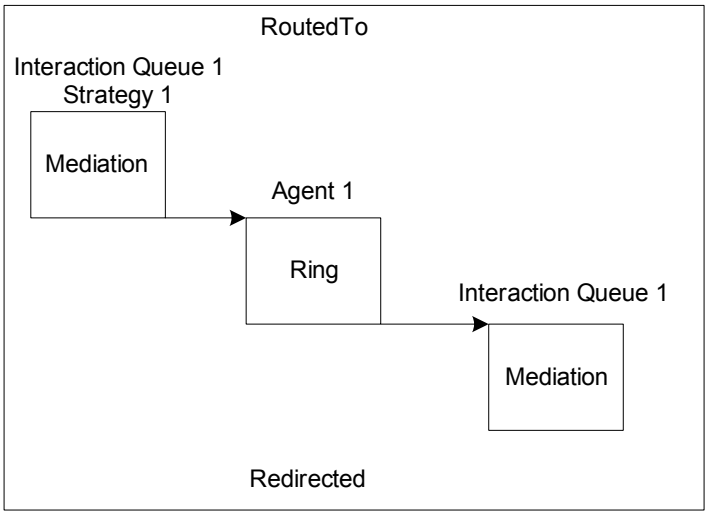

Figure 67: Agent Does Not Accept E-Mail Invitation

Note: When a routing strategy routes to an agent, the strategy is removed from the interaction as soon as the agent is invited into that interaction. In other words, the routing is complete as soon as the agent is invited.

E-Mail Interaction Handled by a Strategy with Autoresponse

Figure 68 shows the outcome of an e-mail interaction that a routing strategy determines can be handled with an Autoresponse. The e-mail is submitted to an inbound interaction queue. The routing strategy pulls the e-mail from the interaction queue, and identifies that it requires an Autoresponse, and an Autoresponse is generated. Strategy 1 is now considered a handling resource. Strategy 1 connected to and stopped the original inbound interaction, represented in the first IRF. Strategy 1 then created an outbound Autoresponse reply, which is represented in the second IRF. The time that Strategy 1 is connected to each e-mail is represented as “Talk” time. Outbound Queue and

Outbound Strategy represent the processing that occurs when sending an e-mail outside the contact center.

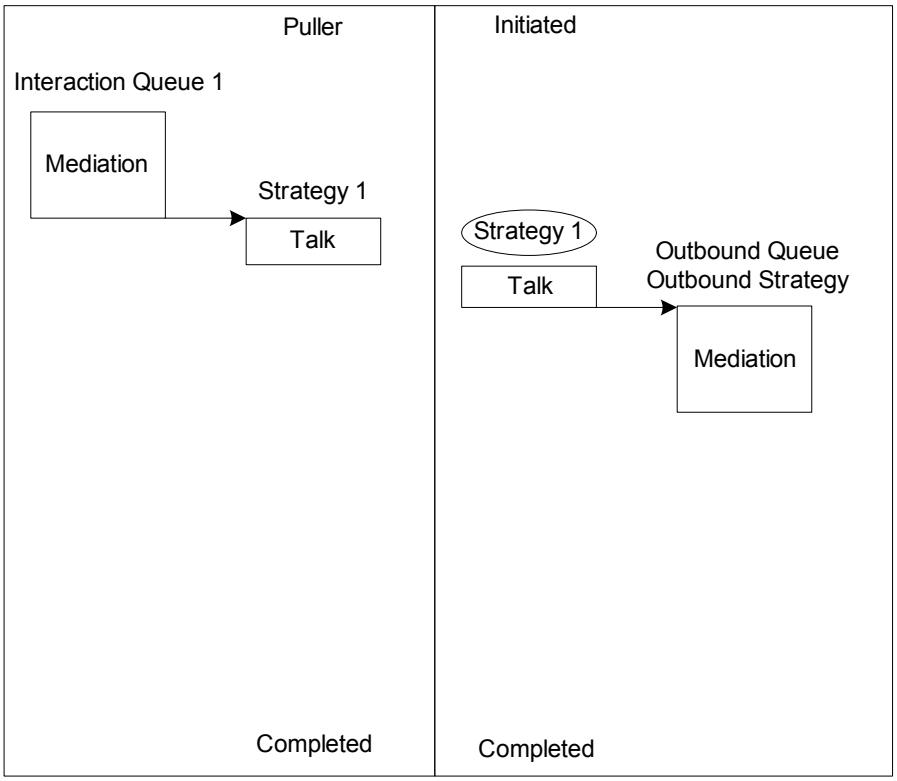

Figure 68: E-Mail Handled by a Strategy with Autoresponse

Agent Transfers E-Mail Directly to Another Agent

Figure 69 e-mail shows the outcome of an e-mail interaction that is routed to an agent, who transfers the e-mail to another agent, who replies to the e-mail.

Agent 1 transfers an inbound interaction to Agent 2. Agent 2 stops the original inbound interaction while creating an outbound reply. The outbound reply is placed into Interaction Queue 2, from which it is sent either out of the contact center to the customer, or perhaps sent first to a supervisor for a quality review.

When an agent directly invites another agent into an interaction, the original agent remains in the interaction until the target agent accepts the invitation. In the case of a transfer, the transfer does not occur until the target agent accepts the invitation.

Note: In this scenario, the original inbound e-mail is transferred. Another scenario presents a variation in which Agent 1 creates an outbound reply and transfers the reply to Agent 2.

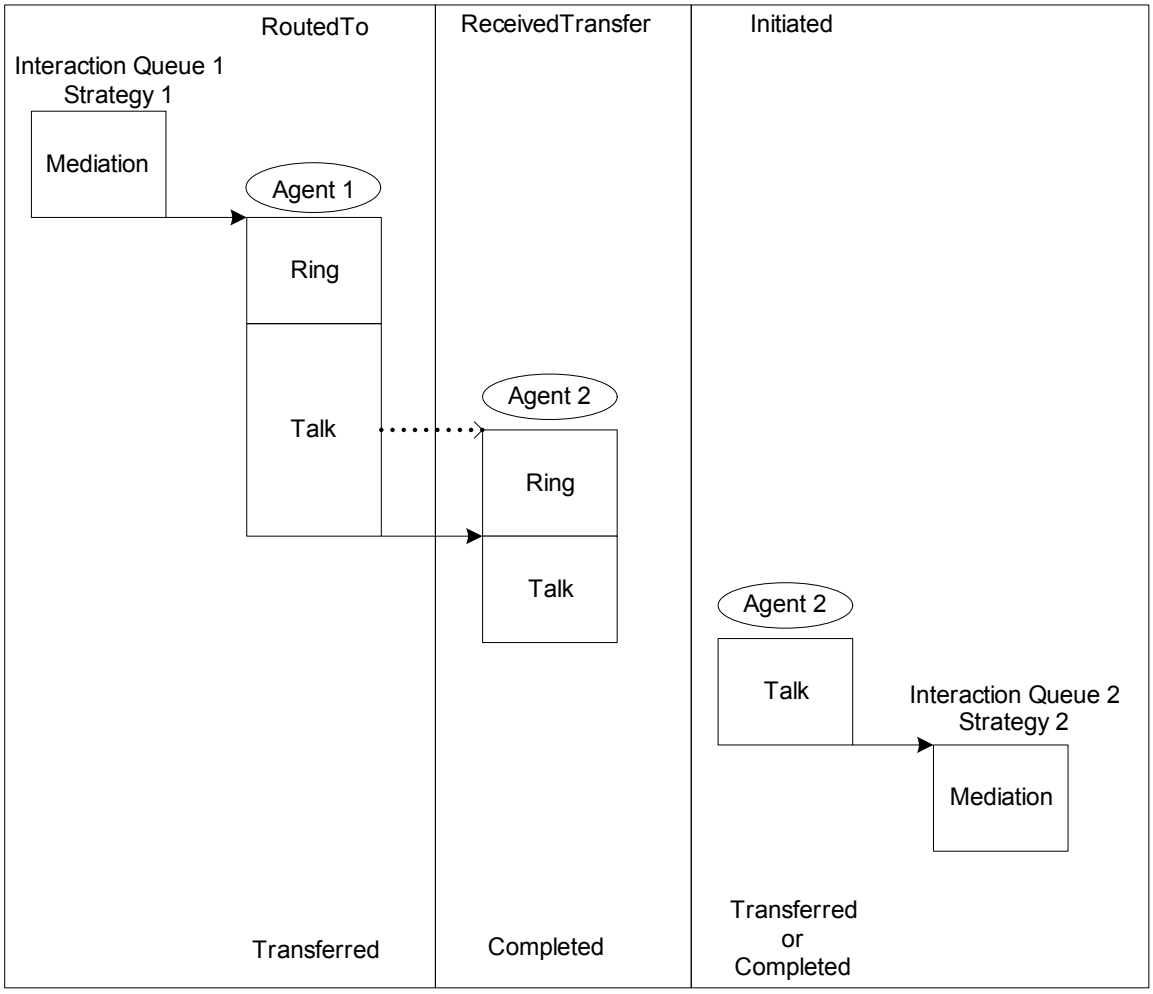

Figure 69: Agent Transfers E-Mail to Another Agent

Agent's Attempt to Transfer E-Mail Directly to Another Agent Fails

Figure 70 shows the outcome of an unsuccessful attempt to transfer an e-mail to another agent.

The interaction is routed to Agent 1. Agent 1 accepts the inbound e-mail and creates an outbound reply, closing the original inbound e-mail. Agent 1 works on this reply, and then attempts to transfer this reply to Agent 2, for Agent 2 to complete the reply. Agent 2 does not accept the invitation into the interaction. Agent 1 remains in the interaction during the attempt to transfer. In this case, since Agent 2 was not available, Agent 1completes the reply, and places it into

Interaction Queue 2, from which it is sent either out of the contact center to the customer, or perhaps first to a supervisor for a quality review.

Note: If Interaction Queue 2 and Strategy 2 are an Outbound Queue and a Outbound Strategy which simply are used to send the reply out of the Contact Center, then the `Result` for Agent 1's reply is `Completed`; otherwise it is `Transferred`.

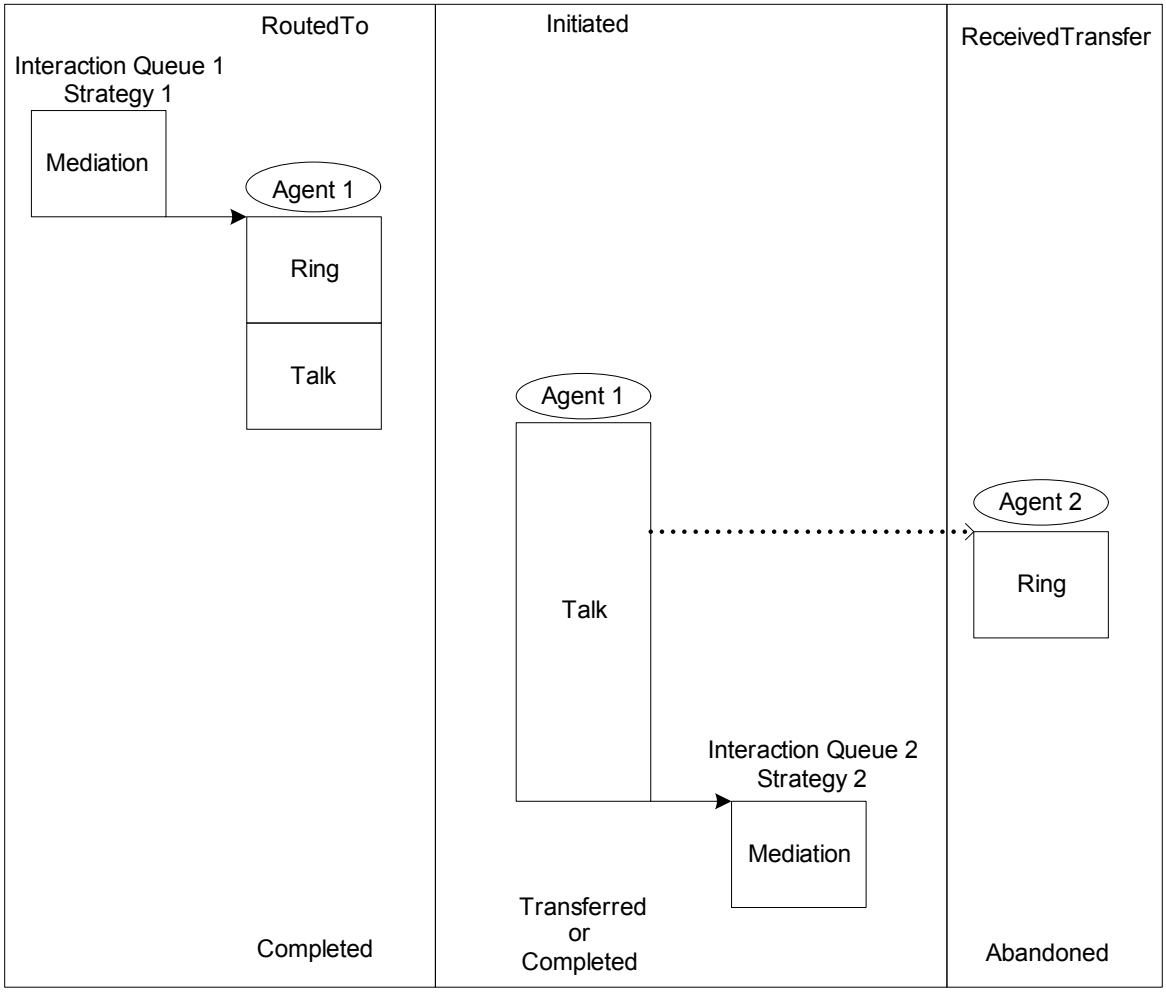

Figure 70: Unsuccessful Direct Agent-to-Agent E-Mail Transfer Attempt

Transfer of E-Mail from Agent to Agent through a Queue

Figure 71 shows the outcome of an e-mail interaction that is routed to an agent, who transfers the e-mail through a Queue to another agent, who replies to the e-mail.

Agent 1 transfers an inbound interaction through a Queue to Agent 2. Agent 2 stops the original inbound interaction while creating an outbound reply. The outbound reply is placed into Interaction Queue 3, from which it will either be sent out of the contact center to the customer (shown with a result of `Completed`), or perhaps to another agent for additional processing, such as to a supervisor for a quality review (shown with a result of `Transferred`).

Note: In this scenario, the original inbound e-mail is transferred. Another scenario presents a variation where Agent 1 creates an outbound reply, and transfers the reply.

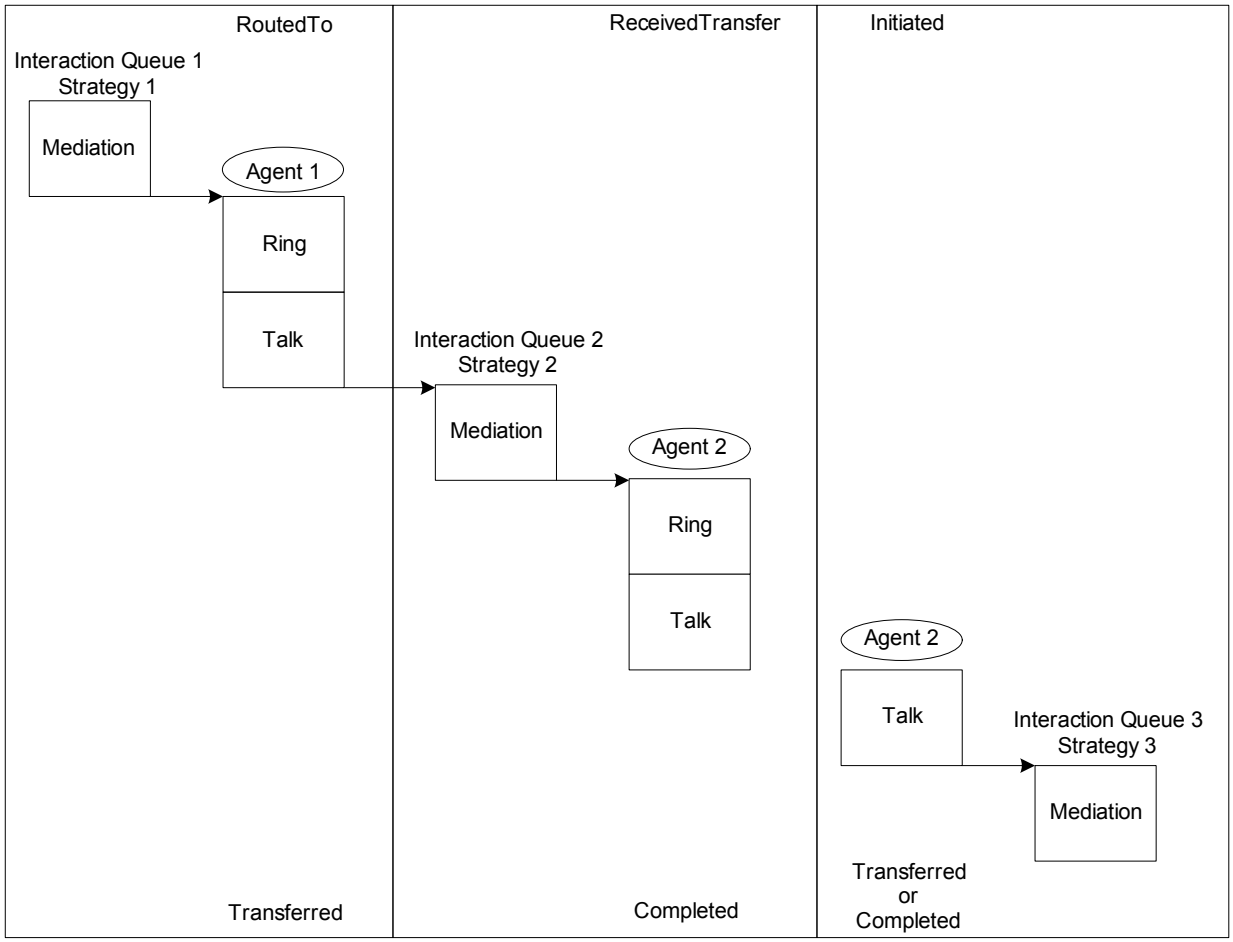

Figure 71: Transfer of E-Mail from Agent to Agent through a Queue

Unsuccessful Transfer from Agent to Agent through a Queue

Figure 72 shows the outcome of an unsuccessful attempt to transfer an e-mail interaction from one agent to another agent through a queue.

The interaction is routed to Agent 1. Agent 1 accepts the inbound e-mail and create an outbound reply, closing the original inbound e-mail. Agent 1 works on this reply, and then attempts to transfer the reply interaction through a queue to another agent for continued processing. The reply interaction is routed from the queue to Agent 2. Agent 2 does not accept the invitation, and this revoked invitation is returned back to Interaction Queue 2.

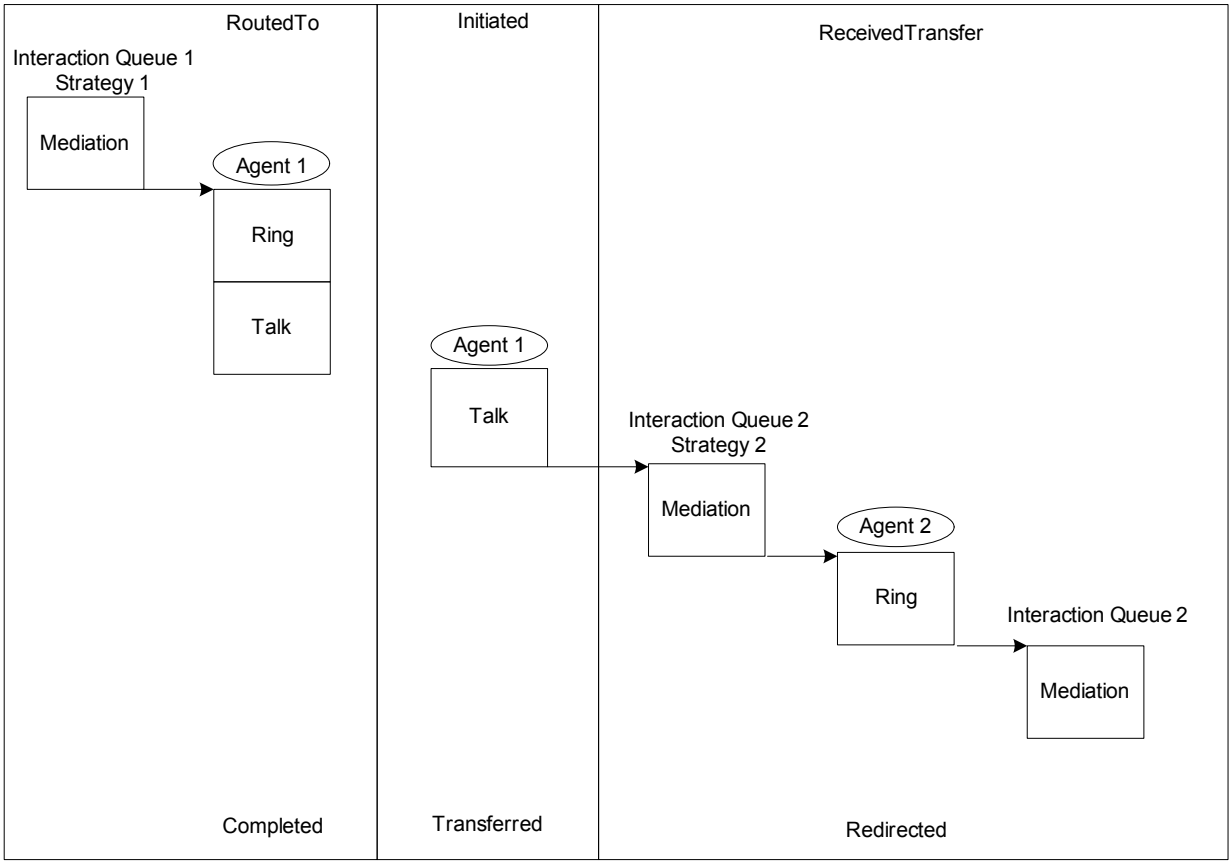

Figure 72: Unsuccessful Transfer from Agent to Agent through a Queue

Agent Consults to Another Agent Before Sending Reply

Figure 73 shows an Agent consulting with another Agent before sending a reply.

The interaction is routed to Agent 1. Agent 1 accepts the inbound e-mail and creates an outbound reply, closing the original inbound e-mail. Agent 1 initiates a Consult interaction and transfers it to Agent 2. Agent 2 accepts the Consult interaction, and initiates a Consult Reply interaction. In a typical Consult scenario, the Consult Reply is placed into Agent 1's Collaboration

Workbin. It typically remains there for the life of the entire interaction, enabling the Consult Reply to be viewed any time during the processing of the interaction, and making it available for viewing by another agent, if Agent 1 transfers ownership of this interaction to another Agent.

Note: While the Consult Reply remains in the Collaboration Workbin, the Technical Result is `Transferred`. When the Consult Reply is pulled from the Collaboration Workbin and closed (usually not until the end of the entire Interaction), a new row is added showing the Consult Reply being pulled from the Workbin, and `Completed`.

The outbound reply is placed into an Interaction Queue, from which it is either sent out of the contact center to the customer (shown with a `Result` of `Completed`), or perhaps to another agent for additional processing, such as to a supervisor for a quality review (shown with a `Result` of `Transferred`).

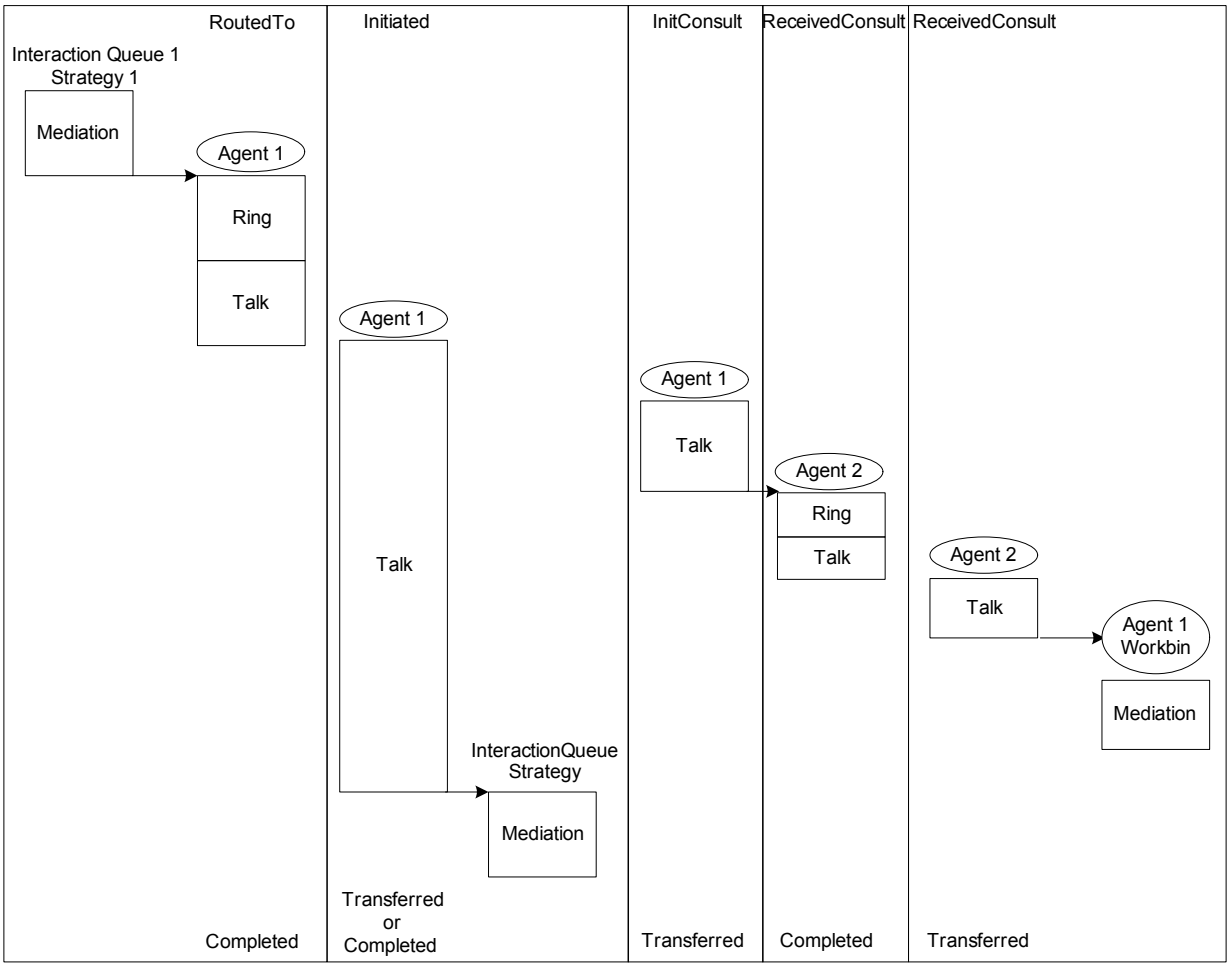

Figure 73: Agent Consults to Another Agent Before Sending Reply

Agent Unsuccessfully Consults to Another Agent Before Sending Reply

Figure 74 shows an Agent's unsuccessful attempt to consult with another Agent before sending a reply.

The interaction is routed to Agent 1. Agent 1 accepts the inbound e-mail and creates an outbound reply, closing the original inbound e-mail. Agent 1 initiates a Consult interaction and transfers it to Agent 2. Agent 2 does not accept the invitation into the Consult interaction. Agent 1 continues working on the outbound reply. The outbound reply is placed into an Interaction Queue, from which it is either sent out of the contact center to the customer (shown with a `Result` of `Completed`), or perhaps to another agent for additional processing, such as to a supervisor for a quality review (shown with a `Result` of `Transferred`).

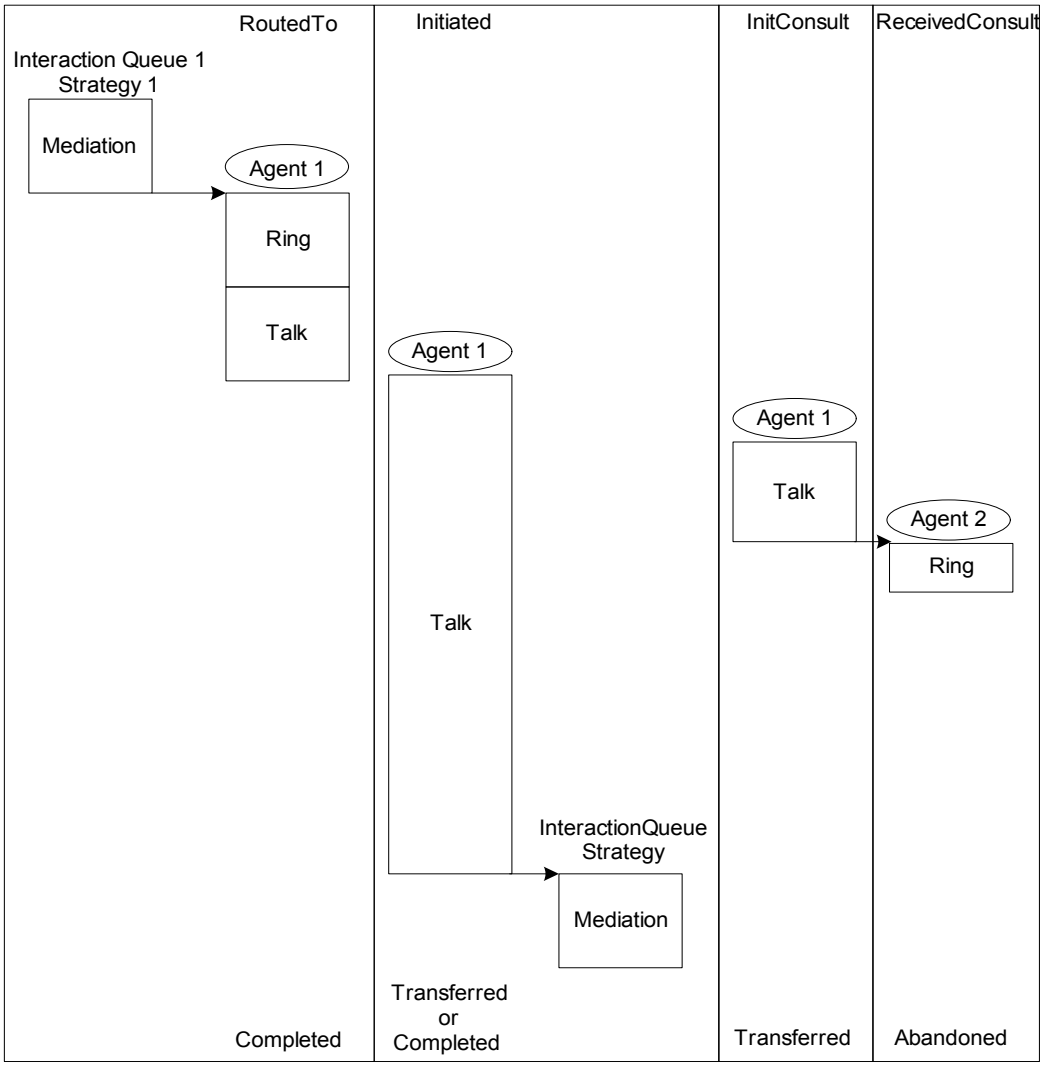

Figure 74: Agent Unsuccessfully Consults to Another Agent Before Sending Reply

Agent Saves Draft Reply Before Sending

Figure 75 shows the outcome of an e-mail interaction that is routed to an agent who replies to the e-mail, after first saving an initial version of the reply in a workbin.

This interaction flow begins in Mediation, while the interaction waits in Interaction Queue 1 and while Strategy 1 attempts to find a routing target. The interaction is routed to Agent 1.Agent 1accepts the invitation into the interaction, and creates an outbound reply e-mail, closing the original inbound e-mail. The outbound reply e-mail is saved by Agent 1 in a Draft Workbin. Later, Agent 1 pulls the reply e-mail from the Workbin, making some final modifications to the reply, and then places the reply into Interaction Queue 2, from which it will either be sent out of the contact center to the customer (shown with a result of `Completed`), or perhaps to another agent for additional processing, such as to a supervisor for a quality review (shown with a result of `Transferred`).

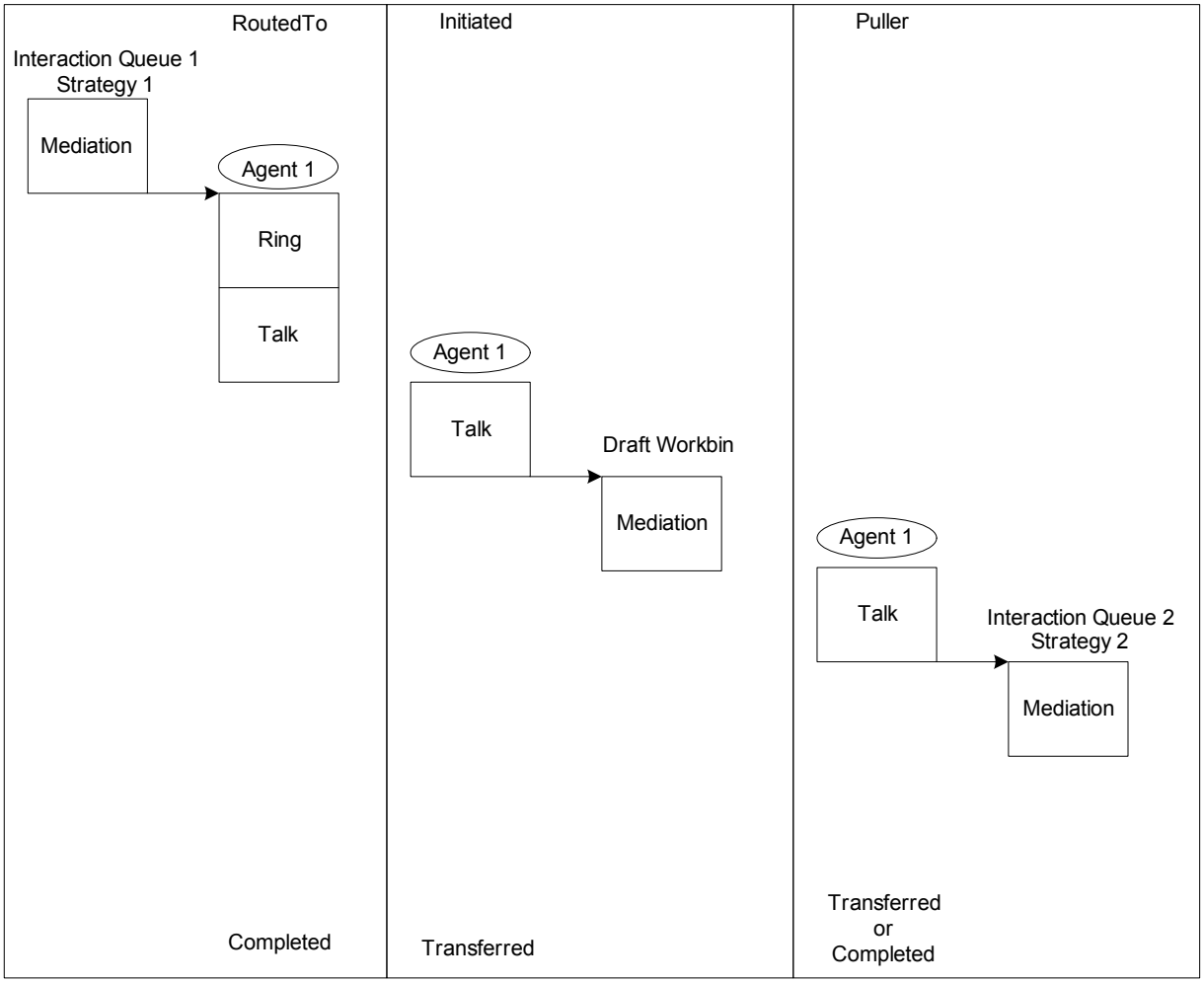

Figure 75: Agent Saves Draft Reply Before Sending

Agent Pulls E-Mail from an Interaction Queue or Workbin

Figure 76 shows the outcome when an e-mail interaction is pulled from an Interaction Queue or Workbin for further handling by Agent 1. After working on the e-mail, Agent 1 places the interaction into Interaction Queue 2. If the e-mail is an outbound e-mail that is then sent out of the contact center, Agent 1's result is shown as `Completed`; otherwise it is shown as `Transferred`.

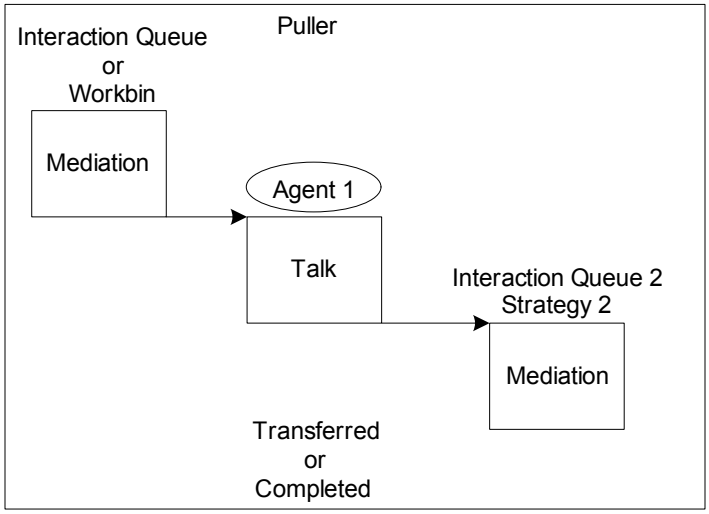

Figure 76: Agent Pulls E-Mail from Workbin

Multipart Reply

Figure 77 shows the outcome of a multipart reply.

In this example, Agent 1 initiates two outbound replies. For example, Agent 1 may have received a new customer order, and initiated one reply, which is to be completed by Agent 2 in Shipping, while also initiating a reply to the customer confirming the order and providing billing-related information. Agent 1 transfers the first outbound reply directly to Agent 2. Agent 1 remains in this outbound reply until Agent 2 accepts the invitation into the interaction. Agent 2 later completes this reply, placing it in an outbound queue, from which it is then sent out of the contact center. Agent 1also creates and completes a second outbound reply, now closing the original inbound interaction, and places this second reply in an outbound queue, from which it is sent out of the contact center. In this example, the second reply created was actually the first reply sent to the customer.

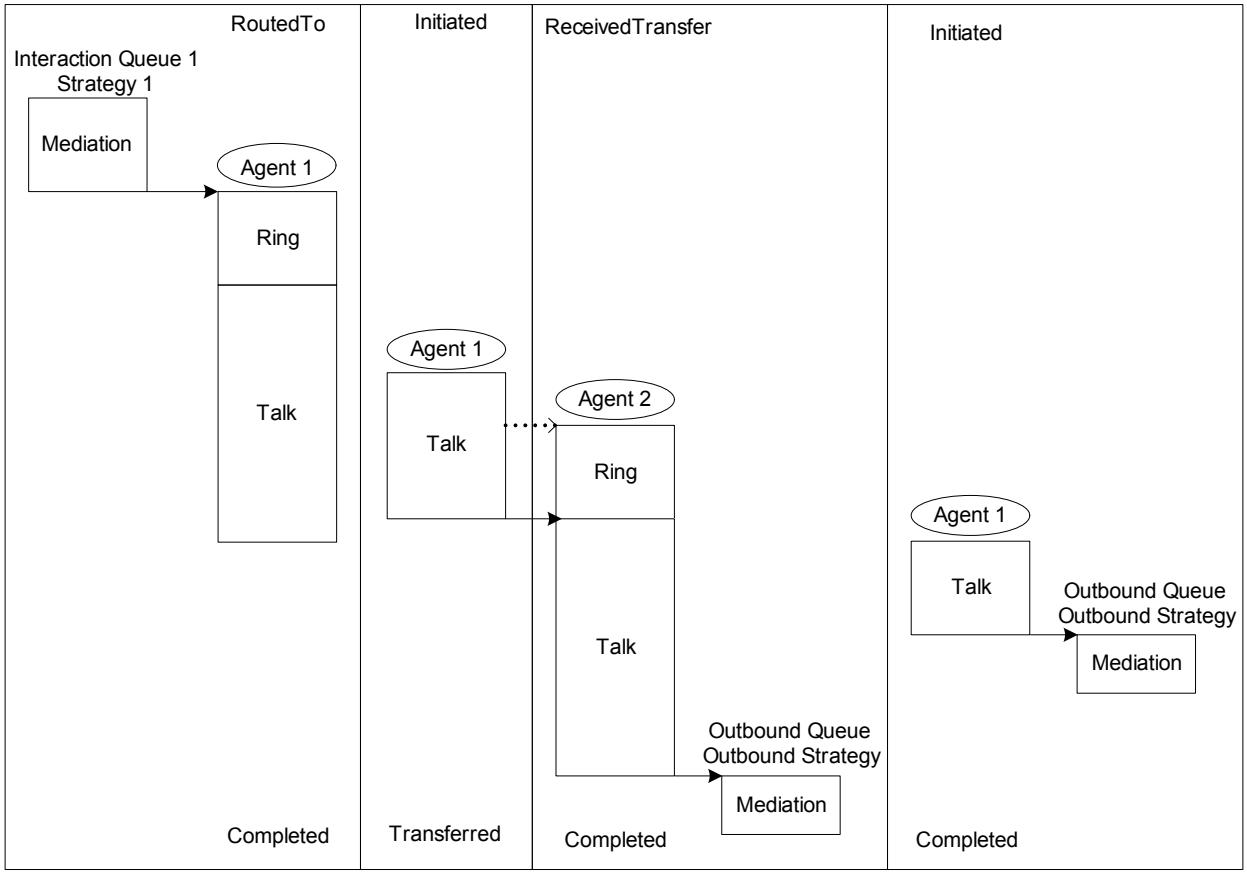

Figure 77: Agent Sends Multipart Reply

Multipart E-Mail Reply with Unsent Reply

Figure 78 on page 135 shows the outcome of a multipart reply in which one of the replies was stopped without being sent.

This example illustrates the `OutboundStopped` Technical Result. The `OutboundStopped` Technical Result applies to any outbound multimedia interaction that is stopped without being sent; it is not limited to outbound replies. In this example, the outbound reply initiated by Agent 1, and transferred to Agent 2, is stopped by Agent 2. Agent 2 worked on the outbound reply for some time, but then stopped the reply without sending it.

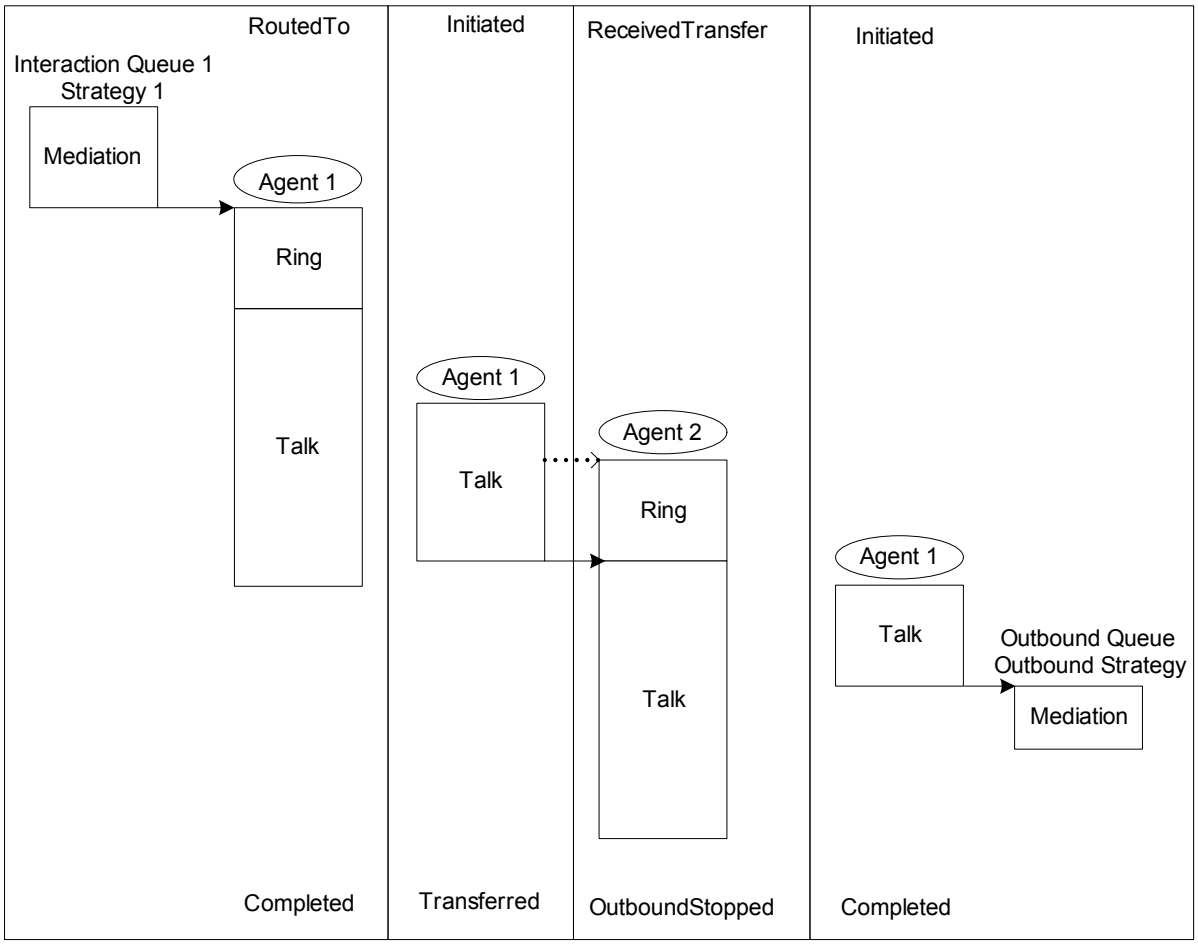

Figure 78: Multipart E-Mail Reply with Unsent Reply

Chat Interactions

This section contains several examples of chat flows. Each example represents a different outcome:

- A routing strategy routes the chat interaction to an agent, and the agent replies (see page 136).
- A routing strategy routes the chat interaction to an agent, but the agent does not accept the invitation (see page 137).
- A routing strategy routes the chat interaction to an agent, who transfers it to another agent (see page 137).
- A routing strategy routes the chat interaction to an agent, who unsuccessfully attempts to transfer it to another agent (see page 138).
- A routing strategy routes the chat interaction to an agent, who conferences in another agent (see page 139).

- An agent attempts to conference another agent in to chat with the customer, but fails (see page 140).
- The customer abandons the chat interaction while waiting in the interaction queue (see page 141).
- The customer abandons the chat interaction during routing (see page 142).
- A routing strategy routes the chat interaction to an agent, but the customer abandons the interaction while the agent was being alerted (see page 142).

Strategy Delivers Chat to Agent, and Agent Replies

Figure 79 shows the outcome of outcome of a chat interaction that a routing strategy routes to an agent, who accepts the invitation. This interaction flow begins in Mediation, while the interaction waits in Interaction Queue 1, and while Strategy 1 attempts to find a routing target. The interaction is routed to Agent 1. Agent 1 accepts the invitation into the interaction, and chats with the customer.

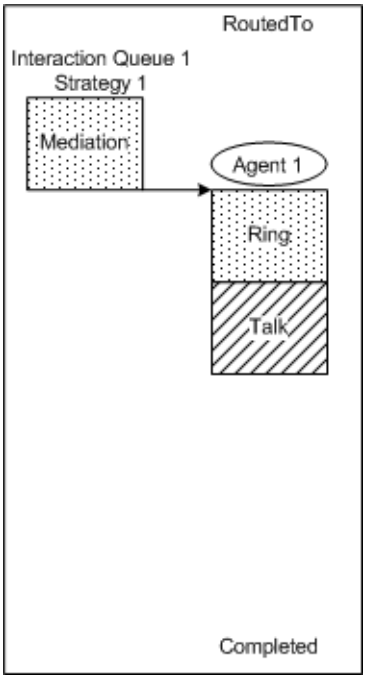

Figure 79: Strategy Routes Chat to Agent, and Agent Replies

Agent Invited into Chat and Invitation Revoked

Figure 80 shows the outcome of a chat interaction that a routing strategy routes to an agent, who does not accept the invitation. This interaction flow begins in Mediation, while the interaction waits in Interaction Queue 1, and while Strategy 1 attempts to find a routing target. The interaction is routed to Agent 1. Agent 1 does not accept the invitation into the interaction. The chat interaction is returned to the interaction queue so that it can be reprocessed.

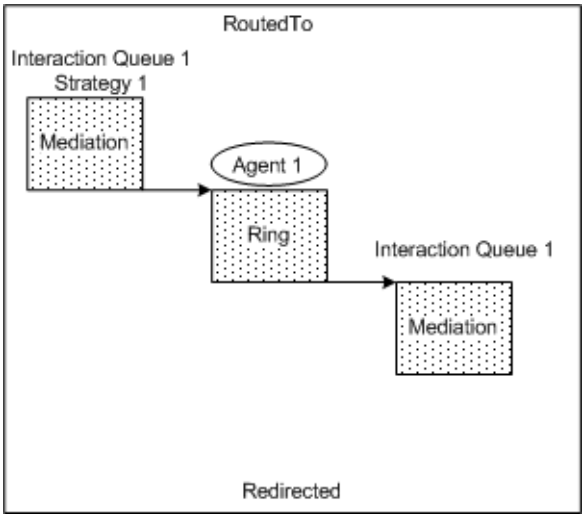

Figure 80: Agent Does Not Accept Chat Invitation

Agent Transfers Chat to Another Agent

Figure 81 shows the outcome of a chat interaction that is routed to an agent, who transfers the chat to another agent.

This interaction flow begins in Mediation, while the interaction waits in Interaction Queue 1, and while Strategy 1 attempts to find a routing target. The interaction is routed to Agent 1. Agent 1 accepts the invitation into the interaction, and chats with the customer. Agent 1 then transfers the chat interaction to Agent 2, who accepts the invitation and then chats with the customer.

Note: After Agent 1 initiates the transfer of the chat interaction to Agent 2, Agent 1 remains in the chat interaction until Agent 2 accepts the invitation. The customer does not wait for Agent 2 to connect since the customer is still chatting with Agent 1.

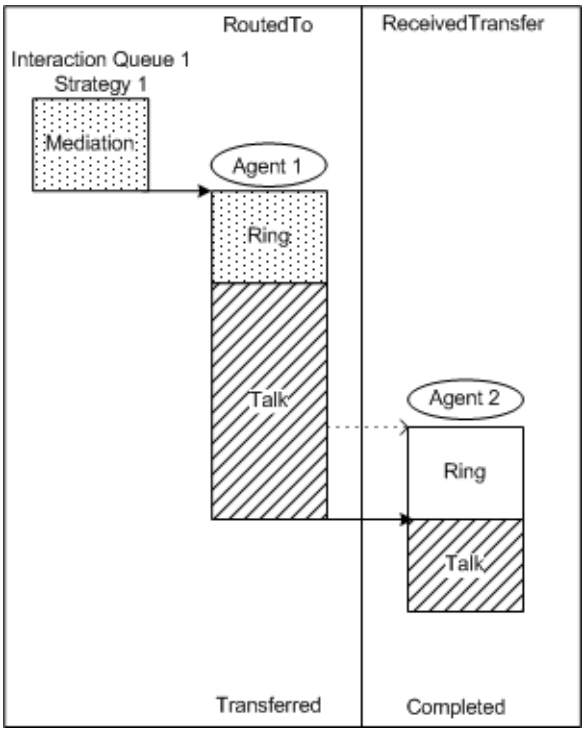

Figure 81: Agent Transfers Chat to Another Agent

Agent's Attempt to Transfer Chat to Another Agent Fails

Figure 82 shows the outcome of an unsuccessful attempt to transfer a chat interaction to another agent.

This interaction flow begins in Mediation, while the interaction waits in Interaction Queue 1, and while Strategy 1 attempts to find a routing target. The interaction is routed to Agent 1. Agent 1 accepts the invitation into the interaction, and chats with the customer. Agent 1 then attempts to transfer the chat interaction to Agent 2. Agent 2 does not accept the invitation. Agent 1 remains in the chat interaction, and in this example, completes the chat with the customer.

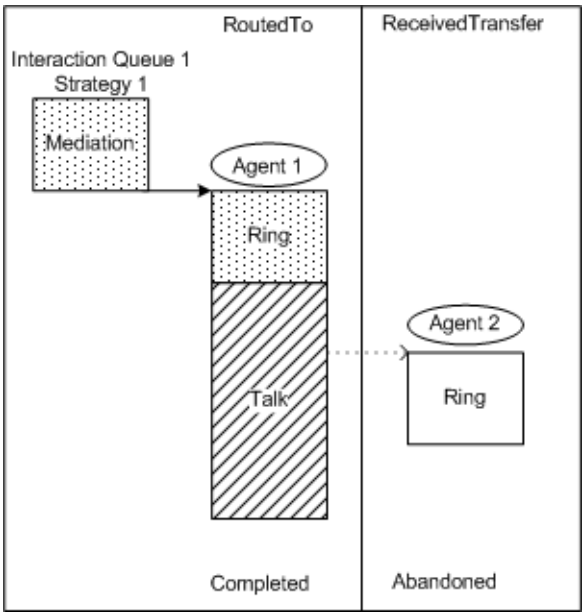

Figure 82: Unsuccessful Agent-to-Agent Chat Transfer Attempt

Agent Conferences In Another Agent

Figure 83 shows the outcome of a chat interaction that is routed to an agent, who conferences in another agent.

This interaction flow begins in Mediation, while the interaction waits in Interaction Queue 1, and while Strategy 1 attempts to find a routing target. The interaction is routed to Agent 1. Agent 1 accepts the invitation into the interaction, and chats with the customer. Agent 1 then attempts to conference in Agent 2. Agent 2 accepts the invitation and then also chats with the customer.

Note: Agent 1 remains in the chat while Agent 2 is invited into the interaction. The customer does not wait for Agent 2 to connect since the customer is still chatting with Agent 1.

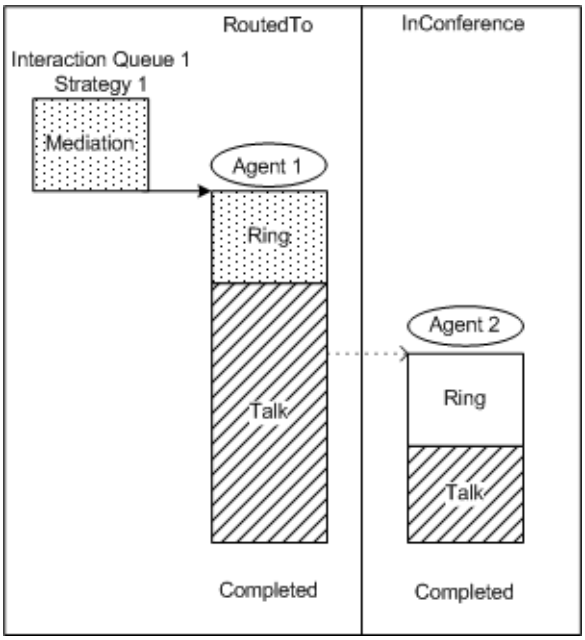

Figure 83: Agent Conferences in Another Agent

Agent's Attempt to Conference in Another Agent Fails

Figure 84 on page 141 shows the outcome of an unsuccessful attempt to conference another agent into a chat interaction.

This interaction flow begins in Mediation, while the interaction waits in Interaction Queue 1, and while Strategy 1 attempts to find a routing target. The interaction is routed to Agent 1. Agent 1 accepts the invitation into the interaction, and chats with the customer. Agent 1 then attempts to conference in Agent 2. Agent 2 does not accept the invitation. Agent 1 remains in the chat interaction, and in this example, completes the chat with the customer.

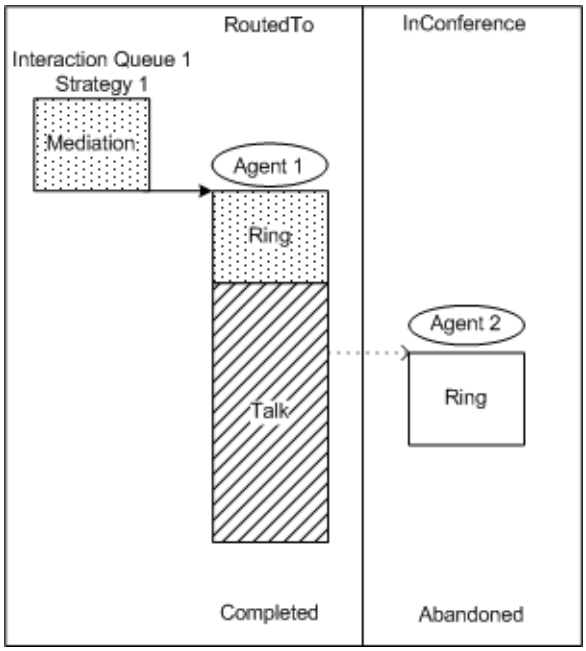

Figure 84: Agent Attempts to Conference in Another Agent. but Fails

Customer Abandons Chat in Queue

Figure 85 shows the outcome of outcome of a chat interaction that is submitted to an inbound interaction queue, but is abandoned by the customer while it is in the interaction queue.

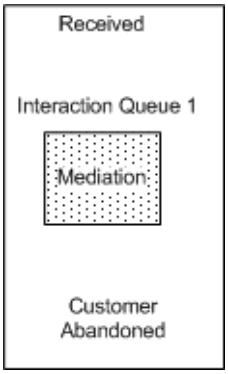

Figure 85: Customer Abandons Chat in Queue

Customer Abandons Chat During Routing

Figure 86 shows the outcome of outcome of a chat interaction that is submitted to an inbound interaction queue, but is abandoned by the customer while a routing strategy is attempting to route the interaction.

The Resource Role of `Puller` is used because the strategy pulled the interaction from the interaction queue.

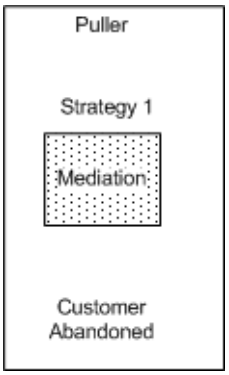

Figure 86: Customer Abandons Chat During Routing

Customer Abandons Chat During Agent Alerting

Figure 87 shows the outcome of outcome of a chat interaction that is abandoned while the agent is being alerted.

This interaction flow begins in Mediation, while the interaction waits in Interaction Queue 1, and while Strategy 1 attempts to find a routing target. The interaction is routed to Agent 1. The customer abandons the chat interaction before the agent accepts the invitation.

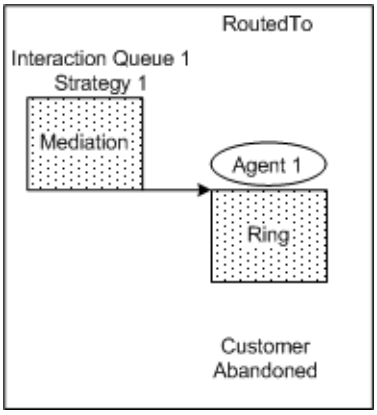

Figure 87: Customer Abandons Chat During Agent Alerting

Chapter

5 Data Lineage: Voice of Data

This chapter describes how Genesys Info Mart tracks data, enabling you to understand what data is collected and how it is processed. This information can be used for reporting and to assess data accuracy (data validation). This chapter contains the following sections:

Data Lineage Overview

Data lineage provides information that records history of job execution and data transform for each piece of data. Data stored as part of data lineage allows for bi-directional data tracking and enables you to answer the following questions:

- What process created the piece of data?
- What was the source of the data?
- What data was created by a specified job?
- What data in the system was created based on specified source data?

Data lineage has two aspects:

- **Voice of data**—This feature pertains to data quality validation and troubleshooting. It enables you to trace a particular data item in a source system based on data in the target system, and also to trace data in the opposite direction (from source to target).
- **Voice of process**—Provides data processing history, and traces which ETL process created this piece of data. It also traces in the opposite direction (from process to data).

Note: This chapter focuses on voice of data. For information about voice of process, see the *Genesys Info Mart 8.0 Operations Guide*. For detailed descriptions of the tables and views related to Data Lineage, see the *Genesys Info Mart 8.0 Reference Manual* for your RBDMS.

Voice of Data

Voice of data functionality enables you to trace data origins or targets.

To use voice of data, you must be familiar with which GIDB tables provide which kind of data. This information, called *static mapping*, cannot be derived from the schema. The connections are presented in Table 7 and discussed in more detail below.

Table 7: Data Mapping

Genesys Info Mart		GIDB	
Table	**Field**	**Table**	**Field**
INTERACTION_FACT	MEDIA_SERVER_IXN_GUID	GIDB_G_CALL_V, GIDB_G_CALL_MM	CALLID
INTERACTION_RESOURCE_FACT	INTERACTION_RESOURCE_ID[a]	GIDB_G_PARTY_V, GIDB_G_PARTY_MM	PARTY_KEY
INTERACTION_RESOURCE_FACT	PARTYGUID	GIDB_G_PARTY_V, GIDB_G_PARTY_MM	PARTYGUID
MEDIATION_SEGMENT_FACT	MEDIA_SERVER_IXN_GUID	GIDB_G_CALL_V, GIDB_G_CALL_MM	CALLID
	MEDIATION_GUID	GIDB_G_PARTY_V, GIDB_G_PARTY_MM GIDB_VIRTUAL_QUEUE_V, GIDB_VIRTUAL_QUEUE_MM	PARTYGUID VQID
RESOURCE_	RESOURCE_CFG_DBID, RESOURCE_CFG_TYPE_ID	GIDB_GC_AGENT, GIDB_GC_ENDPOINT, GIDB_GC_SCRIPT	ID (dbid)
GROUP_	GROUP_CFG_DBID	GIDB_GC_GROUP	ID

Table 7: Data Mapping (Continued)

Genesys Info Mart		GIDB	
PLACE	PLACE_CFG_DBID	GIDB_GC_PLACE	ID
SKILL	SKILL_CFG_DBID	GIDB_GC_SKILL	ID

a. The PARTY_KEY of the last party in the Interaction Resource Fact record is used as the INTERACTION_RESOURCE_ID. Primary keys can match in many-to-one relationship.

The information in this table is a starting point that enables you to:

- Trace a record back to the source records that triggered its creation, or trace data from its origin to its final target.
- Select an interaction in Genesys Info Mart and pull out from ICON database all corresponding records.
- Trace the information in these records back to the source application logs (for example, the T-Server, SIP Server, or Interaction Server logs).

How to Use Voice of Data

The basic concept of Voice of Data is that each interaction can be traced from its initial entry into your environment through its conclusion by way of specific linking IDs that enable you to trace, in either direction, the processing records from IDB to the Genesys Info Mart database via the GIDB.

To accomplish this, each of the interaction records in the Info Mart Fact tables are traceable back to source table in GIDB using keys that indicate for a particular record from which records in the source tables it was created. Multiple links enable you to trace the interaction records in the Fact tables to the GIDB, from where we can find records in ICON databases and application logs.

For example, the INTERACTION_RESOURCE_FACT table stores the PARTYGUID of the handling party. A simple join between GIDB_G_PARTY with either G_PARTY or PARTYGUID links the INTERACTION _RESOURCE_FACT table with the corresponding party record in GIDB/IDB.

Use Cases

The examples in this section show some specific ways to use Voice of Data. These provide only a small sampling of the sorts of questions you can answer using Voice of Data.

Identify Source of GIDB Data

"How do I know where specific GIDB data comes from?"

To identify the source for GIDB data, join the GIDB table of interest with the `CTL_DS` table using `DATA_SOURCE_KEY, CTL_DS`. The value for `DS_DBID` in the resulting data set provides the value of the data source DBID (for example DBID of the T-Server for voice call data).

Identify the Source ICON Instance

"We use multiple instances of ICON. Which one created this specific record?"

To identify instance of ICON which was the source of a particular record in GIDB, use the `GSYS_SYS_ID` field in the GIDB record. This field contains the DBID of the ICON application as defined in the Configuration Layer.

Chapter

6 Representing Dates and Times of Day

This chapter describes how Genesys Info Mart represents dates and times of day. Because of the large volume of data handled by Genesys Info Mart, most SQL queries of a Fact table are constrained by date and time.

This chapter contains the following sections:

Dates and Times of Day

Dates and times of day are stored in the `START_TS` and `END_TS` fields, which mark the start and end of each handling stage. The `START_DATE_TIME_KEY` and `END_DATE_TIME_KEY` reference the `DATE_TIME` dimension, which exists in all Fact tables.

Dates and times are stored in Coordinated Universal Time (UTC) format. You can express local, enterprise, or tenant time using custom `DATE_TIME` dimensions that offset the UTC time by a specified amount of time.

Note: For instructions on configuring custom `DATE_TIME` tables, see the *Genesys Info Mart 8.0 Deployment Guide*.

Since in UTC, `0` = `1970 January 1`, each custom `DATE_TIME` dimension table will associate this UTC key with a different local time that is relevant for your enterprise. This enables you to use the same keys to create reports in different time zones.

How Dates and Times Can Be Constrained

Each Fact table row has a surrogate key, START_DATE_TIME_KEY, that references the DATE_TIME dimension that represents its start date and time. This surrogate key can constrain the Fact table rows by start date and time of day. Similarly, the END_DATE_TIME_KEY can be used to constrain the Fact table rows by end date and time of day.

Each Fact table row contains measurements that represent the start date and time of day, and the end date and time of day. These measurements can constrain Fact table rows by any arbitrary time span, based on whether the Fact table row:

- Starts and ends within the time span.
- Starts before, and ends within, the time span.
- Starts within, and ends after, the time span.
- Starts before, and ends after, the time span.

In any case, you must create the appropriate database indexes in order to efficiently retrieve the data you want.

All Fact tables have surrogate key references to the DATE_TIME dimension that represent the 15 minute date and time interval in which a fact started and ended.

The DATE_TIME dimension is useful for constraining based on an arbitrary range of 15-minute time intervals, because this single dimension includes both date and time of day. The dimension keys increase regularly each 15 minutes.

Working with Timestamps and the DATE_TIME Dimension

The following example illustrates how Genesys Info Mart represents the date and time of an inbound call in local time.

Example

An inbound call arrives at a contact center in San Francisco on October 21, 2009 at 5:05 pm local time (PDT). This time corresponds to 1:05 am on October 22, 2009 in UTC GMT time zone, or 1256173500 seconds, expressed in UTC integer format. This integer is stored in the START_TS field in the table containing data about the call.

The call's start time also falls into a 15-minute time interval that begins on October 22, 2009 at 1:00 am in the UTC GMT time zone, or 1256173200 seconds in UTC integer format. This integer is stored in the START_DATE_TIME_KEY field in the tables containing data related to the call.This value is a surrogate key that can be used to link to the corresponding

DATE_TIME_KEY field in any DATE_TIME_CUSTOM dimension. These custom tables contain text labels for the day of the week, month, year, and so on, in whichever local time zone formats your business requires.

In this example, a DATE_TIME_CUSTOM table has been created for the Pacific time zone containing labels in local PDT format. The START_DATE_TIME_KEY field in the Fact table containing the UTC integer 1256173200 (corresponding to 5:00 pm PDT), can be used to link to this DATE_TIME_CUSTOM dimension. The correct text labels for the Pacific time zone can then be retrieved for your reports.

Calculating Timestamps

To show timestamps in reports converted to a particular time zone, use a simple calculation combining the START_TS (or END_TS) field of a Fact table with the DATE_TIME_KEY and CAL_DATE fields of the DATE_TIME_CUSTOM table created for that time zone.

For example, to convert the timestamp value, 1256173500, from the Example on page 148, where the time of call arrival is stored in UTC seconds format in the START_TS field of the corresponding INTERACTION_RESOURCE_FACT (IRF) row in an MSSQL RDBMS, execute the following query on the custom DATE_TIME_CUSTOM dimension and IRF table:

```
select DTC.CAL_DATE + CAST ((IRF.START_TS - DTC.DATE_TIME_KEY) as
float) / CAST (86400 as float)
from DATE_TIME_CUSTOM DTC, INTERACTION_RESOURCE_FACT IRF
where DTC.DATE_TIME_KEY = IRF.START_DATE_TIME_KEY
```

The resulting value is October 21, 2009 at 5:05 pm in PDT time zone.

To make the same conversion in an Oracle RDBMS, execute the following query:

```
select DTC.CAL_DATE + (IRF.START_TS - DTC.DATE_TIME_KEY) / 86400
from DATE_TIME_CUSTOM DTC, INTERACTION_RESOURCE_FACT IRF
where DTC.DATE_TIME_KEY = IRF.START_DATE_TIME_KEY
```

Calendar Years and Week-Numbering Years

There are two available ways to number the weeks in a year:

- **Full-week numbering**—In this system, weeks always contain seven days and always start on the day of the week specified as Day 1 in the first-day-of-week configuration option. This system supports the ISO-8601 week configuration used in the European Union and Russia.

- **Simple-week numbering**—This results in the week calendar matches the calendar year. Week 1 begins on January 1. As a result, the first day of the week differs each year. Most of the time, Weeks 1 and 52 will have fewer than seven days. This is the functionality used in previous releases of Genesys Info Mart.

Week-Numbering Table Fields

The `DATE_TIME` table contains several fields used to support the full-week numbering system.

- `WEEK_YEAR`—This column stores a Week Numbering Year. This year may be different from Calendar year.

  For example, in ISO-8601, 31 December of 2007 is Week 1 Day 1 of 2008. So in this case we have 2007 as the Calendar year and 2008 as the Week Numbering year.
- `LABEL_YYYY_WE_D`—The label for the day of the week.
- `LABEL_TZ`—This field stores the time zone offset.

Week-Numbering Configuration Options

There are also a number of configuration options that control how week number is done.

- `first-day-of-week`—Specifies the day of the week on which the week should start. This is commonly Sunday in the United States and Monday for countries using ISO-8601.
- `day-of-first-week`—Specifies the day that defines first week of the new year. This day must be included in the first week.

  For example, ISO-8601 requires that the first week of the year always contain the first Thursday of the calendar year. So if `first-day-of-week` is set to `Monday`, then for ISO support `day-of-first-week` should be set to `4`.

  A convention used in the United States is that the first week of the year is the week that contains the first Saturday. In this case, if `First-day-of-week` is set to `Sunday`, `day-of-first-week` should be set to `7`.
- `simple-week-numbering`—To have the first week of the year start on January 1, set this option to `true`. This retains the functionality used in previous releases of Genesys Info Mart. The default setting for this option is `true`.

  This results in a partial week at the beginning of the year that begins on a day other than the week-start-day, and another partial week at the end of the year.

Note: You may need to reaggregate data after changing these options.

Maintenance of the DATE_TIME Dimension

The DATE_TIME dimension is a calendar that needs to be set up (populated) ahead of time. Maintenance of the DATE_TIME dimension is controlled by the day-time-min-days-ahead and day-time-max-days-ahead configuration options

Job_InitializeGIM and Job_MaintainGIM add records to the DATE_TIME dimension if the last existing record is earlier than current-date + day-time-min-days-ahead. Records are added until current-date + day-time-max-days-ahead.

For example, take a scenario in which day-time-min-days-ahead is set to 183 day-time-max-days-ahead is set to 366, and today is March 30, 2010.

Case 1: DATE_TIME is populated until January 1, 2011.

Since (January 1, 2011 - March 30, 2010) > 183, Job_InitializeGIM and Job_MaintainGIM will not add any records to DATE_TIME.

Case 2: DATE_TIME is populated until June 1, 2010.

Since (June 1, 2010 - March 30, 2010) < 183, Job_InitializeGIM and Job_MaintainGIM will add records to DATE_TIME until (March 30, 2010 + 366) = March 30, 2011.

Chapter

7 Unpopulated Genesys Info Mart Columns

This chapter describes the fields, attributes, and table columns that have limited values, or are not populated in this release of Genesys Info Mart. It contains the following sections:

External Resource ID

The RESOURCE_ dimension EXTERNAL_RESOURCE_ID is always set to Null.

Source Type

The SOURCE_TYPE column in the CTL_TRANSFORM HISTORY table is reserved for internal use.

Processing Status Key

The PROCESSING_STATUS_KEY field, which references the CTL_PROCESSING_STATUS dimension, is reserved for internal use.

Provider Tag

The PROVIDERTAG column contains the ID of the ICON provider class, such as 5 for the configuration information provider (cfg). The column is reserved for internal use.

Purge Flag

The PURGE_FLAG field, which occurs in all Fact tables, is always unpopulated.

Interaction Resource Ordinal

The INTERACTION_RESOURCE_ORDINAL column in the INTERACTION_RESOURCE_FACT table is reserved for internal use.

Last Interaction Resource

The LAST_INTERACTION_RESOURCE column in the INTERACTION_RESOURCE_FACT table is reserved for internal use.

Transform Discards

The STG_TRANSFORM_DISCARDS.GUID column is reserved for internal use.

Related Documentation Resources

The following resources provide additional information that is relevant to this software. Consult these additional resources, as necessary.

Framework

- The *Framework 8.0 Management Layer User's Guide* provides information about the concepts, terminology, and procedures that apply to this layer of the Genesys Framework.
- The *Framework 8.0 Configuration Options Reference Manual* provides information about configuration options for Framework components.
- The *Framework 8.0 Configuration Manager Help* provides information about using Configuration Manager in either an enterprise or a multi-tenant environment.
- The *Framework 8.0 Deployment Guide* provides information about configuring, installing, starting, and stopping Framework components.
- The *Framework 8.0 Combined Log Events Help* describes log events that Genesys server applications generate and that Solution Control Interface displays. The *Framework 8.0 Combined Log Events Help* includes descriptions of Genesys Info Mart log events.

Interaction Concentrator

- The *Interaction Concentrator 8.0 Deployment Guide* provides information about architecture, configuration requirements, and installation steps for Interaction Concentrator, and it describes how to make data from the Genesys Outbound Contact solution available in Interaction Database (IDB).

- The *Interaction Concentrator 8.0 User's Guide* provides basic information about IDB architecture and detailed information about Interaction Concentrator features and functionality, including attached data processing, available stored procedures, and integration with other Genesys products.
- The *Interaction Concentrator 8.0 Physical Data Model* for your relational database management system (RDBMS) provides information about the IDB schemas.

Genesys Info Mart

- The *Genesys Info Mart 8.0 Deployment Guide* provides information about architecture, configuration requirements, and installation steps for Genesys Info Mart and the Genesys Info Mart Administration Console.
- The *Genesys Info Mart 8.0 Operations Guide* provides information about the Genesys Info Mart jobs. The jobs extract, transform, and load (ETL) data, maintain the Info Mart database, and migrate the database schema as required. The guide also explains how to use the Genesys Info Mart Administration Console to monitor and administer the jobs and how to purge the Info Mart database.
- The *Genesys Info Mart 8.0 User's Guide* provides information about how to use data that is stored by Genesys Info Mart for contact center historical reporting.
- The *Genesys Info Mart 8.0 Reference Manual* for your RDBMS provides information about the Info Mart database schema.
- The *Genesys Info Mart 8.0 Database Size Estimator* helps you estimate the size of your Info Mart database when you are planning your deployment. The estimator is a Microsoft Office Excel 2007 spreadsheet that is available from the Genesys Technical Support website.
- Release Notes and Product Advisories for this product, which are available on the Genesys Technical Support website at http://genesyslab.com/support.

Reporting and Analytics Aggregates

- The *Reporting and Analytics Aggregates 8.0 Deployment Guide* describes how to deploy the Reporting and Analytics Aggregates (RAA) package provided with Genesys Info Mart.
- The *Reporting and Analytics Aggregates 8.0 Reference Manual* describes the aggregate tables that are available to Genesys Info Mart customers with deployment of RAA.
- The *Reporting and Analytics Aggregates 8.0 User's Guide* describes the aggregation process, provides the aggregation hierarchy, and explains how to enable aggregation of user data.

Genesys Interactive Insights

- The *Genesys Interactive Insights 8.0 Deployment Guide* describes how to install Genesys Interactive Insights (GI2) and set up the environment required in order to run the GI2 reports.
- The *Genesys Interactive Insights 8.0 Universe Guide* describes, in detail, the reports and measures that are provided in the GI2 release.
- The *Genesys Interactive Insights 8.0 User's Guide* summarizes how to operate GI2 reports and provides basic instructions for customizing your own reports.

Genesys

- The *Genesys Technical Publications Glossary,* which ships on the Genesys Documentation Library DVD, provides a comprehensive list of the Genesys and computer-telephony integration (CTI) terminology and acronyms that are used in this document.

Information about supported hardware and third-party software is available on the Genesys Technical Support website in the following documents:

- *Genesys Supported Operating Environment Reference Manual*
- *Genesys Supported Media Interfaces Reference Manual*

Consult the following additional resources as necessary:

- The *Genesys Migration Guide*, which ships on the Genesys Documentation Library DVD, provides documented migration strategies for Genesys product releases. Contact Genesys Technical Support for more information.
- The *Genesys Hardware Sizing Guide* provides information about Genesys hardware sizing guidelines for the Genesys 8.x releases.
- The *Genesys Interoperability Guide* provides information on the compatibility of Genesys products with various Configuration Layer Environments; Interoperability of Reporting Templates and Solutions; and *Gplus* Adapters Interoperability.
- The *Genesys Licensing Guide* introduces you to the concepts, terminology, and procedures that are relevant to the Genesys licensing system.
- The *Genesys Database Sizing Estimator 8.0 Worksheets* provides a range of expected database sizes for various Genesys products.

For additional system-wide planning tools and information, see the release-specific listings of System Level Documents on the Genesys Technical Support website. These documents are accessible from the `system level documents by release` tab in the Knowledge Base `Browse Documents` Section.

Genesys product documentation is available on the:

- Genesys Technical Support website at http://genesyslab.com/support.
- Genesys Documentation Library DVD, which you can order by e-mail from Genesys Order Management at orderman@genesyslab.com.

Document Conventions

This document uses certain stylistic and typographical conventions—introduced here—that serve as shorthands for particular kinds of information.

Document Version Number

A version number appears at the bottom of the inside front cover of this document. Version numbers change as new information is added to this document. Here is a sample version number:

80gim_dep_09-2010_v8.0.001.00

You will need this number when you are talking with Genesys Technical Support about this product.

Screen Captures Used in This Document

Screen captures from the product graphical user interface (GUI), as used in this document, may sometimes contain minor spelling, capitalization, or grammatical errors. The text accompanying and explaining the screen captures corrects such errors *except* when such a correction would prevent you from installing, configuring, or successfully using the product. For example, if the name of an option contains a usage error, the name would be presented exactly as it appears in the product GUI; the error would not be corrected in any accompanying text.

Type Styles

Table 8 describes and illustrates the type conventions that are used in this document.

Table 8: Type Styles

Type Style	Used For	Examples
Italic	• Document titles • Emphasis • Definitions of (or first references to) unfamiliar terms • Mathematical variables Also used to indicate placeholder text within code samples or commands, in the special case where angle brackets are a required part of the syntax (see the note about angle brackets on page 160).	Please consult the *Genesys Migration Guide* for more information. Do *not* use this value for this option. A *customary and usual* practice is one that is widely accepted and used within a particular industry or profession. The formula, $x + 1 = 7$ where x stands for . . .

Table 8: Type Styles (Continued)

Type Style	Used For	Examples
Monospace font (Looks like `teletype` or `typewriter text`)	All programming identifiers and GUI elements. This convention includes: • The *names* of directories, files, folders, configuration objects, paths, scripts, dialog boxes, options, fields, text and list boxes, operational modes, all buttons (including radio buttons), check boxes, commands, tabs, CTI events, and error messages. • The values of options. • Logical arguments and command syntax. • Code samples. Also used for any text that users must manually enter during a configuration or installation procedure, or on a command line.	Select the `Show variables on screen` check box. In the `Operand` text box, enter your formula. Click `OK` to exit the `Properties` dialog box. T-Server distributes the error messages in `EventError` events. If you select `true` for the `inbound-bsns-calls` option, all established inbound calls on a local agent are considered business calls. Enter `exit` on the command line.
Square brackets ([ ])	A particular parameter or value that is optional within a logical argument, a command, or some programming syntax. That is, the presence of the parameter or value is not required to resolve the argument, command, or block of code. The user decides whether to include this optional information.	`smcp_server -host [/flags]`
Angle brackets (< >)	A placeholder for a value that the user must specify. This might be a DN or a port number specific to your enterprise. **Note:** In some cases, angle brackets are required characters in code syntax (for example, in XML schemas). In these cases, italic text is used for placeholder values.	`smcp_server -host <confighost>`

Index

Symbols

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

T

U

V

W