Cisco Unified CCX Database Schema Guide, Release 11.5(1)

Cisco Unified Contact Center Express and Cisco Unified IP IVR
Release 11.5(1)

August 10, 2016

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Cisco Unified CCX Database Schema Guide, Release 11.5(1)

CONTENTS

Preface

Change History

Change	Date
Initial release of document for 11.5(1)	August 10, 2016

About This Guide

The *Cisco Unified CCX Database Schema Guide* for Cisco Unified Contact Center Express (Unified CCX) describes how data is organized in the Unified CCX databases. This document provides detailed description of the records and fields in each database table and enables you to create your own reports.

Audience

This manual is intended for system managers, administrators, and developers who want to create custom reports using the generally available third-party programs that create reports from databases.

Organization

The "Database Table Details" section on page 2 describes each table in the Unified CCX database. The descriptions are arranged in the alphabetical order by table name. Each description includes a detailed explanation of each record in the table. The Index helps you find information in this book.

Conventions

This manual uses the following conventions:

Convention	Description
boldface font	**Boldface** font is used to indicate commands, such as user entries, keys, buttons, and folder and submenu names. For example: • Choose **Edit** > **Find**. • Click **Finish**.
italic font	*Italic* font is used to indicate the following: • To introduce a new term. Example: A *skill group* is a collection of agents who share similar skills. • For emphasis. Example: *Do not* use the numerical naming convention. • A syntax value that the user must replace. Example: IF (*condition, true-value, false-value*) • A book title. Example: See the *Unified CCX Installation Guide*.
`window font`	`Window` font, such as Courier, is used for the following: • Text as it appears in code or that the window displays. Example: `<html><title>Cisco Systems,Inc. </title></html>`
< >	Angle brackets are used to indicate the following: • For arguments where the context does not allow italic, such as ASCII output. • A character string that the user enters but that does not appear on the window such as a password.

Related Documentation

The following documents contain additional information about the Unified CCX database and the information stored in the database:

- *Unified CCX Administration Guide*
- *Cisco Unified Contact Center Express Historical Reporting Guide*
- *Cisco Unified Contact Center Express Report Developer Guide*

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, see the monthly *What's New in Cisco Product Documentation*, which also lists all new and revised Cisco technical documentation, at:

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html

Subscribe to the *What’s New in Cisco Product Documentation* as a Really Simple Syndication (RSS) feed and set content to be delivered directly to your desktop using a reader application. The RSS feeds are a free service and Cisco currently supports RSS version 2.0.

Documentation Feedback

To provide your feedback for this document, send an email to:

mailto:contactcenterproducts_docfeedback@cisco.com

CHAPTER 1

Database Schema

This manual describes how data is organized in the Unified CCX database. The Unified CCX database stores the data in the following two types of databases:

- db_cra – Used to store information for historical and real-time reports, including Unified CCX configuration information, stored procedures, and call statistics.
- db_cra_repository - Used to store information related to prompts, grammars, scripts, and documents.

All the tables described in this document are in the above two databases.

Note SQL is case-insensitive and the queries written against the database can be in any case. However, you might have to change the case for the column names depending on the third-party tool that you use for querying the database. Refer to the documentation for these third-party tools for more information.

The following sections include these topics:

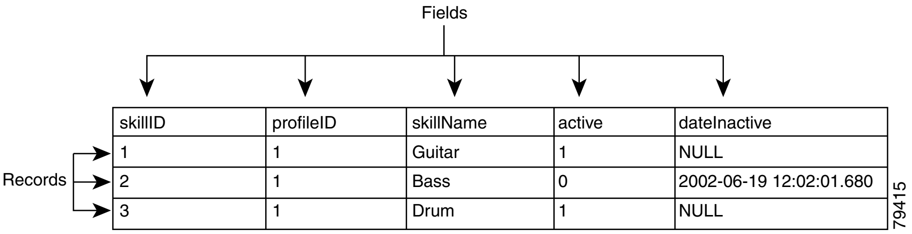
General Database Concepts

This section provides an overview of some basic database concepts.

Tables, Columns, and Rows

A database contains one or more tables of data. Each table in a database defines a set of columns, which are called *fields*. Within each table, the database stores data in rows, which are called *records*. Each record (row) contains one value for each field (column) of the table. For example, Figure 2 shows an example of a Skill table, which contains five fields. This example shows a Skill table with three records.

Figure 2 *Skill Table Fields and Records*


Fields
Records
79415

skillID	profileID	skillName	active	dateInactive
1	1	Guitar	1	NULL
2	1	Bass	0	2002-06-19 12:02:01.680
3	1	Drum	1	NULL

Database tables and the number and names of their fields are constant. The number of records in a table and the data that those records contain will vary according your system.

Table Relationships

Related tables in a database share one or more common fields. For example, both the Skill and the SkillGroup tables include the skillID field. Each record in the Skill table is related to each record in the SkillGroup table that shares the same skillID value.

Relationships between tables can be one-to-one or one-to-many. For example, because one skill can be associated with many skill groups, the relationship between the Skill and SkillGroup tables is one-to-many. On the other hand, each call or call leg has its own set of data about the agent who handled the call and other information. Therefore, the relationship between the AgentConnectionDetail and ContactCallDetail table is one-to-one.

Each database table description in this manual is followed by a Related Tables section. These sections show the fields by which a table is related to other tables. If the fields have different names in each table, these sections show the mapping.

Database Table Details

This section provides information about Unified CCX system database tables, their records, and their fields.

Each description provides the following information:

- Database Table Name—Name of the Unified CCX database table.
- Field Name—Name of a field as it appears in the database table.
- Description—Description of the field, including valid values where appropriate.
- Storage—Information about the data in each field as follows:
  - Data type used for the field in the database.

Note For storage characteristics and limitations of the data types used for the fields in the databases refer to "IBM Informix SQL Reference Guide". The date and time in the database fields are stored in Coordinated Universal Time (UTC).

  - Whether the NULL value is valid for the field. “NULL” if the NULL value is or “NOT NULL” if the NULL value is not valid.

Note If the NULL value is valid, the database will record a value of -1 for a numeric field and an empty string for other fields.

– “Primary Key” if the field is a primary key, or part of a primary key, in the database table.

Overview of Tables

The following tables are described in this guide:

- AgentConnectionDetail, page 5, contains records written for calls that are connected to an agent.
- AgentStateDetail, page 7, contains records written when an agent changes state.
- AreaCode, page 9, contains the area code and time zone information used for outbound calls.
- Campaign, page 10 contains records with campaign configuration information.
- CampaignCSQMap, page 12, provides a relationship between campaigns, and Contact Service Queues (CSQs).
- CampaignSupervisorMap, page 13, provides a relationship between campaign and the supervisor associated with it.
- CampaignData, page 14, contains records with the campaign data information.
- ChatProblemStatement, page 16, contains the associated problem statements and tag ids for each chat widget.
- ChatTriggerPoint, page 17, contains the chat CSQ tag information.
- ChatUserForm, page 18, contains the user form fields included in each chat widget along with the order of the fields in the widget.
- ChatWidget, page 18, contains the chat widget information.
- ContactCallDetail, page 19, contains records written for every incoming, outgoing, or internal call.
- ContactQueueDetail, page 25, contains records written for calls that are queued for CSQs; one record for each CSQ is queued.
- ContactRoutingDetail, page 27, contains records written for calls that are queued for CSQs; one record for each call.
- ContactServiceQueue, page 28, contains records written for CSQs configured on the Unified CCX Administration user interface.
- CrsApplication, page 31, contains records about applications that are uniquely identified by application name.
- CrsGroup, page 32, contains records about groups that are identified by a combination of group class name and group ID.
- CrsTrigger, page 34, contains records about triggers that are uniquely identified by trigger name.
- DialingList, page 36, contains records with outbound contacts that need to be dialed for a particular campaign.
- MonitoredResourceDetail, page 44, contains records written for agents who are monitored by a supervisor.
- ProfileIDMapping, page 46, contains records written for profiles defined on the Unified CCX Administration user interface.

- PurgeHistory, page 47, contains records written for tracking of the history of purge information for both Manual and Scheduled purge.
- RemoteMonitoringDetail, page 48, contains records written for remote monitoring calls made by a supervisor.
- Resource, page 51, contains records written for resources (agents) that are configured on the Cisco Unified Communications Manager (Unified CM) Administration user interface.
- ResourceGroup, page 53, contains records written for resource groups configured on the Unified CCX Administration user interface.
- ResourceSkillMapping, page 53, is a relationship table between resources and skills.
- RmonCSQConfig, page 54, contains records written for CSQs configured for a supervisor's remote monitoring allowed list on the Unified CCX Administration user interface.
- RmonResConfig, page 55, contains records written for resources configured for a supervisor's remote monitoring allowed list on the Unified CCX Administration user interface.
- RmonUser, page 56, contains records written for remote monitoring supervisors configured on the Unified CCX Administration user interface.
- RtCSQsSummary, page 57, contains real-time statistics for configured CSQs.
- RtICDStatistics, page 58, contains Unified CCX summary statistics.
- Skill, page 60, contains records written for skills configured on the Unified CCX Administration user interface.
- SkillGroup, page 61, is a relationship table between skills and CSQs.
- Supervisor, page 62, contains records written for supervisors configured on the Unified CCX Administration user interface.
- Team, page 63, contains records written for teams configured on the Unified CCX Administration user interface.
- TeamCSQMapping, page 64, is a relationship table between teams and CSQs.
- TextAgentConnectionDetail, page 65, contains information relating to the agent who handled the contact or leg.
- TextAgentStateDetail, page 66, contains information about the chat agent and about the event that caused the chat agent state change.
- TextContactDetail, page 68, contains detailed information about the contact or leg.
- TextContactQueueDetail, page 70, is a relationship table between teams and CSQs.
- TextCustomer Detail, page 71, contains customer related information corresponding to the chat contact.
- WorkflowTask, page 72, contains records written for workflow tasks that are executed.
- EEMActiveEmail, page 73, contains one row for each email message being processed by an agent.
- EEMContactEmailDetail, page 74, contains one row for each email message currently in the system.
- EEMEmailAgentStateDetail, page 76, contains one row for each Email state change for an agent.
- EEMEmailStatusDescription, page 77, contains descriptive text (in English only) for each possible email status.
- EEMQueueAgentDetail, page 77, (associates an agent and an email message, contains one record for each email message an agent is working on.

- EEMReasonCodeDescription, page 79, contains descriptive text (in English only) for each possible reason code.
- EEMStateDescription, page 79, contains descriptive text (in English only) for each email state.
- EEMTablesVersion, page 80, contains a single row indicating the version of the EEM database tables. This is used when upgrading from a previous version.

AgentConnectionDetail

Database table name: AgentConnectionDetail

The Unified CCX system creates a new record in the AgentConnectionDetail table when an agent disconnects a call or a leg by hanging up or by transferring the call. (A new call leg starts each time that a call is transferred, except when a call is transferred from a Cisco Computer Telephony Interface [CTI] port to an agent.)

An AgentConnectionDetail record contains information relating to the agent who handled the call or call leg.

The AgentConnectionDetail table contains the fields shown Table 1-1.

Table 1-1 AgentConnectionDetail Table Fields

Field Name	Description	Storage
sessionID	Identifier that the system assigned to the call. This identifier remains the same for all legs of the call.	decimal(18, 0) NOT NULL Primary Key
sessionSeqNum	Session sequence number that the system assigned to the call or the leg. Each leg of a call is assigned a new sequence number.	smallint NOT NULL Primary Key
nodeID	Unique identifier assigned to each Unified CCX server in the cluster.	smallint NOT NULL Primary Key
profileID	Identifier of the Unified CCX profile that is associated with this record.	int NOT NULL Primary Key
resourceID	Identifier of the agent who handled the call.	int NOT NULL Primary Key
startDateTime	Date and time that the call or the leg started ringing at the device of an agent.	datetime year to fraction (3) NOT NULL Primary Key

Table 1-1 AgentConnectionDetail Table Fields (continued)

Field Name	Description	Storage
endDateTime	Date and time that the call or the leg was transferred or disconnected.	datetime year to fraction (3) NOT NULL
qIndex	A new qIndex is created whenever a Unified CCX call is conferenced to a Unified CCX route point.	smallint NOT NULL Primary Key
gmtOffset	Offset, in minutes, between the local time of the Unified CCX server and Greenwich Mean Time. As the time information is stored in GMT, this field will always be zero.	smallint NOT NULL
ringTime	Amount of time, in seconds, between the time the call or the leg first rang at the extension of an agent and one of the following events: • The agent answered the call or the leg • The caller hung up before the call or the leg was answered • The system retrieved the call or the leg before the call or the leg was answered	smallint NULL
talkTime	Amount of time, in seconds, that passed from the time an agent answered the call or the leg to the time the call or the leg was disconnected or transferred, not including hold time.	smallint NULL
holdTime	Amount of time, in seconds, that the call or the leg spent on hold.	smallint NULL
workTime	Amount of time, in seconds, that an agent spent in Work State after the call or the leg.	smallint NULL
callWrapupData	After-call information that the agent enters through the Agent Desktop user interface while the agent is in the work state.	varchar(40) NULL

Table 1-1 AgentConnectionDetail Table Fields (continued)

Field Name	Description	Storage
callResult	Outcome of the outbound dialer call. 1 = Voice (Customer answered and was connected to agent) 2 = Fax/Modem (Fax machine detected) 3 = Answering Machine (answering machine detected) 4 = Invalid (Number reported as invalid by the network) 5 = Do Not Call (customer does not want to be called again) 6 = Wrong Number (number successfully contacted but wrong number) 7 = Customer Not Home (number successfully contacted but reached the wrong person) 8 = Callback (customer requested regular callback) 9 = Agent Rejected (Agent has skipped or rejected a preview call) 10 = Agent Closed (Agent has skipped or rejected a preview call with the close option) 11 = Busy (busy signal detected) 12 = RNA (the agent lets the call go ring-no-answer) 20 = OB_XFER is default (the agent transfers or conferences the outbound call from another agent)	smallint NULL
dialinglistid	Unique identifier of a contact that is dialed for an outbound campaign. Links with DialingList.dialingListID	int NULL

Related Tables

- ContactCallDetail, page 19 (via sessionID, sessionSeqNum, nodeID, and profileID)
- ContactRoutingDetail, page 27 (via sessionID, sessionSeqNum, nodeID, profileID, and qIndex)
- ContactQueueDetail, page 25 (via sessionID, sessionSeqNum, nodeID, profileID, and qIndex)
- DialingList, page 36 (via dialingListID)
- MonitoredResourceDetail, page 44 (sessionSeqNum maps to monitoredSessionSeqNum, via profileID, and nodeID)
- ProfileIDMapping, page 46 (via profileID)
- PurgeHistory, page 47 (sessionID maps to monitoredSessionID, via profileID, and nodeID)
- Resource, page 51 (via resourceID and profileID)

AgentStateDetail

Database table name: AgentStateDetail

The Unified CCX system creates a new record in the AgentStateDetail table each time the state of an agent changes. An AgentStateDetail record contains information about the agent and about the event that caused the agent state change.

The AgentStateDetail table contains the fields shown in Table 1-2.

Table 1-2 AgentStateDetail Table Fields

Field Name	Description	Storage
agentID	Identifier of the agent whose state has changed.	int NOT NULL Primary Key
eventDateTime	Date and time that the agent state changed.	datetime year to fraction (3) NOT NULL Primary Key
gmtOffset	Offset, in minutes, between the local time of the Unified CCX server and Greenwich Mean Time. As the time information is stored in GMT, this field will always be zero.	smallint NOT NULL
eventType	Event that triggered the agent state change: 1—Log In 2—Not Ready 3—Ready 4—Reserved 5—Talking 6—Work 7—Log Out	smallint NOT NULL Primary Key
reasonCode	Code, as set up in the Cisco Desktop Administrator, for the reason that the agent changed to Not Ready State or to Log Out State. Null if a reason code is not configured.	smallint NOT NULL Primary Key
profileID	Identifier of the Unified CCX profile that is associated with this record.	int NOT NULL Primary Key

Related Tables

- ProfileIDMapping, page 46 (via profileID)
- Resource, page 51 (agentID maps to resourceID and via profileID)

AreaCode

Database table name: AreaCode

The AreaCode table contains a mapping of area codes and their time zones. This table is used as a reference for populating the gmtPhone and dstPhone columns of the DialingList table. This table is pre-populated by the Unified CCX system with the data for North America during the installation process, using a SQL script that the installer invokes. If the Unified CCX is installed in a different location, administrators can enter the area code and time zone information for that region using Unified CCX Administration, and the data is stored in this table.

The AreaCode table contains the information shown in Table 1-3.

Table 1-3 AreaCode Table Fields

Field Name	Description	Storage
profileid	Identifier of the profile.	int NOT NULL Primary Key
createdatetime	Default -CURRENT_TIMESTAMP	datetime year to second NOT NULL
recordid	Unique identifier for the record	int NOT NULL Primary Key
areacode	The area code of the call.	nvarchar(10) NOT NULL Primary Key
regioncode	Uses the same data as that of gmtzone.	nvarchar(10) NULL
daylightsavingsenabled	Indicates whether daylight savings time is observed. • N = Daylight savings time is not observed. • Y = Daylight savings time is observed.	char(1) NOT NULL
gmtzone	Stores identifiers that internally maps to the GMT offset corresponding to the area code.	int NULL
privatedata	Any fields which are to be used internally only.	BLOB NULL

active	Whether the record is active in the system. A record becomes inactive if the team is deleted from the system.
	f = Inactive
	t = Active
		boolean
		NOT NULL

dateinactive	Date this record was deleted.	datetime year to second NULL

Related tables

- DialingList, page 36 (via gmtZonePhone01, gmtZonePhone02, and gmtZonePhone03)
- ProfileIDMapping, page 46 (via ProfileID)

Campaign

Database table name: Campaign

The campaign configuration information is stored in this table. A campaign is associated with one or more CSQs. This mapping of Campaigns and CSQs is stored separately in CampaignCSQMap table.

The Campaign table contains the fields shown in Table 1-4

Table 1-4 Campaign Table Fields

Field Name	Description	Storage
recordid	A unique identifier for the record.	int NOT NULL Primary Key
campaignid	A unique identifier for the campaign.	int NOT NULL Primary Key
profileid	Identifier of the Unified CCX profile that is associated with this record.	int NOT NULL Primary Key
createdatetime	Default -CURRENT_TIMESTAMP	datetime year to second NOT NULL
campaignname	Name of the campaign. Must be unique.	nvarchar(50) NOT NULL
enabled	0 = campaign is enabled 1 = campaign is disabled	smallint NOT NULL
description	A description of the campaign.	varchar(50) NULL
starttime	When the campaign starts (based on server time). This is stored in minutes.	int NOT NULL
endtime	When the campaign ends (based on server time). This is stored in minutes.	int NOT NULL
cachesize	Number of contacts to be retrieved in a batch for dialing for this campaign.	int NOT NULL
maxattempts	Maximum number of attempts made to dial a contact for this campaign.	int NOT NULL

Table 1-4 Campaign Table Fields (continued)

Field Name	Description	Storage
ansmachineretry	0 = Dialer should try dialing a contact again if it reached an answering machine 1 = Dialer should not try dialing a contact again if it reached an answering machine	smallint NOT NULL
callbacktimelimit	The amount of time in minutes before and after the scheduled callback time, during which the Dialer attempts a callback.	int NULL
missedcallback action	Indicates what the Dialer should do if a callback could not be placed at the scheduled time: 0 = reschedule callback to same time the next business day 1 = make an ordinary retry 2 = close record	int NULL
privatedata	Any fields which are used internally only can be stored in this column in a blob.	BLOB NULL

active	Indicates whether the record is active in the system. A record becomes inactive if the campaign is deleted from the system.
	f = Inactive
	t = Active
		boolean
		NOT NULL

dateinactive	Date this record was deleted.	datetime year to second NULL
dialerType	The type of the dialer used for the campaign. The dialer can be any one of the following three types - Predictive, Progressive or Preview Outbound. 0 - Direct Preview Dialer 1 - IVR based Predictive Dialer 2 - IVR based Progressive Dialer Default value = 0	smallint NOT NULL
campaignType	The campaign type can be IVR-based or ICD-based. 0 - IVR based campaign 1 - Agent based campaign Default value = 1	smallint NOT NULL
campaignCallingNum	The campaign calling number that is displayed to the contact. This number is used by the outbound IVR dialer. **Note** This field will have value only if you have an Outbound IVR license on top of Unified CCX premium license in your Unified CCX and the dialer type is progressive or predictive.	BLOB NULL

Table 1-4 Campaign Table Fields (continued)

Field Name	Description	Storage
applicationTrigger	This is the JTAPI trigger associated with this campaign. **Note** This field will have value only if you have an Outbound IVR license on top of Unified CCX premium license in your Unified CCX and the dialer type is progressive or predictive.	BLOB NULL
applicationName	The name of the application associated with the above-mentioned JTAPI trigger. **Note** This field will have value only if you have an Outbound IVR license on top of Unified CCX premium license in your Unified CCX and the dialer type is progressive or predictive.	BLOB NULL

Related Tables

- CampaignCSQMap, page 12 (via campaignID)
- ContactCallDetail, page 19 (via campaignID and profileID)
- DialingList, page 36 (via campaignID)
- ProfileIDMapping, page 46 (via profileID)

CampaignCSQMap

Database table name: CampaignCSQMap

The CampaignCSQMap table shows the relationship between campaigns and contact service queues (CSQs). A new record is created in the CampaignCSQMap table when a campaign is associated with a CSQ in Unified CCX Administration.

The CampaignCSQMap table contains the fields shown in Table 1-5

Table 1-5 CampaignCSQMap Table Fields

Field Name	Description	Storage
recordid	A unique identifier for the record	int NOT NULL Primary Key
campaignid	A unique identifier for the campaign, from the Campaign table.	int NOT NULL Primary Key
csqid	A unique identifier for the CSQ, from the ContactServiceQueue table.	int NOT NULL Primary Key

Table 1-5 CampaignCSQMap Table Fields (continued)

Field Name	Description	Storage

active	Indicates whether the record is active in the system. A record becomes inactive if the campaign is deleted from the system.
	f = Inactive
	t = Active
		boolean
		NULL

createdatetime	Default, CURRENT_TIME_STAMP	datetime year to second NOT NULL
dateinactive	Date this record was deleted.	datetime year to second

Related Tables

- Campaign, page 10 (via campaignID)
- ContactServiceQueue, page 28 (csqID maps to contactServiceQueueID)

CampaignSupervisorMap

Database table name: CampaignSupervisorMap

The CampaignSupervisorMap table shows the relationship between campaign and supervisor associated with that campaign. A new record is created in the CampaignSupervisorMap table when a campaign is associated with a supervisor in Unified CCX Administration.

The CampaignSupervisorMap table contains the fields shown in Table 1-6

Table 1-6 CampaignSupervisorMap Table Fields

Field Name	Description	Storage
recordid	A unique identifier for the record	int NOT NULL Primary Key
campaignid	A unique identifier for the campaign, from the Campaign table.	int NOT NULL Primary Key
supervisorid	A unique identifier for the supervisor, based on the supervisor’s resourceloginid.	nvarchar(50) NOT NULL Primary Key

active	Indicates whether the record is active in the system. A record becomes inactive if the campaign is deleted from the system.
	f = Inactive
	t = Active
		boolean
		NULL

Table 1-6 CampaignSupervisorMap Table Fields (continued)

Field Name	Description	Storage
createdatetime	Default, CURRENT_TIME_STAMP	datetime year to second NOT NULL
dateinactive	Date this record was deleted.	datetime year to second

Related Tables

- Campaign, page 10 (via campaignID)
- Supervisor, page 62 (via supervisorID maps to resourceloginID)

CampaignData

Database table name: CampaignData

If you have configured a campaign as an outbound IVR campaign and have chosen Predictive dialer type, the number of lines per port along with the other parameters are stored in the CampaignData table every half hour.

The CampaignData table contains the fields shown in Table 1-7.

Table 1-7 CampaignData Table Fields

Field Name	Description	Storage
campaignId	The campaign for which the data is recorded	int NOT NULL Primary Key
recordId	A unique identifier for the record	int NOT NULL Primary Key
startDate	Start date and time of the interval	datetime year to fraction NOT NULL
endDate	End date and time of the interval	datetime year to fraction NOT NULL
attemptedCalls	The number of attempted calls in the interval	int NOT NULL
abandonedCalls	The number of abandoned calls in the interval	int NOT NULL

Field Name	Description	Storage
voiceCalls	The number of voice calls in the interval	int NOT NULL
linesPerPort	Lines Per Port value computed depending on the abandoned calls/voice calls	decimal(8, 3) NOT NULL

active	Indicates whether the data stored is for an active campaign or not.
	f = Inactive
	t = Active
		boolean
		NOT NULL

dateInactive	The date on which this campaign was deleted	datetime year to fraction

Related Tables

- Campaign, page 10 (via campaignID)

ChannelProvider

Database table name: ChannelProvider

This table contains the channel provider configurations, including the type of channel, and server details with fully qualified domain name (FQDN), protocol, and port. Channel providers enable the use of non-interactive media channels such as email with Unified CCX.

The ChannelProvider table contains the fields shown in Table 1-8

Table 1-8 ChannelProvider Table Fields

Field Name	Description	Storage
Id	Unique identifier of the channel provider. Server ID is the foreign key that associates this table with the ContactServiceQueue table.	int NOT NULL Primary Key
channelType	Type of contact channel.	varchar (20, 0) NOT NULL
sendserverfqdn	FQDN of the channel provider for sending the channel type.	Varchar(255, 0) NOT NULL
sendprotocol	Sending protocol that is used to communicate with the channel provider.	Varchar(20, 0) NOT NULL
sendserverport	16-bit port number that is used to communicate with the channel provider for sending the channel type.	int NOT NULL
receiveserverfqdn	FQDN of the channel provider for receiving the channel type.	Varchar(255, 0) NOT NULL

Table 1-8 ChannelProvider Table Fields (continued)

Field Name	Description	Storage
receiveprotocol	Receiving protocol that is used to communicate with the channel provider.	Varchar(20, 0) NOT NULL
receiveserverport	16-bit port number that is used to communicate with the channel provider for receiving the channel type.	int NOT NULL
description	Description of the channel provider.	Lvarchar(400)

active	Indicates whether the record is currently active or not. A record becomes inactive if the record is deleted or updated from the system.
	f = Inactive
	t = Active
		Boolean
		NOT NULL

dateinactive	If the active field is "f", this field indicates the date and time that the record became inactive.	datetime year to second
proxytype	Indicates whether Enable/Disable option is selected for SOCKS Proxy in Mail server configuration page.	Lvarchar(25)

Related Tables

- ContactServiceQueue, page 28 (via serverID)

ChatProblemStatement

Database table name: ChatProblemStatement

This table contains the associated problem statements and tag ids for each chat widget. Chat widget is the widget that enables the Unified CCX Administrator to create a chat interface for the end user.

The ChatProblemStatement table contains the fields shown in the below table.

Table 1-9 ChatProblemStatement Table Fields

Field Name	Description	Storage
wdID	Unique ID for each chat widget. It is the foreign key which associates this table with the chatwidget table.	int NOT NULL Primary Key
tagID	The tagID for the csq associated with the problem statement.	nvarchar(50) NOT NULL

Field Name	Description	Storage
wdID	Unique ID for each chat widget. It is the foreign key which associates this table with the chatwidget table.	int NOT NULL Primary Key
problemStmt	The definition of the problem.	lvarchar (256) NOT NULL
psOrder	Order of the problem statement in the chat widget.	int NOT NULL Primary Key

Related Tables

- ChatWidget, page 18 (via wdID)

ChatTriggerPoint

Database table name: ChatTriggerPoint

This table describes chat CSQ tag information. A ChatTriggerPoint is uniquely identified by a csqID and a chattriggerpointname. Chat contacts inserted into Unified CCX are queued to respective CSQs based on the chattriggerpointname present in the contact. When a chat CSQ is created, a new record is inserted into this table. When a CSQ is modified, the old record is marked as inactive, and a new record is inserted into the table with a the new csqID. When a CSQ is deleted, the corresponding record is marked as inactive.

The ChatTriggerPoint table contains the fields shown in Table 1-10.

Table 1-10 ChatTriggerPoint Table Fields

Field Name	Description	Storage
csqID	Numeric identifier for the CSQ.	int NOT NULL
chattriggerpointname	Name of fields present in the chat trigger point.	Ivarchar(256) NOT NULL

active	Indicates whether the record is currently active.
	f = Inactive
	t = Active
		boolean
		NOT NULL

dateinactive	If the active field is “f”, date and time that the record became inactive.	Datetime year to fraction(3) NULL

Related Tables

- ChatUserForm, page 18

ChatUserForm

Database table name: ChatUserForm

This table contains the user form fields included in each chat widget along with the order of the fields in the widget. The ChatUserForm table contains the fields shown in Table 1-11.

Table 1-11 ChatUserForm Table Fields

Field Name	Description	Storage
wdID	Unique ID for each chat widget. It is the foreign key which associates this table with the chatwidget table.	int NOT NULL Primary Key
fieldName	Name of fields present in the user form.	nvarchar(50) NOT NULL
fieldID	ID of fields present in the user form.	int NOT NULL Primary Key
fieldOrder	Order of the field in the widget.	smallint NOT NULL

active	Indicates whether the record is currently active or not.
	f = Inactive
	t = Active
		Boolean
		NOT NULL

lastmodifieddate	The date and time when the user form details were last modified.	datetime year to fraction

Related Tables

- ChatWidget, page 18 (via wdID)

ChatWidget

Database table name: ChatWidget

This table stores the chat widget information. The ChatWidget table contains the fields shown in below table.

Table 1-12 ChatWidget Table Fields

Field Name	Description	Storage
wdID	A unique ID for each widget.	int NOT NULL Primary Key
wdName	Name of the widget.	nvarchar(50) NOT NULL
wdDescription	The description of the widget that is configured in the Unified CCX Administration.	lvarchar (256) NOT NULL
wdCode	Blob data to store the HTML code generated for the widget.	BLOB NULL

active	Indicates whether the widget is currently active or not.
	f = Inactive
	t = Active
		boolean
		NOT NULL

lastModifiedDate	The date and time on which the widget details were last modified.	datetime year to fraction (3) NOT NULL

Related Tables

- ChatProblemStatement, page 16
- ChatUserForm, page 18

ContactCallDetail

Database table name: ContactCallDetail

The Unified CCX system creates a new record in the ContactCallDetail table for each call or call leg processed by the system. A new call leg starts each time that a call is transferred or redirected, except when a call is transferred from a Cisco CTI port to an agent.

A ContactCallDetail record contains detailed information about the call or leg. At least one such record will exist for each call.

The ContactCallDetail table contains the fields shown in Table 1-13.

Table 1-13 ContactCallDetail Table Fields

Field Name	Description	Storage
sessionID	Identifier that the system assigned to the call. This identifier remains the same for all legs of the call.	decimal(18, 0) NOT NULL Primary Key
sessionSeqNum	Session sequence number that the system assigned to the call or the leg. Each leg of a call is assigned a new sequence number.	smallint NOT NULL Primary Key
nodeID	Unique identifier assigned to each server in the cluster.	smallint NOT NULL Primary Key
profileID	Identifier of the Unified CCX profile that is associated with this record.	int NOT NULL Primary Key
contactType	Contact type of the call or the leg: 1—Incoming. Outside call received by the Unified CCX system. 2—Outgoing. Call originated by the Unified CCX system, other than a call made within the system. 3—Internal. Call transferred or conferenced between agents, or a call made within the system. 4—Redirect in 5—Transfer in 6—Preview Outbound 7—IVR Outbound 8—Agent Outbound 9—Agent Outbound call transferred to IVR	smallint NOT NULL
contactDisposition	Disposition of the call or the leg. 1—Abandoned 2—Handled 3—Do not care 4—Aborted[1] 5-22—Rejected 99—Cleared	smallint NOT NULL
dispositionReason	Reason why the call is aborted or rejected by the system.	varchar(100) NULL

Table 1-13 ContactCallDetail Table Fields (continued)

Field Name	Description	Storage
originatorType	Originator of the call or the leg: 1—Agent. Call originated by an agent. 2—Device. Call originated by a simulated caller (used for testing) and an agent phone where the agent is not currently logged in. 3—Unknown. Call originated by an outside caller through a gateway or by an unknown device.	smallint NOT NULL
originatorID	Numeric identifier of the agent who originated the call or the leg. Used only if originatorType is 1.	int NULL
originatorDN	If originatorType is 1 and the call was placed by the agent using the non-IPCC extension then this field contains the non-IPCC extension, else it contains an empty character (''). If originatorType is 2, this field shows the CTI port number. If originatorType is 3, this field shows the telephone number of the caller as received by the Unified CM, if available. An empty character ('') if originatorType is 1. This is not applicable for agent based progressive and predictive outbound calls.	nvarchar(30) NULL
destinationType	Destination of the call or the leg: 1—Agent. Call presented to an agent. 2—Device. Call presented to a route point. 3—Unknown. Call presented to an outside destination through a gateway or to an unknown device. Null if no destination.	smallint NULL
destinationID	Numeric identifier of the agent who received the call or the leg. Used only if destinationType is 1.	int NULL
destinationDN	If the destinationType is 1 and the call was received by an agent using the non-IPCC extension, then this field contains the non-IPCC extension, else it contains an empty character (''). If destinationType is 2, this field shows the CTI port number. If destinationType is 3, this field shows the telephone number called, if available. An empty character ('') if destinationType is 1.	nvarchar(30) NULL

Table 1-13 ContactCallDetail Table Fields (continued)

Field Name	Description	Storage
startDateTime	For an incoming call or a leg, date and time that the call or the leg started to ring in the system. For an internal call or for an outgoing call, date and time that the call originated. For a transferred call or a leg, endDateTime of the transferring call or leg.	datetime year to fraction (3) NOT NULL
endDateTime	Date and time that this call or the leg was transferred or was disconnected.	datetime year to fraction (3) NOT NULL
gmtOffset	Offset, in minutes, between the local time of the Unified CCX server and Greenwich Mean Time. As the time information is stored in GMT, this field will always be zero.	smallint NOT NULL
calledNumber	Telephone number of the device to which the call or leg was presented. If the call or leg was placed to a Unified CCX Route Point, this field shows the directory number configured in the Unified CM for that Route Point. If the call was placed to an external party, this field shows the telephone number dialed by the caller.	nvarchar(30) NULL
origCalledNumber	Telephone number dialed by the caller if the call was placed from an IP phone. The Unified CM directory number to which the VoIP gateway routed the call if the call was placed from outside the VoIP[2] network (for example, from the PSTN[3] or a TDM[4] PBX[5]). Null if the caller picked up the phone but did not dial any digits.	nvarchar(30) NULL
applicationTaskID	Identifier of the Unified CCX or Cisco Unified IP IVR[6] (Unified IP IVR) application task that is associated with the call or the leg. Null for a call that does not have an application associated with it.	decimal(18,0) NULL
applicationID	Identifier of the Unified CCX or Unified IP IVR application that processed the call or the leg. Null for a call or a leg that does not have an application associated with it.	int NULL
applicationName	Name of the Unified CCX or Unified IP IVR application associated with the call. Null for a call or a leg that does not have an application associated with it.	nvarchar(30) NULL
connectTime	Amount of time, in seconds, between the start time of the call or the leg and the end time of the call or the leg.	smallint NULL

Table 1-13 ContactCallDetail Table Fields (continued)

Field Name	Description	Storage
customVariable1	Contents of the variable _ccdrVar1, if this variable is set by the Set Enterprise Call step in the script that the Unified CCX or Unified IP IVR application associated with this call or this leg invoked. Null if this variable is not set.	varchar(40) NULL
customVariable2	Contents of the variable _ccdrVar2, if this variable is set by the Set Enterprise Call step in the script that the Unified CCX or Unified IP IVR application associated with this call or this leg invoked. Null if this variable is not set.	varchar(40) NULL
customVariable3	Contents of the variable _ccdrVar3, if this variable is set by the Set Enterprise Call step in the script that the Unified CCX or Unified IP IVR application associated with this call or this leg invoked. Null if this variable is not set.	varchar(40) NULL
customVariable4	Contents of the variable _ccdrVar4, if this variable is set by the Set Enterprise Call step in the script that the Unified CCX or Unified IP IVR application associated with this call or this leg invoked. Null if this variable is not set.	varchar(40) NULL
customVariable5	Contents of the variable _ccdrVar5, if this variable is set by the Set Enterprise Call step in the script that the Unified CCX or Unified IP IVR application associated with this call or this leg invoked. Null if this variable is not set.	varchar(40) NULL
customVariable6	Contents of the variable _ccdrVar6, if this variable is set by the Set Enterprise Call step in the script that the Unified CCX or Unified IP IVR application associated with this call or this leg invoked. Null if this variable is not set.	varchar(40) NULL
customVariable7	Contents of the variable _ccdrVar7, if this variable is set by the Set Enterprise Call step in the script that the Unified CCX or Unified IP IVR application associated with this call or this leg invoked. Null if this variable is not set.	varchar(40) NULL
customVariable8	Contents of the variable _ccdrVar8, if this variable is set by the Set Enterprise Call step in the script that the Unified CCX or Unified IP IVR application associated with this call or this leg invoked. Null if this variable is not set.	varchar(40) NULL

Table 1-13 ContactCallDetail Table Fields (continued)

Field Name	Description	Storage
customVariable9	Contents of the variable _ccdrVar9, if this variable is set by the Set Enterprise Call step in the script that the Unified CCX or Unified IP IVR application associated with this call or this leg invoked. Null if this variable is not set.	varchar(40) NULL
customVariable10	Contents of the variable _ccdrVar10, if this variable is set by the Set Enterprise Call step in the script that the Unified CCX or Unified IP IVR application associated with this call or this leg invoked. Null if this variable is not set.	varchar(40) NULL
accountNumber	Account number entered by the caller.	varchar(40) NULL
callerEnteredDigits	Phone number entered by the caller.	varchar(40) NULL
badCallTag	Tag for a bad call. Default = N	char(1) NULL

transfer	Was this call leg transferring the call:
	t = transfer
	f = no
		boolean
		NULL

redirect	Was this call leg redirecting the call:
	t = redirect
	f = no
		boolean
		NULL

conference	Was this a call leg conferencing the call:
	t = conference
	f = no
		boolean
		NULL

flowout	When this flag is set, it means this call leg is sent to another application or destination outside the system.	boolean NULL

metServiceLevel	Did the call meet the service level:
	t = met service level
	f = no
	Note: Reserved for future use.
		boolean
		NULL

campaignID	Unique identifier of the campaign that generated this call.	int NULL
OrigProtocolCallRef	Unique identifier to identify a call leg that enters the Unified CCX system. This is used to trace a call which has traversed from some product to the Unified CCX.	Varchar(32) NULL

Table 1-13 ContactCallDetail Table Fields (continued)

Field Name	Description	Storage
DestProtocolCallRef	Unique Identifier to identify a call leg that exits the Unified CCX system. This is used to trace a call which has traversed from Unified CCX to some other product.	Varchar(32) NULL
CallResult	The result of an IVR based or agent based progressive or predictive outbound call.	smallint NULL

1. For aborted calls, the corresponding value in ContactQueueDetail.disposition = 1 (abandoned).
2. VoIP = Voice over Internet Protocol
3. PSTN = Public Switched Telephone Network
4. TDM = Time-Division Multiplexing
5. PBX = Private Branch Exchange
6. IVR = Interactive Voice Response

Related Tables

- AgentConnectionDetail, page 5 (via sessionID, sessionSeqNum, nodeID, and profileID)
- Campaign, page 10 (via campaignID)
- ContactQueueDetail, page 25 (via session ID, sessionSeqNum, nodeID, profileID)
- ContactRoutingDetail, page 27 (via sessionID, sessionSeqNum, nodeID, and profileID)
- MonitoredResourceDetail, page 44 (sessionSeqNum maps to monitoredSessionSeqNum, via sessionID, profileID, and nodeID)
- ProfileIDMapping, page 46 (via profileID)
- PurgeHistory, page 47 (sessionID maps to monitoredSessionID, via sessionID, sessionSeqNum, profileID, and nodeID)
- Resource, page 51 (via originatorID/destinationID maps to resourceID when originatorType/destinationType is 1, via profileID)

ContactQueueDetail

Database table name: ContactQueueDetail

The Unified CCX system writes the record when the call is queued for CSQs; then one of the following happens:

- Call is abandoned while queued for CSQs
- Call is being dequeued
- Caller is connected to an agent

The Contact Queue Detail table contains the fields shown in Table 1-14.

Table 1-14 ContactQueueDetail Table Fields

Field Name	Description	Storage
sessionID	Identifier that the system assigned to the call. This identifier remains the same for all legs of the call.	decimal(18,0) NOT NULL Primary Key
sessionSeqNum	Session sequence number that the system assigned to the call or the leg. Each leg of a call is assigned a new sequence number.	smallint NOT NULL Primary Key
profileID	Identifier of the Unified CCX profile that is associated with this record.	int NOT NULL Primary Key
nodeID	Unique identifier assigned to each server in the cluster.	smallint NOT NULL Primary Key
targetID	Numeric ID of the CSQ or the agent depending upon the targetType. 0—Numeric record ID of the CSQ. (See record ID description in the Contact Service Queue Table) 1—Numeric agent ID (see resourceID description in the Resource Table)	int NOT NULL Primary Key
targetType	Indicates whether the call was queued for a CSQ or for an agent. 0 = CSQ 1 = Agent	smallint NOT NULL Primary Key
qIndex	A new qIndex is created whenever the Unified CCX call is conferenced to a Unified CCX route point.	smallint NOT NULL Primary Key
queueOrder	The order of the call in the queue.	smallint NOT NULL
disposition	Disposition for this leg of the call for this CSQ. • Abandoned = 1[1] • Handled by CSQ = 2 • Dequeued from CSQ = 3 • Handled by script = 4 • Handled by another CSQ = 5	smallint NULL
metServiceLevel	Call answered within the configured number of seconds of queue time for this CSQ. • Yes = t • No = f	boolean NULL

Table 1-14 ContactQueueDetail Table Fields (continued)

Field Name	Description	Storage
queueTime	Number of seconds the caller spent in queue for this CSQ and this leg of the call.	smallint NULL
dialinglistid	Unique identifier of a contact that is dialed for an outbound campaign. Links with DialingList.dialingListID.	int NULL

1. For aborted calls, ContactQueueDetail.disposition = 1 and the corresponding ContactCallDetail.contactDisposition = 4 (aborted).

Related Tables

- AgentConnectionDetail, page 5 (via sessionID, sessionSeqNum, nodeID, profileID, and qIndex)
- ContactCallDetail, page 19 (via sessionID, sessionSeqNum, nodeID, and profileID)
- ContactRoutingDetail, page 27 (via sessionID, sessionSeqNum, nodeID, profileID, and qIndex)
- ContactServiceQueue, page 28 (targetID maps to recordID when targetType is 0, via profileID)
- MonitoredResourceDetail, page 44 (sessionSeqNum maps to monitoredSessionSeqNum, via profileID, and nodeID)
- ProfileIDMapping, page 46 (via profileID)
- PurgeHistory, page 47 (sessionID maps to monitoredSessionID, via profileID and nodeID)
- Resource, page 51 (targetID maps to resourceID when targetType is 1, and via profileID)

ContactRoutingDetail

Database table name: ContactRoutingDetail

The Unified CCX system creates a new record in the ContactRoutingDetail table for each Unified CCX call or call leg that is queued for one or more CSQs. A new call leg starts each time that a call is transferred or redirected, except when a call is transferred from a Cisco CTI port to an agent. The system also creates a new record in the ContactRoutingDetail table if a call is conferenced to a Unified CCX script.

A ContactRoutingDetail record contains information about call priority and accumulated queue time. This differs from the ContactQueueDetail record which shows individual queue time for each CSQ.

The Contact Routing Detail table contains the fields shown in Table 1-15.

Table 1-15 ContactRoutingDetail Table Fields

Field Name	Description	Storage
sessionID	Identifier that the system assigned to the call. This identifier remains the same for all legs of the call.	decimal(18, 0) NOT NULL Primary Key
sessionSeqNum	Session sequence number that the system assigned to the call or the leg. Each leg of a call is assigned a new sequence number.	smallint NOT NULL Primary Key

Table 1-15 ContactRoutingDetail Table Fields (continued)

Field Name	Description	Storage
nodeID	Unique identifier assigned to each server in the cluster.	smallint NOT NULL Primary Key
profileID	Identifier of the Unified CCX profile that is associated with this record.	int NOT NULL Primary Key
qIndex	A new qIndex is created whenever the Unified CCX call is conferenced to a Unified CCX route point.	smallint NOT NULL Primary Key
origPriority	Priority level assigned to the call or the leg when it was first queued. Null if a priority was not assigned.	smallint NULL
finalPriority	Priority level of the call or the leg when it ended. Null if a priority was not assigned.	smallint NULL
queueTime	Time, in seconds, that the call or the leg was queued before an agent picked up the call or the leg. This is the accumulated queue time perceived by the caller if the call is queued for more than one CSQ; in contrast, the ContactQueueDetail record records queue time for each individual CSQ.	smallint NULL
startDateTime	For an incoming call or a leg, date and time that the call or the leg was queued for the first CSQ.	datetime year to fraction (3) NOT NULL

Related Tables

- AgentConnectionDetail, page 5 (via sessionID, sessionSeqNum, nodeID, profileID, and qIndex)
- ContactCallDetail, page 19 (via sessionID, sessionSeqNum, nodeID, and profileID)
- ContactQueueDetail, page 25 (via sessionID, sessionSeqNum, nodeID, profileID, and qIndex)
- MonitoredResourceDetail, page 44 (sessionSeqNum maps to monitoredSessionSeqNum, via profileID, and nodeID)
- ProfileIDMapping, page 46 (via profileID)
- PurgeHistory, page 47 (sessionID maps to monitoredSessionID, via profileID, and nodeID)

ContactServiceQueue

Database table name: ContactServiceQueue

The Unified CCX system creates a new record in the ContactServiceQueue table when a CSQ is set up in Unified CCX Administration.

A ContactServiceQueue record contains information about the CSQ. One such record exists for each active and inactive CSQ. When a CSQ is deleted (deactivated), its record still remains in the database marked as inactive; that is, the active field value is “f”.

The ContactServiceQueue table contains the fields shown in Table 1-16.

Table 1-16 ContactServiceQueue Table Fields

Field Name	Description	Storage
contactServiceQueueID	Numeric identifier of the CSQ. This ID does not change when CSQ attributes are changed through the Unified CCX Administration user interface.	int NOT NULL
profileID	Identifier of the Unified CCX profile that is associated with this record.	int NOT NULL
CSQName	Name of the CSQ as set up in Unified CCX Administration.	nvarchar(50) NOT NULL
resourcePoolType	Type of resource pool that is set up in Unified CCX Administration: 1—Resource groups 2—Resource skills	smallint NOT NULL
resourceGroupID	If resourcePoolType is 1, unique identifier used to locate the associated resource group in the Resource Group table. Not used if resourcePoolType is 2.	int NULL
selectionCriteria	Resource pool selection model that is set up in the Unified CCX Administration.	nvarchar(30) NOT NULL
skillGroupID	If resourcePoolType is 2, unique identifier used to locate the associated skill group in the SkillGroup table. Not used if resourcePoolType is 1.	int NULL
serviceLevel	Goal, in seconds, for the maximum time that a caller spends in the queue before the call is answered by an agent, as set up in Unified CCX Administration.	int NOT NULL
serviceLevelPercentage	Goal for the percentage of calls that meet the service level that is shown in the serviceLevel field, as set up in Unified CCX Administration.	smallint NOT NULL
active	Indicates whether the record is active: f — Inactive t — Active A record becomes inactive if the CSQ is deleted from the system or if the attributes are changed through the Unified CCX Administration user interface. When an attribute is changed, the record is marked inactive; that is, the active field is changed to “f”, and a new record is created.	boolean NOT NULL

Table 1-16 ContactServiceQueue Table Fields (continued)

Field Name	Description	Storage
autoWork	Whether an agent goes to Work State after handling a call from this CSQ: f —No t —Yes	boolean NOT NULL
dateInactive	If the active field is “f”, date and time that the record became inactive.	datetime year to fraction (3) NULL
queueAlgorithm	Criterion that specifies how contacts are queued, as set up in Unified CCX Administration.	nvarchar(30) NOT NULL
recordID	Identifier of this record. When any CSQ attribute, such as service level, is changed through the Unified CCX Administration user interface, the record is marked inactive; that is, the value of the active field changes to “f”, and a new record is created with a new record ID; the contactServiceQueueID stays the same for that CSQ.	int NOT NULL Primary Key
orderList	Reserved for future use.	int NULL
wrapupTime	Time in seconds that agent is placed in Work state. Possible values: 1 – 7200 0 – disabled	smallint NULL
prompt	The prompt value is used for remote monitoring. The customer can record the name of the CSQ and store it in a WAV file. This field contains the name of the WAV file.	lvarchar (256) NOT NULL
privateData	Any fields which are used internally only can be stored in this column in a blob.	BLOB NULL
queueType	A type of the CSQ. Possible values: 0 – voice CSQ 1 – email CSQ	smallint NOT NULL

Related Tables

- ContactQueueDetail, page 25 (recordID maps to targetID when targetType is 0, and via profileID)
- ProfileIDMapping, page 46 (via profileID)
- PurgeHistory, page 47 (recordID maps to origMonitoredID when origMonitoredIDType is 2, and via profileID)
- ResourceGroup, page 53 (via resourceGroupID and profileID)
- SkillGroup, page 61 (via skillGroupID and profileID)
- TeamCSQMapping, page 64 (contactServiceQueueID maps to csqID, and via profileID

CrsApplication

Database Table Name: CrsApplication

The CrsApplication table records application information. An application is uniquely identified by applicationName. When an application is created, a new record is inserted into this table. When an application is modified, the old record is marked as inactive, and a new record is inserted into the table with a new recordID. When an application is deleted, the corresponding record is marked as inactive.

The CrsApplication table contains the information shown in Table 1-17

Table 1-17 CrsApplication Table Fields

Field Name	Description	Storage
record ID	Unique numeric ID for each record. Introduced for historical reporting purposes. Possible values: 1, 2, 3....	int NOT NULL Primary Key
profileID	The indentifier of the profile	int NOT NULL Primary Key
applicationID	Configurable application identifier. Not unique for an application. Exposed for Cisco Unified Intelligent Contact Management Enterprise (Unified ICME) integration. Configured on Unified CCX Administration, modifiable. Possible values: -1, 1, 2, 3...	int NOT NULL
configClass	Represents application configuration class. Possible values: • com.cisco.app.ApplicationConfig • ApplicationConfig.class	lvarchar(512) NOT NULL
version	Specifies internal configuration schema version. Possible values: 1	int NOT NULL
configImplClass	Represents application configuration implementation class. Possible value: com.cisco.crs.app.ScriptApplicationConfig	lvarchar(512) NOT NULL
applicationName	Name that uniquely identifies the application	nvarchar(50) NOT NULL

Table 1-17 CrsApplication Table Fields (continued)

Field Name	Description	Storage
applicationType	The type of application. Possible values: • Busy • Ring-No-Answer • Cisco Script Application • Simulation Script • Unified ICME Post-Routing • Unified ICME Translation Routing	nvarchar(128) NOT NULL
applicationEnabled	Whether or not the application is enabled. Possible values: • f = disabled • t = enabled	boolean NOT NULL
numOfSessions	Maximum number of sessions	int NOT NULL
description	The description of the application that is configured in the Unified CCX Administration.	nvarchar(128) NULL
privateData	Internal data not exposed to customers.	BLOB NULL
createDateTime	The time when the record is created or updated. Default value: Current year to second	datetime year to second NOT NULL
active	Whether this record is active. Possible values: • f = inactive • t = active	boolean NOT NULL
dateInactive	If active = f, the time when this record became inactive.	datetime year to second NULL

Related Table

- ProfileIDMapping, page 46 (via profileID)

CrsGroup

Database Table Name: CrsGroup table

The CrsGroup table describes group information. A group is uniquely identified by the combination of groupClassName and groupID. When a group is created, a new record is inserted into this table. When a group is modified, the old record is marked as inactive, and a new record is inserted into the table with a new recordID. When a group is deleted, the corresponding record is marked as inactive.

The CrsGroup table contains the information shown in Table 1-18.

Table 1-18 CrsGroup Table Fields

Field Name	Description	Storage
recordID	A unique numeric ID for each record. Introduced for historical reporting purposes.	int NOT NULL Primary Key
profileID	Identifier of the profile. Possible values: 1, 2, 3...	int NOT NULL Primary Key
configClass	Represents Group configuration class. Possible values: GroupConfig.class	lvarchar(512) NOT NULL
version	Specifies internal configuration schema version. Possible values: 2	int NOT NULL
configImplClass	Represents group configuration implementation class. Possible values: • com.cisco.crs.email. • CiscoEmailControlGroupConfig.	lvarchar(512) NOT NULL
groupClass	Uniquely identifies a group together with the groupID. The class of channels being managed by the group.	lvarchar(400) NOT NULL
groupID	Uniquely identifies a group together with groupClassName. Group identifier unique for a give class of channels.	int NOT NULL
groupType	Type of the group, corresponding to type of the channels managed by the group as defined since CRS 3.0.	nvarchar(128) NOT NULL

groupEnabled	Whether the group is enabled.
	Possible values:
	f = disabled
	t = enabled
		boolean
		NOT NULL

numOfChannels	Number of channels defined in the group.	int NOT NULL
description	Description of the group.	nvarchar(128) NULL
privateData	Internal data not exposed to customers.	BLOB NULL

Table 1-18 CrsGroup Table Fields (continued)

Field Name	Description	Storage
createDateTime	When the group was created. Default value: Current year to second	datetime year to second NOT NULL

active	Whether this record is active.
	Possible values:
	f = inactive
	t = active
		boolean
		NOT NULL

dateInactive	If active = f, the time when the record became inactive.	datetime year to second NULL

Related Table

- ProfileIDMapping, page 46 (via profileID)

CrsTrigger

Database table name: CrsTrigger

The CrsTrigger table describes trigger information. A trigger is uniquely identified by a trigger name (triggerName). When a trigger is created, a new record is inserted into this table. When a trigger is modified, the old record is marked as inactive, and a new record is inserted into the table with a new recordID. When a trigger is deleted, the corresponding record will be marked as inactive.

The CrsGroup table contains the information shown in Table 1-19:

Table 1-19 CrsTrigger Table Fields

Field Name	Description	Storage
recordID	Unique numeric ID for each record. Introduced for historical reporting purposes.	int NOT NULL Primary Key
profileID	Identifier of the profile. Possible values: 1, 2, 3...	int NOT NULL Primary Key
configClass	Represents trigger configuration class. Possible values: ApplicationTriggerConfig.class	lvarchar(512) NOT NULL
version	Specifies internal configuration schema version. Possible values: 3	int NOT NULL

Table 1-19 CrsTrigger Table Fields (continued)

Field Name	Description	Storage
configImplClass	Represents trigger configuration implementation class. Possible values: • com.cisco.crs.email. • CiscoEmailControlGroupConfig	lvarchar(512) NOT NULL
triggerName	Uniquely identifies a trigger. Available from CRS 4.5 onwards. The API does limit the string length. Go back and revisit the length.	nvarchar(50) NOT NULL
triggerType	Hard coded. Possible values: • Cisco Http Trigger • Cisco JTAPI Trigger	nvarchar(128) NOT NULL
applicationName	Application name being triggered by the trigger.	nvarchar(50) NOT NULL
triggerEnabled	Whether the trigger is enabled Possible values: • f = disabled • t = enabled	boolean NOT NULL
numOfSessions	Maximum number of sessions Possible values: 0, 1, 2...	int NOT NULL
idleTimeout	Idle time out in milliseconds	int NOT NULL
triggerLocale	Default locale for the trigger. Possible values: • system.default (the currently configured system default locale) • accept.trigger (the locale provided by the incoming event)	nvarchar(50) NOT NULL
description	Description of the trigger	nvarchar(128) NULL
misc1	For HTTP trigger, this field contains the URL. For JTAPI and call triggers, this is the dialed number (DN).	lvarchar(256) NULL
misc2	For JTAPI trigger, this is the partition.	lvarchar(256) NULL
privateData	Internal data not exposed to customers, such as parameters or groups associated with a trigger.	BLOB NULL

Table 1-19 CrsTrigger Table Fields (continued)

Field Name	Description	Storage
createDateTime	When the trigger was created. Default value: Current year to second	datetime year to second NOT NULL

active	Whether this record is active.
	Possible values:
	f = inactive
	t = active
		boolean
		NOT NULL

dateInactive	If active = f, the time when the record became inactive.	datetime year to second NOT NULL

Related Table

- ProfileIDMapping, page 46 (via profileID)

DialingList

Database table name: DialingList

The DialingList table contains the outbound contacts that need to be dialed for a particular campaign. This table is populated when a text file containing the outbound contacts is imported from the Campaigns configuration page in the Unified CCX Administration.

When the outbound contacts are imported into the database from the Unified CCX Administration, the callStatus field has the default value of 1 (Pending); that is, the contacts are yet to be dialed.

The DialingList table contains the information shown in Table 1-20.

Table 1-20 DialingList Table Fields

Field Name	Description	Storage
recordid	A unique identifier for the record.	int NOT NULL Primary Key
dialinglistid	A unique identifier for a contact.	int NOT NULL Primary Key
profileid	Identifier of the Unified CCX profile that is associated with this record.	int NOT NULL Primary Key
campaignid	Campaign identifier	int NOT NULL

Table 1-20 DialingList Table Fields (continued)

Field Name	Description	Storage
createdatetime	Default -CURRENT_TIMESTAMP	datetime year to second NOT NULL
accountnumber	The account number of the contact (from the imported file). This field is sent to the agent desktop.	nvarchar(25) NULL
firstname	The first name of the contact (from the imported file).	nvarchar(50) NULL
lastname	The last name of the contact (from the imported file).	nvarchar(50) NULL
phone01	Primary phone number of the contact (from the imported file).	varchar(28) NOT NULL
phone02	Additional number of the contact (from the imported file). The number is dialed when the agent selects Skip-Next for the preview call.	varchar(28) NULL
phone03	Additional number of the contact (from the imported file). This number is dialed if attempts to dial the first two numbers are unsuccessful.	varchar(28) NULL
gmtzonephone01	The time zone for the first phone number of the contact.	smallint NOT NULL
dstphone01	0 = Daylight Savings Time (DST) is observed at this phone number. 1 = DST is not observed at this phone number	smallint NOT NULL
gmtzonephone02	The time zone for the second phone number of the contact.	smallint NOT NULL
dstphone02	0 = DST is observed at this phone number. 1 = DST is not observed at this phone number.	smallint NOT NULL
gmtzonephone03	The time zone for the third phone number of the contact.	smallint NOT NULL
dstphone03	0 = DST is observed at this phone number. 1 = DST is not observed at this phone number.	smallint NOT NULL
callbacknumber	Phone number to be used for callback (can be supplied by the agent).	varchar(28) NULL
callbackdatetime	Customer requested callback time.	datetime year to second NULL

Table 1-20 DialingList Table Fields (continued)

Field Name	Description	Storage
callstatus	The status of the contact record: 1 = Pending. The call is pending. 2 = Active. The record is sent (active) to the Outbound subsystem for dialing. 3 = Closed. The record is closed. 4 = Callback. The record is marked for a callback. 5 = Max Calls. Maximum attempts have been reached for this record (considered closed). 6 = Retry. The call is redialed immediately whenever there is any miss. 7 = Unknown. If the Outbound subsystem was restarted with records in the Active (2) state, the records are moved to this state. 8 = Retries with delay. The call is redialed as it was either busy, no answer, customer abandoned or system abandoned. Retry time is set as per the corresponding configuration in the Unified CCX Application Administration web interface.	smallint NOT NULL

Table 1-20 DialingList Table Fields (continued)

Field Name	Description	Storage
callresult	The call result from the last call placed for this record. 1 = Voice. Customer answered and was connected to agent. 2 = Fax. Fax machine reached. 3 = Answering machine. Answering machine reached. 4 = Invalid. Number reported as invalid by the network or by the agent. 5 = Do Not Call. Customer does not want to be called again. 6 = Wrong Number. Number successfully contacted but wrong number. 7 = Wrong Person. Number successfully contacted but reached the wrong person. 8 = Callback. Customer requested regular callback. 9 = Skip/Reject. Agent skipped or rejected a preview call. 10 = Skip-Close/Reject-Close. Agent skipped or rejected a preview call with the close option. 11 = Busy. Busy signal detected or marked busy by agent. 12 = Agent did not respond to the preview call within the timeout duration. 13 = Callback Failed - this value is not written to the database; this is for internal use only. 14 = Callback missed and marked for Retry. 15 = Customer's phone timed out either due to Ring No Answer (RNA) or Gateway failure. 16 = Call was abandoned because IVR port was unavailable or Unified CCX failed to transfer the call to the IVR port. 17 = Call failed due any one of the reasons. 18 = Customer abandoned as customer or agent disconnected the call within the time limit as configured in “Abandoned Call Wait Time” in Unified CCX Application Administration web interface.	smallint NOT NULL
callresult01	The call result from the last time phone01 was called. Values are the same as for callResult.	smallint NULL
callresult02	The call result from the last time phone02 was called. Values are the same as for callResult.	smallint NULL
callresult03	The call result from the last time phone03 was called. Values are the same as for callResult.	smallint NULL

Table 1-20 DialingList Table Fields (continued)

Field Name	Description	Storage
lastnumberdialed	The last number dialed. 1 = phone01 2 = phone02 3 = phone03	smallint NULL
callsmadetophone01	The number of call attempts made to phone01. If there is an error in an attempt to call this number, the attempt is not counted here.	smallint NULL
callsmadetophone02	The number of call attempts made to phone02. If there is an error in an attempt to call this number, the attempt is not counted here.	smallint NULL
callsmadetophone03	The number of call attempts made to phone03. If there is an error in an attempt to call this number, the attempt is not counted here.	smallint NULL
retry	Indicates whether the contact has to be retried.	boolean NULL
active	Contacts becomes inactive for a campaign in the following scenarios: 1 = delete a campaign 2 = delete all the contacts for a campaign 3 = when callStatus becomes 3 (closed) or 5 (max calls) f = Inactive t = Active	boolean NOT NULL
dateinactive	The date when record became inactive.	datetime year to second NULL
numMissedCallback	Number of missed callbacks.	smallint NULL

Related Tables

- AgentConnectionDetail, page 5 (via dialingListID and profileID)
- AreaCode, page 9 (via gmtZone)
- Campaign, page 10 (via campaignID and profile ID)
- ContactCallDetail, page 19 (via campaignID and profileID)
- DialingList, page 36 (via campaignID)
- ProfileIDMapping, page 46 (via profileID)

DialingListHistory

Database table name: DialingListHistory

The DialingListHistory table contains the outbound contacts that are inactive. As part of the automatic purge process, all the inactive records will get moved to this table from the DialingList table.

The DialingListHistory table contains the information shown in Table 1-21.

Table 1-21 DialingListHistory Table Fields

Field Name	Description	Storage
recordid	A unique identifier for the record.	int NOT NULL Primary Key
dialinglistid	A unique identifier for a contact.	int NOT NULL Primary Key
profileid	Identifier of the Unified CCX profile that is associated with this record.	int NOT NULL Primary Key
campaignid	Campaign identifier	int NOT NULL
createdatetime	Default -CURRENT_TIMESTAMP	datetime year to second NOT NULL
accountnumber	The account number of the contact (from the imported file). This field is sent to the agent desktop.	nvarchar(25) NULL
firstname	The first name of the contact (from the imported file).	nvarchar(50) NULL
lastname	The last name of the contact (from the imported file).	nvarchar(50) NULL
phone01	Primary phone number of the contact (from the imported file).	varchar(28) NOT NULL
phone02	Additional number of the contact (from the imported file). The number is dialed when the agent selects Skip-Next for the preview call.	varchar(28) NULL
phone03	Additional number of the contact (from the imported file). This number is dialed if attempts to dial the first two numbers are unsuccessful.	varchar(28) NULL
gmtzonephone01	The time zone for the first phone number of the contact.	smallint NOT NULL

Table 1-21 DialingListHistory Table Fields (continued)

Field Name	Description	Storage
dstphone01	0 = Daylight Savings Time (DST) is observed at this phone number. 1 = DST is not observed at this phone number	smallint NOT NULL
gmtzonephone02	The time zone for the second phone number of the contact.	smallint NOT NULL
dstphone02	0 = DST is observed at this phone number. 1 = DST is not observed at this phone number.	smallint NOT NULL
gmtzonephone03	The time zone for the third phone number of the contact.	smallint NOT NULL
dstphone03	0 = DST is observed at this phone number. 1 = DST is not observed at this phone number.	smallint NOT NULL
callbacknumber	Phone number to be used for callback (can be supplied by the agent).	varchar(28) NULL
callbackdatetime	Customer requested callback time.	datetime year to second NULL
callstatus	The status of the contact record: 1 = Pending. The call is pending. 2 = Active. The record is sent (active) to the Outbound subsystem for dialing. 3 = Closed. The record is closed. 4 = Callback. The record is marked for a callback. 5 = Max Calls. Maximum attempts have been reached for this record (considered closed). 6 = Retry. The call is redialed immediately whenever there is any miss. 7 = Unknown. If the Outbound subsystem was restarted with records in the Active (2) state, the records are moved to this state. 8 = Retries with delay. The call is redialed as it was either busy, no answer, customer abandoned or system abandoned. Retry time is set as per the corresponding configuration in the Unified CCX Application Administration web interface.	smallint NOT NULL

Table 1-21 DialingListHistory Table Fields (continued)

Field Name	Description	Storage
callresult	The call result from the last call placed for this record. 1 = Voice. Customer answered and was connected to agent. 2 = Fax. Fax machine reached. 3 = Answering machine. Answering machine reached. 4 = Invalid. Number reported as invalid by the network or by the agent. 5 = Do Not Call. Customer does not want to be called again. 6 = Wrong Number. Number successfully contacted but wrong number. 7 = Wrong Person. Number successfully contacted but reached the wrong person. 8 = Callback. Customer requested regular callback. 9 = Skip/Reject. Agent skipped or rejected a preview call. 10 = Skip-Close/Reject-Close. Agent skipped or rejected a preview call with the close option. 11 = Busy. Busy signal detected or marked busy by agent. 12 = Agent did not respond to the preview call within the timeout duration. 13 = Callback Failed - this value is not written to the database; this is for internal use only. 14 = Callback missed and marked for Retry. 15 = Customer's phone timed out either due to Ring No Answer (RNA) or Gateway failure. 16 = Call was abandoned because IVR port was unavailable or Unified CCX failed to transfer the call to the IVR port. 17 = Call failed due any one of the reasons. 18 = Customer abandoned as customer or agent disconnected the call within the time limit as configured in “Abandoned Call Wait Time” in Unified CCX Application Administration web interface.	smallint NOT NULL
callresult01	The call result from the last time phone01 was called. Values are the same as for callResult.	smallint NULL
callresult02	The call result from the last time phone02 was called. Values are the same as for callResult.	smallint NULL
callresult03	The call result from the last time phone03 was called. Values are the same as for callResult.	smallint NULL

Table 1-21 DialingListHistory Table Fields (continued)

Field Name	Description	Storage
lastnumberdialed	The last number dialed. 1 = phone01 2 = phone02 3 = phone03	smallint NULL
callsmadetophone01	The number of call attempts made to phone01. If there is an error in an attempt to call this number, the attempt is not counted here.	smallint NULL
callsmadetophone02	The number of call attempts made to phone02. If there is an error in an attempt to call this number, the attempt is not counted here.	smallint NULL
callsmadetophone03	The number of call attempts made to phone03. If there is an error in an attempt to call this number, the attempt is not counted here.	smallint NULL
retry	Indicates whether the contact has to be retried.	boolean NULL
active	Contacts becomes inactive for a campaign in the following scenarios: 1 = delete a campaign 2 = delete all the contacts for a campaign 3 = when callStatus becomes 3 (closed) or 5 (max calls) f = Inactive t = Active	boolean NOT NULL
dateinactive	The date when record became inactive.	datetime year to second NULL
numMissedCallback	Number of missed callbacks.	smallint NULL

Related Tables

- AgentConnectionDetail, page 5 (via dialingListID and profileID)
- AreaCode, page 9 (via gmtZone)
- Campaign, page 10 (via campaignID and profile ID)
- ContactCallDetail, page 19 (via campaignID and profileID)
- DialingList, page 36 (via campaignID)
- ProfileIDMapping, page 46 (via profileID)

MonitoredResourceDetail

Database table name: MonitoredResourceDetail

The MonitoredResourceDetail table records the actual agents who are monitored. The RemoteMonitoringDetail table records the original agent or the CSQ that the supervisor plans to monitor. Monitoring a CSQ involves monitoring the agents who handle calls for that CSQ. So the actual agents (which can be more than one) that are monitored will be recorded in the MonitoredResourceDetail table.

The MonitoredResourceDetail table contains the fields shown in Table 1-22.

Table 1-22 MonitoredResourceDetail Table Fields

Field Name	Description	Storage
sessionid	Identifier that the system assigned to the call. This identifier remains the same for all legs of the call. It is the sessionID of the IVR call; that is, when the supervisor starts monitoring, the monitoring call itself is an IVR call. The supervisor monitors one or more Unified CCX calls.	decimal(18) NOT NULL Primary Key
startmonitoringreqtime	The time and date that the remote supervisor attempted to monitor the agent.	datetime year to fraction (3) NOT NULL
startmonitoringcalltime	The time and date that the supervisor began monitoring the call.	datetime year to fraction (3) NOT NULL Primary Key
monitoredrsrcid	Identifier of the resource being monitored.	int NOT NULL
monitoredsessionseqnum	The session sequence number of the Unified CCX call that is being monitored.	smallint NOT NULL
profileid	Identifier of the Unified CCX profile that is associated with this record.	int NOT NULL Primary Key
gmtoffset	Offset, in minutes, between the local time of the Unified CCX server and Greenwich Mean Time. As the time information is stored in GMT, this field will always be zero.	int NOT NULL
nodeid	Unique identifier assigned to each server in the cluster.	smallint NOT NULL Primary Key

Related Tables

- AgentConnectionDetail, page 5 (monitoredSessionSeqNum maps to sessionSeqNum, via nodeID, profileID)
- ContactCallDetail, page 19 (monitoredSessionSeqNum maps to sessionSeqNum, via sessionID, nodeID, profileID)

- ContactQueueDetail, page 25 (monitoredSessionSeqNum maps to sessionSeqNum, via nodeID, profileID)
- ContactRoutingDetail, page 27 (monitoredSessionSeqNum maps to sessionSeqNum, via nodeID, profileID)
- PurgeHistory, page 47 (via sessionID, profileID, nodeID, startMonitoringReqTime)
- Resource, page 51 (monitoredRsrcID maps to resourceID, via profileID)

MediaCustomerDataMapping

Database table name: MediaCustomerDataMapping

The MediaCustomerDataMapping table contains mapping between the customer data fields for a specific media (such as email and other media types) and the order of fields stored in the TextCustomerDetail table.

The MediaCustomerDataMapping table contains the fields shown in Table 1-23.

Note Do not edit this table directly. It is for internal use only

Table 1-23 MediaCustomerDataMapping Table Fields

Field Name	Description	Storage
recordId	Unique identifier for the record.	int NOT NULL Primary Key
mediaType	Type of the media such as email and other media types.	varchar(30) NOT NULL Primary Key
fieldName	Name of the field in the customer data.	varchar(50) NOT NULL
columnId	Field ID in the TextCustomerDetail where this field is stored.	int NOT NULL

Related Tables

- TextCustomer Detail, page 71 (via columnId)

ProfileIDMapping

Database table name: ProfileIDMapping

The Unified CCX system creates a new record in the ProfileIDMapping table when a new profile is set up in the Unified CCX Administration.

A ProfileIDMapping record shows the mapping of the profile name to its unique identifier.

The ProfileIDMapping table contains the information shown in Table 1-24.

Table 1-24 ProfileIDMapping Table Fields

Field Name	Description	Storage
profileName	Name of the profile, as set up in the Unified CCX Administration.	nvarchar(50) NOT NULL Primary Key
profileID	Identifier of the profile.	int NOT NULL

Related Tables

- AgentConnectionDetail, page 5 (via ProfileID)
- AgentStateDetail, page 7 (via profileID)
- AreaCode, page 9 (via profileID)
- Campaign, page 10 (via profile ID)
- ContactCallDetail, page 19 (via profileID)
- ContactQueueDetail, page 25 (via profileID)
- ContactRoutingDetail, page 27 (via profileID)
- ContactServiceQueue, page 28 (via profileID)
- DialingList, page 36 (via profile ID)
- MonitoredResourceDetail, page 44 (via profileID)
- PurgeHistory, page 47 (via profileID)
- Resource, page 51 (via ProfileID)
- ResourceGroup, page 53 (via profileID)
- ResourceSkillMapping, page 53 (via profileID)
- RmonCSQConfig, page 54 (via profileID)
- RmonResConfig, page 55 (via profileID)
- RmonUser, page 56 (via profileID)
- Skill, page 60 (via ProfileID)
- SkillGroup, page 61 (via profileID)
- Supervisor, page 62 (via profileID)
- Team, page 63 (via profileID)
- TeamCSQMapping, page 64 (via profileID)

PurgeHistory

Database table name: PurgeHistory

PurgeHistory is mainly to keep track of the history of purge information for both Manual and Scheduled purge.

The PurgeHistory table contains the information shown in Table 1-25.

Table 1-25 PurgeHistory Table Fields

Field Name	Description	Storage
nodeId	Unique identifier assigned to each server in the cluster.	int NOT NULL Primary Key
purgeHistoryId	Sequence numbers.	int NOT NULL Primary Key
purgeType	PurgeType MANUAL or SCHEDULED.	nvarchar(10)
purgeState	PurgeState can be any one of, RUNNING, COMPLETED_SUCCESSFULLY, COMPLETED_WITH_ERRORS, UNKNOWN.	nvarchar(30)
purgeStartedDateTime	Purge start time.	datetime year to fraction(3)
hrDbSizeBeforePurge	Historical db size before purge which will have the value using store procedure getDbSize with column name as "used”.	int
configDbSizeBeforePurge	Config db size before purge which will have the value using store procedure getDBspaceUsage('db_cra') with column name as "used".	int
oldestRecDateTimeBeforePurge	Oldest record date and time before purge.	datetime year to fraction(3)
purgeCompletedDateTime	Purge completion time.	datetime year to fraction(3)
hrDbSizeAfterPurge	Historical db size after purge which will have the value using store procedure getDbSize with column name as "used"	int
configDbSizeAfterPurge	Config db size after purge which will have the value using store procedure getDBspaceUsage('db_cra') with column name as "used"	int
oldestRecDateTimeAfterPurge	Oldest record date time after purge.	datetime year to fraction(3)
purgetRunTime	Purge run time in minutes which is the difference between purgeCompletedDataTime and purgeStartedDateTime	int

RemoteMonitoringDetail

Database table name: RemoteMonitoringDetail

The Remote Monitoring Detail Record provides information about sessions where remote monitoring is used.

The RemoteMonitoringDetail table contains the fields shown in Table 1-26.

Table 1-26 RemoteMonitoringDetail Table Fields

Field Name	Description	Storage
sessionid	Identifier that the system assigned to the call. This identifier remains the same for all legs of the call. This is the sessionID of the IVR call; that is, the call that the supervisor makes to monitor other Unified CCX calls.	decimal(18) NOT NULL Primary Key
startmonitoringreqtime	The time and date that the remote supervisor attempted to monitor the agent.	datetime year to fraction (3) NOT NULL Primary Key
remoteloginid	The numeric ID the supervisor enters before starting to monitor a call	varchar(50) NOT NULL
rmonid	Numeric ID of the supervisor who does the monitoring.	int NOT NULL
endmonitoringtime	The date and time the monitoring ended.	datetime year to fraction (3) NOT NULL
origmonitoredid	If origMonitoredIDType is: • 1 (agent), this field contains the extension of the agent being monitored. • 2 (CSQ), this field contains the CSQ ID of the CSQ being monitored.	int NOT NULL
origmonitoredidtype	Indicates an agent or a CSQ. 1 = agent 2 = CSQ	smallint NOT NULL

Table 1-26 RemoteMonitoringDetail Table Fields (continued)

Field Name	Description	Storage
cause	The termination cause of a monitoring session: • 3 = Normal (Monitored) • 100 = Normal (Agent RNA) • 0 = Error (Other) • –9 = Error (Unable to Stop Monitoring) • –8 = Error (Unable to Monitor New Call) • –7 = Error (Agent Logged Off) • –6 = Error (Network Problem) • –5 = Error (VoIP Server unable to communicate) • –4 = Error (Monitoring not allowed) • –3 = Error (Agent not logged in) • –2 = Error (Invalid input) • –1 = Error (Other)	smallint NULL
sessionSeqNum	The sequence number for the IVR call; that is, the call the supervisor makes to monitor other Unified CCX calls.	smallint NOT NULL
monitoredSessionID	The sessionID of the monitored Unified CCX call.	decimal(18) NOT NULL
profileID	Identifier of the Unified CCX profile that is associated with this record.	int NOT NULL Primary Key
gmtOffset	Offset, in minutes, between the local time of the Unified CCX server and Greenwich Mean Time. As the time information is stored in GMT, this field will always be zero.	int NOT NULL
nodeID	Unique identifier assigned to each server in the cluster.	smallint NOT NULL Primary Key

Related Tables

- AgentConnectionDetail, page 5 (monitoredSessionID maps to sessionID, via nodeID, profileID)
- ContactCallDetail, page 19 (monitoredSessionID maps to sessionID, via sessionID, sessionSeqNum, nodeID, profileID)
- ContactQueueDetail, page 25 (monitoredSessionID maps to sessionID, via nodeID, profileID)
- ContactRoutingDetail, page 27 (monitoredSessionID maps to sessionID, via nodeID, profileID)
- ContactServiceQueue, page 28 (origMonitoredID maps to contactServiceQueueID when origMonitoredIDType is 2, via profileID)
- MonitoredResourceDetail, page 44 (via sessionID, profileID, nodeID, startMonitoringReqTime)

- RmonUser, page 56 (remoteLoginID maps to loginID, via profileID, rmonID)

Resource

Database table name: Resource

The Unified CCX system creates a new record in the Resource table when the Unified CCX system retrieves agent information from the Unified CM.

A Resource record contains information about the resource (agent). One such record exists for each active and inactive resource. When a resource is deleted, the old record is flagged as inactive; when a resource is updated, a new record is created and the old one is flagged as inactive.

The Resource table contains the fields shown in Table 1-27.

Table 1-27 Resource Table Fields

Field Name	Description	Storage
resourceID	Numeric identifier of the resource.	int NOT NULL Primary Key
profileID	Identifier of the Unified CCX profile that is associated with this record.	int NOT NULL Primary Key
resourceLoginID	The login name assigned to the resource in the Unified CM.	nvarchar(50) NOT NULL
resourceName	The first name and the last name of the resource.	nvarchar(50) NOT NULL
resourceGroupID	Resource group to which the resource belongs. Null if no resource group is assigned to the resource.	int NULL
resourceType	Type of the resource: 1—Agent 2—Supervisor 3—Administrator	smallint NOT NULL
active	Whether this record is active: f —Inactive t —Active A record becomes inactive if the resource is deleted or updated.	boolean NOT NULL
autoAvail	Determines whether the resource goes to Ready State after handling a Unified CCX call: f —No t —Yes	boolean NOT NULL

Table 1-27 Resource Table Fields (continued)

Field Name	Description	Storage
extension	The Unified CCX extension of the resource.	nvarchar(50) NOT NULL
orderInRG	Order in which the resource resides within the resource group. Null if no resource group is assigned to the resource.	int NULL
dateInactive	If the active field is “f”, date and time that the record became inactive.	datetime year to fraction(3) NULL
resourceSkillMapID	Identifier used to locate the associated skill set of the resource in the ResourceSkillMapping table. The ResourceSkillMapping table can contain multiple records for one resource.	int NOT NULL
assignedTeamID	Identifier of the resource’s assigned team.	int NOT NULL
resourceFirstName	The resource’s first name.	nvarchar(50) NOT NULL
resourceLastName	The resource’s last name.	nvarchar(50) NOT NULL
resourceAlias	The resource’s alias name.	nvarchar (50) NULL

Related Tables

- AgentConnectionDetail, page 5 (via resourceID, profileID)
- AgentStateDetail, page 7 (resourceID maps to agentID, via profileID)
- ContactCallDetail, page 19 (resourceID maps originatorID/destinationID when originatorType/destinageType is 1, via profileID)
- ContactQueueDetail, page 25 (resourceID maps to targetID when targetType is 1, via profileID)
- MonitoredResourceDetail, page 44 (resourceID maps to monitoredRsrcID, via profileID)
- ProfileIDMapping, page 46 (via profileID)
- PurgeHistory, page 47 (resourceID maps to origMonitoredID when origMonitoredIDType is 1, via profileID)
- ResourceGroup, page 53 (via resourceGroupID, profileID)
- ResourceSkillMapping, page 53 (via resourceSkillMapID, profileID)
- Supervisor, page 62 (via resourceLoginID, profileID)
- Team, page 63 (assignedTeamID maps to teamID, via profileID)

ResourceGroup

Database table name: ResourceGroup

The Unified CCX system creates a new record in the ResourceGroup table when a resource group is set up in the Unified CCX Administration.

A ResourceGroup record contains information about the resource group. One such record exists for each active and inactive resource group.

The Resource Group table contains the fields shown in Table 1-28.

Table 1-28 Resource Group Table Fields

Field Name	Description	Storage
resourceGroupID	Numeric identifier of the resource group.	int NOT NULL Primary Key
profileID	Identifier of the Unified CCX profile that is associated with this record.	int NOT NULL Primary Key
resourceGroupName	Name of the resource group, as set up in the Unified CCX Administration.	nvarchar(50) NULL
active	Whether the record is active in the Unified CCX system: f —Inactive t —Active A record becomes inactive if the resource group is deleted or updated.	boolean NOT NULL
dateInactive	If the active field is “f”, date and time that the record became inactive.	datetime year to fraction(3) NULL

Related Tables

- ContactServiceQueue, page 28 (via resourceGroupID, profileID)
- ProfileIDMapping, page 46 (via profileID)
- Resource, page 51 (via resourceGroupID, profileID)

ResourceSkillMapping

Database table name: ResourceSkillMapping

The Unified CCX system creates a new record in the ResourceSkillMapping table when an agent is associated with a skill in the Unified CCX Administration.

A ResourceSkillMapping record contains information about all of the skills that are assigned to resources.

The ResourceSkillMapping table contains the fields shown in the below table.

Table 1-29 ResourceSkillMapping Table Fields

Field Name	Description	Storage
resourceSkillMapID	Identifier of the skill set that is associated with a resource.	int NOT NULL Primary Key
skillID	Identifier of the skill that is associated with a resource.	int NOT NULL Primary Key
profileID	Identifier of the Unified CCX profile that is associated with this record.	int NOT NULL Primary Key
competenceLevel	Competence level associated with the skill, as set up in the Unified CCX Administration. Values range from 1 (lowest) to 10 (highest).	smallint NOT NULL

Related Tables

- ProfileIDMapping, page 46 (via profileID)
- Resource, page 51 (via resourceSkillMapID and profileID)
- Skill, page 60 (via skillID and profileID)

RmonCSQConfig

Database table name: RmonCSQConfig

The Remote Monitoring Contact Service Queue Configuration table contains the CSQs that a remote monitoring supervisor is allowed to monitor (the supervisor's allowed list). This table is updated when you configure the Unified CCX system through the Unified CCX Administration pages.

The RmonCSQConfig table contains the fields shown in Table 1-30.

Table 1-30 RmonCSQConfig Table Fields

Field Name	Description	Storage
rmonID	Numeric identifier of the remote supervisor.	int NOT NULL Primary Key

Table 1-30 RmonCSQConfig Table Fields (continued)

Field Name	Description	Storage
contactServiceQueueID	The numeric identifier of the CSQ, relating to contactServiceQueueID in the ContactServiceQueue table.	int NOT NULL Primary Key
profileID	Identifier of the Unified CCX profile that is associated with this record.	int NOT NULL Primary Key

Related Tables

- ContactServiceQueue, page 28 (via contactServiceQueueID, and profileID)
- ProfileIDMapping, page 46 (via profileID)
- RmonResConfig, page 55 (via rmonID and profileID)
- RmonUser, page 56 (via rmonID and profileID

RmonResConfig

Database table name: RmonResConfig

The Remote Monitoring Resource Configuration table contains the list of the agents (resources) that a remote monitoring supervisor is allowed to monitor (the supervisor's allowed list). This table is updated when you configure the system through the Unified CCX Administration pages.

The RmonResConfig table contains the fields shown in Table 1-31.

Table 1-31 RmonResConfig Table Fields

Field Name	Description	Storage
rmonID	Numeric identifier of the remote supervisor.	int NOT NULL Primary Key
resourceLoginID	The login ID of the resource that the remote supervisor is allowed to monitor.	nvarchar(50) NOT NULL Primary Key
profileID	Identifier of the Unified CCX profile that is associated with this record.	int NOT NULL Primary Key

Related Tables

- ProfileIDMapping, page 46 (via ProfileID)
- Resource, page 51 (via resourceLoginID, profileID
- RmonCSQConfig, page 54 (via rmonID, profileID)

- RmonUser, page 56 (via rmonID, profileID)

RmonUser

Database table name: RmonUser

The Remote Monitoring User table provides information about the supervisor who is logged in to remotely monitor agents.

The RmonUser table contains the fields shown in Table 1-32.

Table 1-32 RmonUser Table Fields

Field Name	Description	Storage
rmonID	Numeric identifier of the remote supervisor.	int NOT NULL Primary Key
LoginID	User login name of the remote supervisor.	nvarchar(50) NOT NULL
name	Name of the supervisor.	nvarchar(50) NOT NULL
profileID	Identifier of the Unified CCX profile that is associated with this record.	int NOT NULL Primary Key
type	The type of supervisor: 0 = regular supervisor 1 = remote monitoring supervisor	int NOT NULL

active	Determines whether the remote supervisor is active.
	f = inactive
	t = active
		boolean
		NOT NULL

dateInactive	Date and time the remote supervisor became inactive.	datetime year to second NULL

Related Tables

- ProfileIDMapping, page 46 (via profileID)
- PurgeHistory, page 47 (loginID maps to remoteLoginID, via profileID, and rmonID
- RmonCSQConfig, page 54 (via rmonID and profileID)
- RmonResConfig, page 55 (via rmonID and profileID)

RtCSQsSummary

Database table name: RtCSQsSummary

The rtcsqssummary table contains real-time statistics about all configured Contact Service Queues in the system. This table gets updated automatically when real-time snapshot data writing for this table is enabled through the Unified CCX Administration pages (**Tools** > **Real-time snapshot configuration** menu option). The updating frequency is based on the configured data writing interval.

The RtCSQsSummary table contains the fields shown in Table 1-33.

Table 1-33 RtCSQsSummary Table Fields

Field Name	Description	Storage
csqname	Name of the contact service queue.	nvarchar(50) NOT NULL Primary Key
loggedinagents	Number agents who are logged in.	int NOT NULL
availableagents	Number of available (idle) agents.	int NOT NULL
unavailableagents	Number of unavailable agents.	int NOT NULL
totalcalls	Total number of calls.	int NOT NULL
oldestcontact	Oldest contact in the queue.	int NOT NULL
callshandled	Number of calls handled.	int NOT NULL
callsabandoned	Number of calls abandoned.	int NOT NULL
callsdequeued	Number of calls dequeued.	int NOT NULL
avgtalkduration	Average talk duration.	int NOT NULL
avgwaitduration	Average wait duration.	int NOT NULL
longesttalkduration	Longest talk duration.	int NOT NULL
longestwaitduration	Longest wait duration.	int NOT NULL

Table 1-33 RtCSQsSummary Table Fields (continued)

Field Name	Description	Storage
callswaiting	Number of calls waiting.	int NOT NULL
enddatetime	The date and time that this table data was last updated.	datetime year to second NOT NULL
workingagents	Number of agents who are in the working state.	int NOT NULL
talkingagents	Number of agents who are in the talking state.	int NOT NULL
reservedagents	Number of agents who are in the reserved state.	int NOT NULL
startdatetime	The date and time that this table's statistics get collected.	datetime year to second NOT NULL
convavgtalkduration	Average talk duration in HH:MM:SS format.	varchar(25) NOT NULL
convavgwaitduration	Average wait duration in HH:MM:SS format.	varchar(25) NOT NULL
convlongesttalkduration	Longest talk duration in HH:MM:SS format.	varchar(25) NOT NULL
convlongestwaitduration	Longest wait duration in HH:MM:SS format.	varchar(25) NOT NULL
convoldestcontact	Oldest call in the queue in HH:MM:SS format.	varchar(25) NOT NULL

Related Tables

None.

RtICDStatistics

Database table name: RtICDStatistics

The RtICDStatistics table contains real-time summary statistics about Unified CCX. This table gets updated automatically when real-time snapshot data writing for this table is enabled through the Unified CCX Administration pages (**Tools > Real-time snapshot configuration** menu option.) The updating frequency is based on the configured data writing interval.

The RtICDStatistics table contains the fields shown in Table 1-34.

Table 1-34 RtICDStatistics Table Fields

Field Name	Description	Storage
type	Contact Service Queue type that identifies the contact type it services. It can be either voice or e-mail.	nvarchar (50) NOT NULL Primary Key
totalcsqs	Number of CSQs configured.	int NOT NULL
loggedinagents	Number of agents who are logged in.	int NOT NULL
workingagents	Number of agents who are in the working state.	int NOT NULL
reservedagents	Number of agents who are in the reserved state.	int NOT NULL
talkingagents	Number of agents who are in the talking state.	int NOT NULL
availableagents	Number of available (idle) agents.	int NOT NULL
unavailableagents	Number of unavailable agents.	int NOT NULL
totalcalls	Total number of calls.	int NOT NULL
callswaiting	Number of calls waiting.	int NOT NULL
callshandled	Number of calls handled.	int NOT NULL
callsabandoned	Number of calls abandoned.	int NOT NULL
avgtalkduration	Average talk duration.	int NOT NULL
avgwaitduration	Average wait duration.	int NOT NULL
longesttalkduration	Longest talk duration.	int NOT NULL
longestwaitduration	Longest wait duration.	int NOT NULL
oldestcontact	Oldest contact in the queue.	int NOT NULL

Table 1-34 RtICDStatistics Table Fields (continued)

Field Name	Description	Storage
startdatetime	Data collection starting time.	datetime year to second NOT NULL
enddatetime	Date and time this table was last updated.	datetime year to second NOT NULL
convavgtalkduration	Average talk duration in HH:MM:SS format.	varchar(25) NOT NULL
convavgwaitduration	Average wait duration in HH:MM:SS format	varchar(25) NOT NULL
convlongesttalkduration	Longest talk duration in HH:MM:SS format.	varchar(25) NOT NULL
convlongestwaitduration	Longest wait duration in HH:MM:SS format.	varchar(25) NOT NULL
convoldestcontact	Oldest call in the queue in HH:MM:SS format.	varchar(25) NOT NULL

Related Tables

None.

Skill

Database table name: Skill

The Unified CCX system creates a new record in the Skill table when a skill is set up in the Unified CCX Administration.

A Skill record contains information about a skill. One such record exists for each configured skill.

The Skill table contains the fields shown in Table 1-35.

Table 1-35 Skill Table Fields

Field Name	Description	Storage
skillID	Numeric identifier of the skill.	int NOT NULL Primary Key
profileID	Identifier of the Unified CCX profile that is associated with this record.	int NOT NULL Primary Key

Table 1-35 Skill Table Fields (continued)

Field Name	Description	Storage
skillName	Name of the skill, as set up in the Unified CCX Administration.	nvarchar(50) NOT NULL
active	Determines whether the record is active in the Unified CCX system: f —Inactive t —Active A record becomes inactive if the skill is deleted or updated.	boolean NOT NULL
dateInactive	If the active field is “f”, date and time that the record became inactive.	datetime year to fraction(3) NULL

Related Tables

- ProfileIDMapping, page 46 (via profileID)
- ResourceSkillMapping, page 53 (via skillID and profileID)
- SkillGroup, page 61 (via skillID and profileID)

SkillGroup

Database table name: SkillGroup

The Unified CCX system creates a new record in the SkillGroup table when skills are associated with a CSQ in the Unified CCX Administration.

A SkillGroup record describes each skill that is associated with the CSQ.

The SkillGroup table contains the fields shown in Table 1-36.

Table 1-36 Skill Group Table Fields

Field Name	Description	Storage
skillGroupID	Numeric identifier of the skill group.	int NOT NULL Primary Key
skillID	Numeric identifier of the skill.	int NOT NULL Primary Key
profileID	Identifier of the Unified CCX profile that is associated with this record.	int NOT NULL Primary Key
competenceLevel	Minimum acceptable skill level for agents with this skill, as set up in the Unified CCX Administration. Values range from 1 (lowest) to 10 (highest).	smallint NOT NULL

Table 1-36 Skill Group Table Fields (continued)

Field Name	Description	Storage
active	Determines whether the record is active in the CSQ: f —Inactive t —Active A record becomes inactive if the new skill group is deleted or updated.	boolean NOT NULL
skillWeight	Skills within a CSQ can be assigned weights. This field is used in the weighted skill calculation of the skill-based resource selection algorithm. Default value is 1.	int NOT NULL
skillOrder	Skills within a CSQ can be ordered. This field is used in the order skill calculation of the skill-based resource selection algorithm. Default value is 1.	int NOT NULL

Related Tables

- ContactServiceQueue, page 28 (via skillGroupID and profileID)
- ProfileIDMapping, page 46 (via profileID)
- Skill, page 60 (via skillID and profileID)

Supervisor

Database table name: Supervisor

The Supervisor table contains the information about the supervisor.

The Supervisor table contains the fields shown in Table 1-37.

Table 1-37 Supervisor Table Fields

Field Name	Description	Storage
recordID	Numeric identifier of this supervisor.	int NOT NULL Primary Key
resourceLoginID	User ID in the Unified CM configuration.	nvarchar(50) NOT NULL
managedTeamID	Team identifier of the managed team.	int NOT NULL
profileID	Identifier of the Unified CCX profile that is associated with this record.	int NOT NULL Primary Key

Table 1-37 Supervisor Table Fields

Field Name	Description	Storage
supervisorType	Type of supervisor for this team 0 = Primary 1 = Secondary	smallint NOT NULL

active	Indicates whether the record is active in the Unified CCX system. A record becomes inactive if a team is deleted or updated.
	f = Inactive
	t = Active
		boolean
		NOT NULL

dateInactive	Date this record was deleted.	datetime year to second NULL

Related Tables

- Resource, page 51 (via resourceLoginID and profileID
- ProfileIDMapping, page 46 (via ProfileID)
- Team, page 63 (managedTeamID maps to teamID, via profileID)

Team

Database table name: Team

The Team table contains information about specific teams.

The fields in the Team table are shown in Table 1-38.

Table 1-38 Team Table Fields

Field Name	Description	Storage
teamID	Numeric identifier for this team.	int NOT NULL Primary Key
profileID	Identifier of the Unified CCX profile that is associated with this record.	int NOT NULL Primary Key
teamName	Name of this team.	nvarchar(50) NOT NULL

Table 1-38 Team Table Fields (continued)

Field Name	Description	Storage

active	Indicates whether the record is active in the Unified CCX system. A record becomes inactive if a team is deleted or updated.
	f = Inactive
	t = Active
		boolean
		NOT NULL

dateInactive	Date this record was deleted.	datetime year to fraction(3) NULL

Related Tables

- ProfileIDMapping, page 46 (via ProfileID)
- Resource, page 51 (teamID maps to assignedTeamID and via profileID
- Supervisor, page 62 (teamID maps to managedTeamID and via profileID)
- TeamCSQMapping, page 64 (via teamID and profileID)

TeamCSQMapping

Database table name: TeamCSQMapping

The TeamCSQMapping table shows the relationship between Teams and CSQs; for example, Team 1 is CSQ3, Team 4 is CSQ10.

The TeamCSQMapping table contains the fields shown in Table 1-39.

Table 1-39 TeamCSQMapping Table Fields

Field Name	Description	Storage
recordID	Numeric identifier for this record.	int NOT NULL Primary Key
csqID	Numeric identifier for the CSQ.	int NOT NULL
teamID	Numeric identifier for the team.	int NOT NULL
profileID	Identifier of the Unified CCX profile that is associated with this record.	int NOT NULL Primary Key

Table 1-39 TeamCSQMapping Table Fields (continued)

Field Name	Description	Storage

active	Indicates whether the record is active in the Unified CCX system. A record becomes inactive if a team is deleted or updated.
	f = Inactive
	t = Active
		boolean
		NOT NULL

dateInactive	Date this record was deleted.	datetime year to second NULL

Related Tables

- ContactServiceQueue, page 28 (csqID maps to contactServiceQueueID, and via profileID)
- ProfileIDMapping, page 46 (via ProfileID)
- Team, page 63 (via teamID and profileID)

TextAgentConnectionDetail

Database table name: TextAgentConnectionDetail

The Unified CCX system creates a new record in the TextAgentConnectionDetail table when an agent disconnects a contact or a leg by hanging up or by transferring the contact. A new contact or leg starts each time that a contact is transferred.

The TextAgentConnectionDetail record contains information relating to the agent who handled the contact or leg.

The TextAgentConnectionDetail table contains the fields shown in Table 1-40.

Table 1-40 TextAgentConnectionDetail Table Fields

Field Name	Description	Storage
ContactID	Alphanumeric identifier for the contact.	varchar (64) NOT NULL Primary Key
ContactSeqNum	Contact sequence number that the system assigned to the contact or the leg. Each leg of a contact is assigned a new contact sequence number. To be used later.	smallint NOT NULL Primary Key
nodeID	Numeric identifier for the node.	smallint NOT NULL
resourceID	Numeric identifier for the resource.	int NOT NULL Primary Key

Table 1-40 TextAgentConnectionDetail Table Fields (continued)

Field Name	Description	Storage
startDateTime	Date and time that the contact or leg entered the system.	datetime year to fraction (3) NOT NULL Primary Key
endDateTime	Date and time that the contact or the leg was transferred or disconnected.	datetime year to fraction (3) NOT NULL
qIndex	A new qIndex is created whenever a Unified CCX contact is conferenced to a Unified CCX route point. To be used later.	smallint NOT NULL
acceptTime	Amount of time, in seconds, that passed from the time a contact or leg was presented to an agent and the agent answered the contact.	int NULL
talkTime	Amount of time, in seconds, that passed from the time an agent answered the contact or the leg to the time the contact or the leg was disconnected or transferred, not including hold time.	smallint NULL
workTime	Amount of time, in seconds, that an agent spent in Work State after the contact or the leg. To be used later.	smallint NULL
WrapupData	After-contact information that the agent enters through the Agent Desktop user interface while the agent is in the work state. To be used later.	varchar(40) NULL

Related Tables

- TextContactDetail, page 68 (via ContactID, ContactSeqNum, and nodeID)
- TextContactQueueDetail, page 70 (via ContactID, ContactSeqNum, nodeID, and qIndex)

TextAgentStateDetail

Database table name: TextAgentStateDetail

The Unified CCX system creates a new record in the TextAgentStateDetail table each time the state of the agent changes while the agent is handling chat and email. The TextAgentStateDetail record contains information about the agent and about the event that caused the agent state to change.

The TextAgentStateDetail table contains the fields shown in the below table.

Table 1-41 TextAgentStateDetail Table Fields

Field Name	Description	Storage
agentID	Identifier of the agent whose state has changed.	int NOT NULL Primary Key
stateChangeDatetime	Date and time that the chat agent state changed.	datetime year to fraction (3) NOT NULL Primary Key
agentStateID	Event that triggered the chat agent state change: 0—Logon 1—Log off 2—Not available 3—Available 4—Busy 5—Unknown	smallint NOT NULL Primary Key
reasonCode	Code, as written to the database, for the reason that the chat agent changed to Not Ready state or to Log Out state. 32750—Non chat agent 32755—Contact ended 32757—Media handler failure 32760—Login 32763—Contact not accepted 32764—CCX failure 32765—Connection down	smallint NOT NULL Primary Key
routingType	Routing type of the contact or leg: 1—Interactive 2—Non Interactive	smallint NOT NULL Primary Key

Related Table

- Resource, page 51 (via agentID)

TextContactDetail

Database table name: TextContactDetail

The Unified CCX system creates a new record in the TextContactDetail table for each chat and email contact or leg processed by the system. A new contact or leg starts each time a contact is transferred or redirected.

A TextContactDetail record contains detailed information about the contact or leg. At least one such record will exist for each contact or leg.

The TextContactDetail table contains the fields shown in Table 1-42.

Table 1-42 TextContactDetail Table Fields

Field Name	Description	Storage
ContactID	Alphanumeric identifier for this record.	varchar (64) NOT NULL Primary Key
ContactSeqNum	Contact sequence number that the system assigned to the contact or the leg. Each leg of a contact is assigned a new contact sequence number. To be used later.	smallint NOT NULL Primary Key
nodeID	Numeric identifier for the node.	smallint NOT NULL
contactType	Type of contact or leg: 1—Incoming. Outside contact received by the Unified CCX system.	smallint NOT NULL
mediaType	Type of the media such as email and other media types. 1—Chat 3—Email	smallint NOT NULL
contactDisposition	Disposition of the contact or the leg. 1—Abandoned 2—Handled 3—Do not care 4—Aborted [1] 5—Rejected 6—Cleared 7—Unknown	smallint NOT NULL

Table 1-42 TextContactDetail Table Fields (continued)

Field Name	Description	Storage
dispositionReason	Reason why the contact is aborted or rejected by the system. • Unknown • Chat_agent_ended • Chat_customer_ended • Chat_agent_aborted • Chat_agent_abandoned • Chat_customer_abandoned • Chat_abandoned_timeout • Chat_customer_abandoned • Chat _customer_waited • Chat_system_failure • Chat_system_failure_before_agent_joined • Chat_agent_connection_failure • Chat_agent_end_before_in _chatroom	varchar(100) NULL
originatorType	Originator of the contact or the leg: 1—Agent. Contact originated by an agent. 2—Unknown. Contact originated from outside.	smallint NOT NULL
originator	Numeric identifier of the agent who originated the contact or the leg. Used only if originatorType is 1.	nvarchar(50) int NULL
destinationType	Destination of the contact or the leg: 1—Agent. Contact presented to an agent. Null if no destination.	smallint NULL
destination	Numeric identifier of the agent who received the contact or the leg. Used only if destinationType is 1.	nvarchar(50) int NULL
startDateTime	Date and the time that the contact or the leg is presented to the agent.	datetime year to fraction (3) NOT NULL
endDateTime	Date and time that the contact or the leg is transferred or disconnected.	datetime year to fraction (3) NOT NULL
tagID	The string with which the contact or the leg is tagged.	nvarchar(50) NOT NULL

[1] For aborted calls, the corresponding value in ContactQueueDetail.disposition = 1 (abandoned).

Related Tables

- TextAgentConnectionDetail, page 65 (via ContactID, ContactSeqNum, and nodeID)
- TextContactQueueDetail, page 70 (via ContactID, ContactSeqNum, nodeID, and qIndex)

TextContactQueueDetail

Database table name: TextContactQueueDetail

The Unified CCX system writes the record when the contact is queued for chat and email CSQs; then one of the following happens:

- Contact or leg is abandoned while queued for chat and email CSQs
- Contact or leg is being dequeued
- Contact or leg is connected to an agent

The TextContactQueueDetail table contains the fields shown in Table 1-43.

Table 1-43 TextContactQueueDetail Table Fields

Field Name	Description	Storage
ContactID	Alphanumeric identifier for the contact.	varchar (64) NOT NULL Primary Key
ContactSeqNum	Contact sequence number that the system assigned to the contact or the leg. Each leg of a call is assigned a new contact sequence number. To be used later.	smallint NOT NULL Primary Key
nodeID	Numeric identifier for the node.	smallint NOT NULL
csqRecordID	Numeric identifier for the chat and email CSQ.	int NOT NULL Primary Key
qIndex	A new qIndex is created whenever a Unified CCX contact is conferenced to a Unified CCX route point. To be used later.	smallint NOT NULL Primary Key
disposition	Disposition for this leg of the contact for this CSQ. • Abandoned = 1[1] • Handled by CSQ = 2 • Dequeued from CSQ = 3 • Handled by another CSQ = 4	smallint NULL

Table 1-43 TextContactQueueDetail Table Fields (continued)

Field Name	Description	Storage
metServiceLevel	Contact answered within the configured number of seconds of queue time for this CSQ. • Yes = t • No = f To be used later.	boolean NULL
queueTime	Number of seconds the contact spent in queue for this CSQ and this leg of the contact.	int NULL

[1] For aborted calls, ContactQueueDetail.disposition = 1 and the corresponding ContactCallDetail.contactDisposition = 4 (aborted).

Related Tables

- TextAgentConnectionDetail, page 65 (via ContactID, ContactSeqNum, nodeID, and qIndex)
- TextContactDetail, page 68 (via ContactID, ContactSeqNum, and nodeID)
- TextCustomer Detail, page 71 (Via ContactID)
- ContactServiceQueue, page 28 (via csqRecordID)

TextCustomer Detail

Database table name: TextCustomerDetail

The Unified CCX system creates a new record in the TextCustomerDetails table when a chat agent receives the contact.

The TextCustomerDetail table captures customer related information corresponding to the chat contact. Maximum 10 customer fields can be persisted in the table. Each field value can be of maximum 40 characters. The chat customer is advised to limit each form field value to 40 characters so that the data truncation will not happen while storing the customer data into the database. Customer can write custom reports on top of this historical reporting table and use the persisted data.

The TextCustomer Detail table contains the fields shown in Table 1-44.

Table 1-44 TextCustomerDetail Table Fields

Field Name	Description	Storage
ContactID	Alphanumeric identifier for this record.	nvarchar (64) NOT NULL Primary Key
FieldID1 to FieldID10	The unique field IDs corresponding to the field names in the chat user form or MediaCustomerDataMapping table. In actual table schema there are 10 individual columns named FieldID1 through FieldID10.	int NOT NULL

Table 1-44 TextCustomerDetail Table Fields (continued)

Field Name	Description	Storage
FieldValue1 to FieldValue10	Indicates the field values provided for the corresponding field names in the chat user form or MediaCustomerDataMapping table. In actual table schema there are 10 individual columns named FieldValue1 through FieldValue10.	lvarchar (600) NOT NULL
InsertionDate	Indicates the date and time of insertion.	datetime year to fraction(3) NOT NULL

Related Tables

- ChatUserForm, page 18 (via fieldID)
- TextContactDetail, page 68 (via ContactID)
- MediaCustomerDataMapping, page 46 (via FieldID)

WorkflowTask

Database table name: WorkflowTask

A WorkflowTask record contains information about a task or a subtask that runs on the Unified CCX system.

The Workflow Task table contains the fields shown in Table 1-45.

Table 1-45 WorkflowTask Table Fields

Field Name	Description	Storage
taskID	Identifier of the task.	decimal(18, 0) NOT NULL Primary Key
parentTaskID	Identifier of the parent task, if the task is a subtask.	decimal(18, 0) NULL
startDateTime	Date and the time that the task started executing.	datetime year to second NOT NULL
endDateTime	Date and the time that the task completed executing.	datetime year to second NOT NULL
applicationServerID	Unique identifier assigned to each Unified CCX server in the cluster.	smallint NOT NULL Primary Key

Related Tables

- AgentConnectionDetail, page 5 (applicationServerID maps to nodeID)
- ContactCallDetail, page 19 (applicationServerID maps to nodeID)
- ContactQueueDetail, page 25 (applicationServerID maps to nodeID)
- ContactRoutingDetail, page 27 (applicationServerID maps to nodeID)
- MonitoredResourceDetail, page 44 (applicationServerID maps to nodeID)
- PurgeHistory, page 47 (applicationServerID maps to nodeID)

EEMActiveEmail

When an email is being processed by an agent, a record for that email is created in the EEMActiveEmail table. The columns in the table are updated as the agent works on the email. When the email is terminated (sent, deleted, requeued), the record is deleted.

The EEMActiveEmail table contains one row for each email message being processed by an agent

Database table name: EEMActiveEmail

Table 1-46 EEMActiveEmail Table Fields

Field Name	Description	Storage
emailID	Unique record identifier	SERIAL NOT NULL Primary key
emailUIDOnMailServer	Identifier for this email message	nvarchar(20) NULL
fromAddress	Email address of sender	lvarchar(320) NULL
toAddress	Email address(es) of recipient(s)	lvarchar(320) NULL
emailSubject	Subject line	Nvarchar(200) NULL
receivedDateTime	Date the email was received by the mail server, in local time.	datetime year to second NULL
receivedDateTimeGmt	Date the email was received by the mail server (GMT).	datetime year to second NULL
EEMQueueId	Id of email queue email was routed to	Int NULL
queuedDateTime	Date the email was placed into the CSQ, in local time.	datetime year to second NULL

Table 1-46 EEMActiveEmail Table Fields (continued)

Field Name	Description	Storage
queuedDateTimeGmt	Date the email was placed into the CSQ (GMT).	datetime year to second NULL
draftUIDOnMailServer	If message is in draft, the UID of the draft. Currently always blank; reserved for future use	nvarchar(20) NULL
emailStatusFK	The email status	Int NULL Foreign Key
agent	The agent login name, if an agent is working on this email, otherwise null.	nvarchar(64) NULL
getDateTime	Date/Time email was assigned to an agent, in local time.	datetime year to second NULL
getDateTimeGmt	Date/Time email was assigned to an agent (GMT).	datetime year to second NULL
inProcessTimeSeconds	Number of seconds, cumulative, that one or more agents has spent in the processing state for this email.	Int NULL

Related Tables

- EEMEmailStatusDescription, page 77 (via emailStatusFK)
- EEMContactEmailDetail, page 74 (via emailUIDOnMailServer)

EEMContactEmailDetail

When an email message is placed into a CSQ, a record for that email message is created in this table. When the email is assigned to an agent, the agent name column is updated. When the email is terminated (sent, deleted, requeued), the finalDisposition columns are updated.

The EEMContactEmailDetail table contains one row for each email message currently in the system.

Database table name: EEMContactEmailDetail

Table 1-47 EEMContactEmailDetail Table Fields

Field Name	Description	Storage
ContactEmailDetailId	Unique record identifier	SERIAL NOT NULL Primary key
emailUIDOnMailServer	Identifier for this email message	nvarchar(20) NULL

Table 1-47 EEMContactEmailDetail Table Fields (continued)

Field Name	Description	Storage
fromAddress	Email address of sender	lvarchar(320) NULL
toAddress	Email address(es) of recipient(s)	lvarchar(320) NULL
emailSubject	Subject line	nvarchar(200) NULL
EEMQueueId	Id of email queue email was routed to	Int NULL
receivedDateTime	Date the email was received by the mail server	datetime year to second NULL
receivedDateTimeGmt	Date the email was received by the mail server (GMT)	datetime year to second NULL
receivedReasonFK	Index into EEMReasonCodeDescription table, the reason code associated with this message	int NULL
agent	The agent login name, if an agent is working on this email, otherwise null.	nvarchar(64) NULL
finalDispositionTypeFK	Index into EEMEmailStatusDescription	int NULL
finalDispositionDateTime	Date the email was disposed, in local time, otherwise null.	datetime year to second NULL
finalDispositionDateTime GMT	Date the email was disposed (GMT), otherwise null.	datetime year to second NULL
PeerReviewQueueId	Is a review CSQ, the email routed for peer review.	int NULL
PeerReviewUID	New email routed into review queue and gets a new UID. This new UID is stored in PeerReviewUID	nvarchar(20) NULL

Related Tables

- EEMReasonCodeDescription, page 79 (via receivedReasonFK)
- EEMEmailStatusDescription, page 77 (via finalDispositionTypeFK)

EEMEmailAgentStateDetail

When an agent makes an email-related state change, the state change is recorded in the EEMEmailAgentStateDetail table. For each state change, there is an "eventStart" and an "eventEnd". The times for these are recorded in the corresponding eventStartDateTime and eventEndDateTime columns. Human-readable descriptions of states and reason codes are found in the EEMReasonCodeDescription and EEMStateDescription tables. Some events (login and logout) have a zero length duration, as the agent immediately moves to another state.

The EEMEmailAgentStateDetail table contains one row for each Email state change for an agent.

Database table name: EEMEmailAgentStateDetail

Table 1-48 EEMEmailAgentStateDetail Table Fields

Field Name	Description	Storage
EEMEmailAgentStateDetailId	Unique record identifier	SERIAL NOT NULL Primary key
emailUIDOnMailServer	Identifier for this email message	nvarchar(20) NULL
agent	The agent login name, if an agent is working on this email, otherwise null.	nvarchar(64) NULL
eventStartDateTime	Date/Time that the state started on, in local time	datetime year to second NULL
eventStartDateTimeGmt	Date/Time that the state started on (GMT).	datetime year to second NULL
eventEndDateTime	Date/Time that the state ended on, in local time. Null if the state has not ended.	datetime year to second NULL
eventEndDateTimeGmt	Date/Time that the state started on (GMT). Null if the state has not ended.	datetime year to second NULL
stateFK	ID of the state.	Int NULL Foreign Key
reasonCodeFK	Reason code ID, if applicable. Null otherwise.	Int NULL
EEMQueueId	The ID of the CSQ the agent was in when this state transition occurred, if applicable, otherwise null.	Int NULL

Table 1-48 EEMEmailAgentStateDetail Table Fields (continued)

Field Name	Description	Storage
emailStatusFK	The email status	Int NULL Foreign Key

Related Tables

- EEMContactEmailDetail, page 74 (via emailUIDOnMailServer)
- EEMReasonCodeDescription, page 79 (via reasonCodeFK)
- EEMStateDescription, page 79 (via stateFK)

EEMEmailStatusDescription

The EEMEmailStatusDescription table contains descriptive text (in English only) for each possible email status.

Database table name: EEMEmailStatusDescription

Table 1-49 EEMEmailStatusDescription Table Fields

Field Name	Description	Storage
emailStatusId	Unique record identifier	INT NOT NULL Primary key
emailStatusDescription	Text describing the status	nvarchar(20) NULL

EEMQueueAgentDetail

When an email is assigned to an agent, a record is created in the EEMQueueAgentDetail table. It associates the agent and the email message that the agent is working on. The agent can enter a processing state, and later exit the processing state. The elapsed time the agent is in the processing state, is accumulated in the inProcessTimeSeconds column. When the email is terminated (sent, deleted, requeued), the "endDateTime", endTypeFK, and wrapupData columns are updated.

The human-readable descriptions of the receivedReasonFK and endTypeFK columns are found in EEMReasonCodeDescription and EEMEmailStatusDescription tables.

The EEMQueueAgentDetail table associates an agent and an email message and contains one record for each email message an agent is working on.

Database table name: EEMQueueAgentDetail

Table 1-50 EEMQueueAgentDetail Table Fields

Field Name	Description	Storage
queueAgentDetailSequencingId	Unique record identifier.	SERIAL NOT NULL Primary key
ContactEmailDetailFK	ID of a record in EEMContactEmailDetail; refers to the email message belonging to this record.	INT NULL Foreign key
EEMQueueId	Id of email queue email was routed to.	Int NULL
Agent	The agent login name, if an agent is working on this email, otherwise null.	nvarchar(64) NULL
queueDateTime	Date/Time email was placed in the queue, in local time.	datetime year to second NULL
queueDateTimeGmt	Date/Time email was assigned to an agent (GMT).	datetime year to second NULL
getEmailDateTime	Date/Time email was assigned to an agent, in local time.	datetime year to second NULL
getEmailDateTimeGmt	Date/Time email was assigned to an agent (GMT).	datetime year to second NULL
endEmailDateTime	Date/Time email was requeued, sent, or deleted by the agent, in local time. Null if the email was not requeued, sent, or deleted.	datetime year to second NULL
endEmailDateTimeGmt	Date/Time email was requeued, sent, or deleted by the agent (GMT). If email was not requeued, sent, or deleted, then NULL.	datetime year to second NULL
receivedReasonFK	Reason Code ID for why the email was placed in the CSQ.	Int NULL
endTypeFK	Reason Code ID for why the email was terminated by the agent. Null if not terminated.	Int NULL Foreign key
inProcessTimeSeconds	Total time in seconds the agent spent processing this email message.	Int NULL

Table 1-50 EEMQueueAgentDetail Table Fields (continued)

Field Name	Description	Storage
wrapupData	Wrapup selected by agent when email is terminated.	nvarchar 40 NULL

Related Tables

- EEMContactEmailDetail, page 74 (via ContactEmailDetailFK)
- EEMReasonCodeDescription, page 79 (via receivedReasonFK)
- EEMEmailStatusDescription, page 77 (via endTypeFK)

EEMReasonCodeDescription

The EEMReasonCodeDescription table contains descriptive text (in English only) for each possible reason code.

Database table name: EEMReasonCodeDescription

Table 1-51 EEMReasonCodeDescription Table Fields

Field Name	Description	Storage
reasonCodeId	Unique record identifier	INT NOT NULL Primary key
reasonCodeDescription	Text describing the reason code	nvarchar(20) NULL

EEMStateDescription

The EEMStateDescription table contains descriptive text (in English only) for each e-mail state.

Database table name: EEMStateDescription

Table 1-52 EEMStateDescriptionTable Fields

Field Name	Description	Storage
stateId	Unique record identifier	INT NOT NULL Primary key
stateDescription	Text describing the state	nvarchar(20) NULL

EEMTablesVersion

The EEMTablesVersion table contains a single row indicating the version of the EEM database tables. This is used when upgrading from a previous version.

Database table name: EEMTablesVersion

Table 1-53 EEMTablesVersion Table Fields

Field Name	Description	Storage
version	The current database version number	INT NOT NULL
updateDate	Date/time of install or upgrade	datetime year to second NOT NULL

in ContactServiceQueue table 30