Introduction

- The Unified ICM and Unified CC Enterprise Databases, on page 1
- General Concepts, on page 2
- Real-Time and Historical Data, on page 6
- Historical data replication, on page 7

The Unified ICM and Unified CC Enterprise Databases

Unified ICM and Unified CC Enterprise software uses the following types of databases:

- The central database that is part of the Central Controller.
- The local database on each distributor Administration & Data Server.
- The Historical Data Server (HDS) database on a distributor Administration & Data Server.

The following table lists the databases and respective customer instance names.

Database	Database Name
Central	<customer>_sideA or <customer>_sideB
Local	<customer>_awdb
Historical Data Server	<customer>_hds

Unified ICM and Unified CC Enterprise software uses information in the central database to determine how to route each call. This includes information about your telephone system configuration and routing scripts. The local database contains copy of the configuration data and scripts from the central database.

The local database also contains tables of real-time information that describe activity at the call centers. (The Central Controller keeps the real-time information in memory but does not store it in the central database.) This information allows you to monitor current activity within the system.

The central database stores historical information describing past activities at the call centers and within Unified ICM and Unified CC Enterprise systems. A special HDS database on a distributor Administration & Data Server at each site also stores the same historical information. Therefore, either the central database or an HDS database is the historical database for an Administration & Data Server user. You can access historical information that is stored in the historical database to produce reports and screens.

General Concepts

This section gives a brief overview of relational database concepts and details about how data is generated by the system software.

Tables Columns and Rows

A database contains tables of data. A table defines a series of columns or fields. The actual data is stored as rows or records within each table. Each row contains one value for each column of the table. For example, Figure 1 shows a table with five columns. It contains three rows of data.

Announcement Table

NetworkTargetID	AnnouncementType	EnterpriseName	Description	DbFlags
1	0	ann503	Bad data	0
2	0	ann504	Delays	0
3	0	ann505	After hours	1

The data in tables changes for each system, but the definition of tables and columns does not. This manual describes the columns of each table; it does not describe the actual data in table rows.

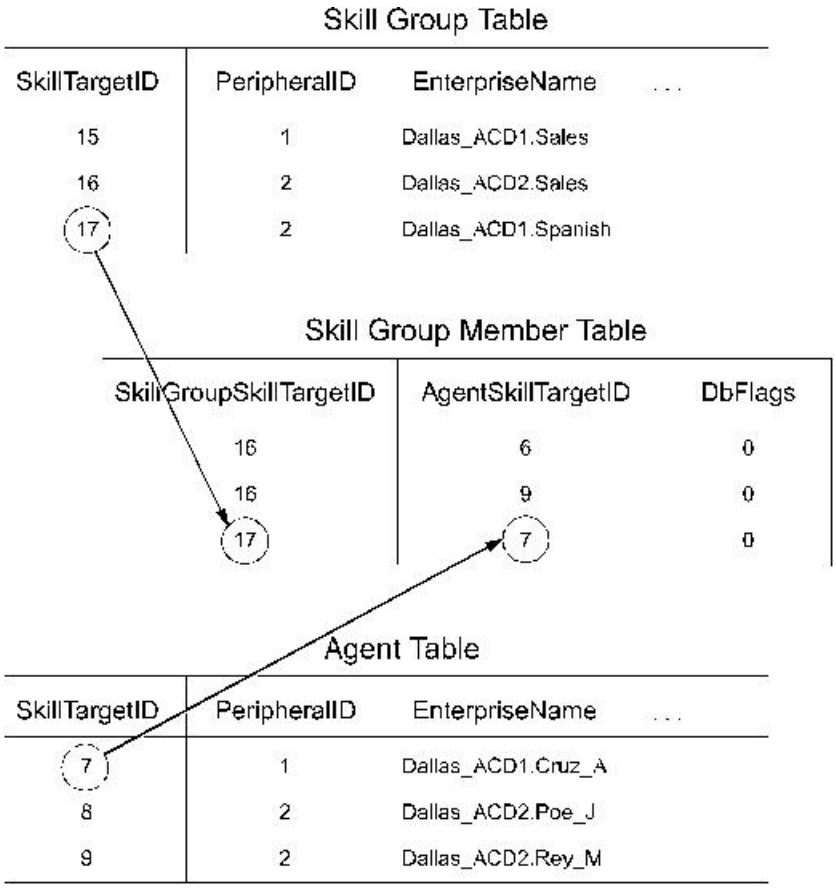
Table Relationships

Related tables in a database share one or more common fields or columns. For example, both the Agent and Peripheral tables include the PeripheralID field. This defines a relationship: each row in the Agent table is related to the row in the Peripheral table that shares the same PeripheralID value.

Relationships between tables can be one-to-one or one-to-many. For example, because one peripheral can be associated with many agents, the relationship between the Peripheral and Agent tables is one-to-many. On the other hand, each peripheral has a single peripheral default route and each peripheral default route belongs to only one peripheral. Therefore, the relationship between the Peripheral and Peripheral Default Route tables is one-to-one.

Sometimes a single row might not be associated with any rows in a related table. For example, it is possible to define a peripheral with no associated agents. Usually, this would only be a temporary condition. In some cases, however, the condition might be permanent. For example, you can define a trunk group but not define the associated trunks.

Sometimes the natural relationship between two tables appears to be many-to-many. For example, each agent can be a member of many skill groups and each skill group can contain many agents. Therefore, the Agent and Skill Group tables appear to have a many-to-many relationship. However, in this case, a third table, called a cross-reference table, actually links the tables so the relationship is actually one-to-many. For example, Figure 2 shows how the Skill Group Member table acts as a cross-reference table for the Agent and Skill Group tables.



The Skill Group Member table contains one record for each member of each skill group. It has one-to-many relationships with both the Agent table and the Skill Group table. This avoids a direct many-to-many relationship between the Agent and Skill Group tables.

Key Fields

One or more fields within a table can form a key. Keys are the fields that you commonly use to locate specific records. Usually the fields that make up a key are defined as NOT NULL (meaning they cannot take the NULL value), but there are many exceptions.

Most tables have a primary key. For example, the PeripheralID field is the primary key for the Peripheral table.

An example of a foreign key is the PeripheralID field in the Agent table. You can use this key to find all agents that are associated with a specific peripheral.

The Agent table contains two alternate keys: the EnterpriseName field, and the combination of the PeripheralID and PeripheralNumber fields. A value for either of these keys uniquely identifies an agent.

The combination of FirstName and LastName is an inversion key for the Agent table. While this key value is not necessarily unique, it is a convenient way to locate specific agents. This table lists the types of keys and the codes that are used for them in the system database.

Key Type	Code	Description
Primary key	PK	Consists of one or more fields that have a unique value for each record in the table. By default, the primary key is the clustered key for the table.
Alternate key	AK	A unique key, other than the primary key, that you can use to locate a specific record.
Foreign key	FK	A primary key from one table that appears in a second table. A foreign key that establishes a one-to-one relationship is always unique. A foreign key that establishes a one-to-many relationship is not unique.
Inversion key	IE	A key that does not necessarily have a unique value, but can be used to locate a group of records within the table.

Note By default, all keys are on the PRIMARY file group for the database. Microsoft SQL Server always creates the PRIMARY file group as the default file group.

The codes from this table are used to identify key fields in each table. If a table has more than one key of the same type, then numbers are attached to the codes. For example, if a table has two alternate keys, then the fields in the first are "AK1" and the fields in the second are "AK2."

Each field is marked as either NULL (meaning that NULL is a valid value) or NOT NULL (meaning that NULL is not valid).

Reserved Fields

Some fields in the database are marked as reserved. This means that system software or the database manager might use the field, but it has no external meaning. You must not modify any field marked as reserved.

Field Applicability

Unless specifically indicated otherwise, table fields apply to both Unified ICM and Unified CCE.

Data Types

This table describes the data types used for fields in the Unified ICM and Unified CCE database.

Unified ICM and Unified CCE Defined Data Type	MS SQL Server Data Type	Null Option Default	Description
CHANGESTAMP	int	NOT NULL	Consists of one or more fields that have a unique value for each record in the table.
DBCHAR	char(1)	NOT NULL	Up to 1 character. The value 1 is the storage size.

Unified ICM and Unified CCE Defined Data Type	MS SQL Server Data Type	Null Option Default	Description
DBDATETIME	datetime	datetime	A date and time accurate to the second. Stored as two four-byte integers (eight bytes total): days before or since January 1, 1900 and seconds since midnight.
DBFLT4	real	NULL	A four-byte floating-point value (7-digit precision).
DBFLT8	float	float	An eight-byte floating-point value (15-digit precision).
DBSMALLDATE	smalldatetime	smalldatetime	A date and time accurate to the minute. Stored as two unsigned two-byte integers (four bytes total): number of days since January 1, 1900 and minutes since midnight.
DBINT	int	NULL	A four-byte integer value between -2,147,483,648 and 2,147,483,647.
DBSMALLINT	smallint	NULL	A two-byte integer value between -32,768 and 32,767.
DESCRIPTION	varchar(255)	NULL	Up to 255 characters. The value 255 is the storage size.
DBTINYINT	tinyint	NOT NULL	A one-byte integer value between 0 and 255.
TELNO	char (10)	NULL	Up to 10 characters. The value 10 is the storage size.
VNAME32	varchar(32)	varchar(32)	Up to 32 characters. The value 32 is the storage size.
VTELNO10	varchar(10)	NULL	Up to 10 characters. The value 10 is the storage size.
VTELNO20	varchar(20)	NULL	Up to 20 characters. The value 20 is the storage size.
char(n)	char(n)	NULL	Up to n characters. The value n is the storage size.
varchar(n)	varchar(n)	NULL	Up to n characters. The value n is the storage size.
image	image	NULL	Up to 2,147,483,647 bytes of binary data. The storage size is determined by the length of the data.
datetime	datetime	NULL	A date and time accurate to the second. Stored as two four-byte integers (eight bytes total): days before or since January 1, 1900 and seconds since midnight.
smalldatetime	smalldatetime	NULL	A date and time accurate to the minute. Stored as two unsigned two-byte integers (four bytes total): number of days since January 1, 1900 and minutes since midnight.

Real-Time and Historical Data

Unified ICM and Unified CCE software maintains real-time and historical status information about certain objects in the system such as service, skill groups, routes, and scripts.

For example, the Route Real Time table contains real-time information about each route. The Route Five Minute and Route Half Hour tables contain historical information about each route. The Route Real Time table contains one row for each route. (It has a one-to-one relationship with the Route table.) The Route Half Hour table contains many rows for each route--Unified ICM and Unified CCE software adds an additional row for each route every half hour. (It has a one-to-many relationship with the Route table.)

Note The Half Hour database tables available in the database are not populated because these tables are not supported. These tables are replaced by the Interval database tables.

The system software updates the real-time tables in the database every ten seconds. Real-time information includes information about what is happening right now (for example, CallsQNow and ExpectedDelay). It also includes summary information about what has happened during the last five minutes (for example, CallsIncomingTo5 and AvgTalkTimeTo5), since the last half-hour historical data (for example, CallsRoutedHalf and CallsAbandQHalf), and since midnight (for example, CallsOfferedToday and CallsHandledToday).

Unified ICM and Unified CCE software generates historical information on five- and 30-minute intervals, with the first interval beginning at midnight. For example, Unified ICM and Unified CCE software adds a new row for each Route to the Route Five Minute table every five minutes. Unified ICM and Unified CCE software adds a new row for each Route to the Route Half Hour table every 30 minutes. Some of the information for the historical tables is derived from accumulation fields in the real-time tables. For example, at the end of each five-minute interval, the value from the CallsOfferedTo5 field in the Route Real Time table is copied to the CallsOfferedTo5 field of the Route Five Minute table.

Each five- and 30-minute row contains a field for the date-time. The time stored in this field is the time at the start of the interval. For example, a Service Five Minute row for the interval from 10:00 a.m. to 10:05 a.m. contains the time 10:00 a.m. However, some fields within the table contain a snapshot of data from the end of the interval. For example, the CallsQNow field of the Service Five Minute table contains the number of calls queued at the end of the five-minute period. Therefore, the Service Five Minute row with the time of 10:00 a.m. tells you the number of calls queued at 10:05 a.m. To find the number of calls queued at 10:00 a.m., look at the Service Five Minute record for 9:55 a.m.

Call Detail Data

Each time Unified ICM and Unified CCE software processes a routing request, it generates a Route Call Detail row that contains information about the request and routing decision it made. Each row includes the day on which the request was handled and a key value generated by Unified ICM and Unified CCE software that is unique among all requests handled that day. These two values together comprise a unique identifier for the call.

When Unified ICM and Unified CCE software receives information that a call is completely done (that is, for example, it has been routed to a peripheral, handled by an agent, and disconnected), then a row about the call is written to the Termination Call Detail table. The Termination Call Detail row indicates the agent, skill group, and service that handled the call. It also contains information such as how long the caller was on hold, and whether the call was transferred to another agent after the initial routing.

If the call was sent to a translation route, the Termination Call Detail row contains the same day and router key values as the Route Call Detail row for the same call. You can use these fields to link the tables and find all the call detail information for a single call. This process is called cradle-to-grave call tracking.

Historical data replication

All tables are replicated in AW-HDS-DDS and HDS-DDS modes.

For AW-HDS, all tables except the following gets replicated:

- Route_Call_Detail
- Route_Call_Variable
- Termination_Call_Detail
- Termination_Call_Variable