Database Schema

About Database Schema

The Cisco Unified Customer Voice Portal (United CVP) reporting server hosts an IBM Informix Dynamic Server (IDS) database, which stores reporting data in a defined database schema. Customers who choose to deploy Cisco Unified Intelligence Center (Unified Intelligence Center) as their reporting platform must add the Informix database as a data source in Unified Intelligence Center.

The schema is fully published so that customers can develop custom reports. Customers may not, however, extend the schema for their own purposes.

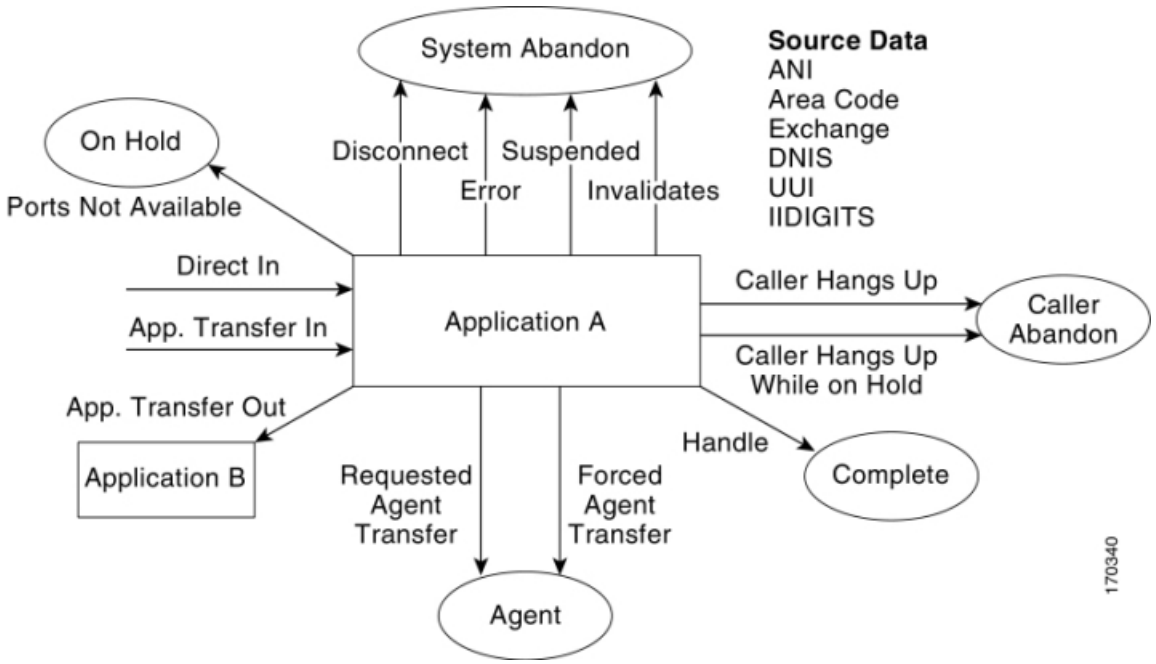
The schema provides Unified CVP customers with the ability to:

- Establish database connectivity with Unified Intelligence Center and to import and run the Unified CVP templates with Unified Intelligence Center.
- Establish database connectivity with other commercial off-the-shelf reporting and analytics engines and then build custom reports against the Unified CVP database schema.

Note Your support provider cannot assist you with custom reports or with commercial (non-Cisco) reporting products.

The following diagram indicates a common set of incoming and outgoing entry and exit states for a call to a self-service application.

Figure 1: Call Flow



Note When basic video is transferred to an audio-only agent, the call remains classified as basic video accepted.

Data Model Diagram

The following entity-relationship diagrams depict the Unified CVP database schema.

Figure 2: Call Tables

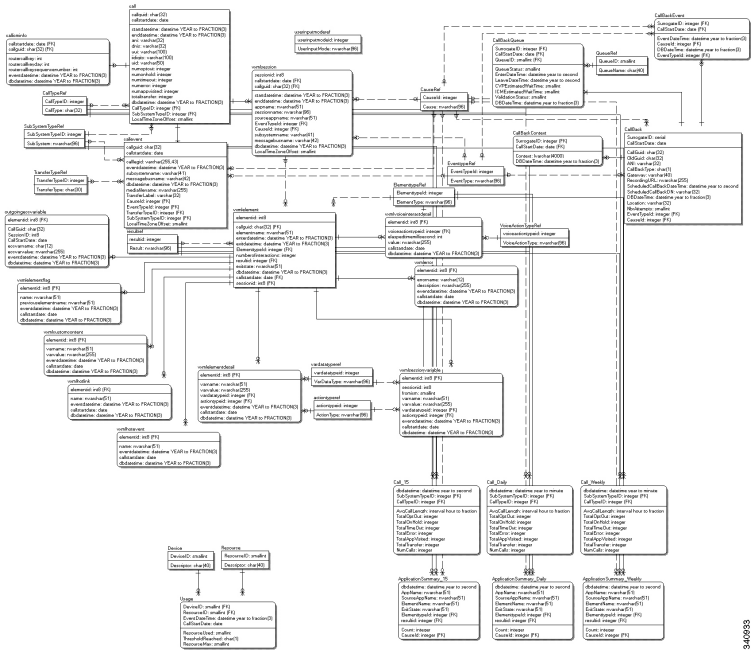


Figure 3: Callback Tables

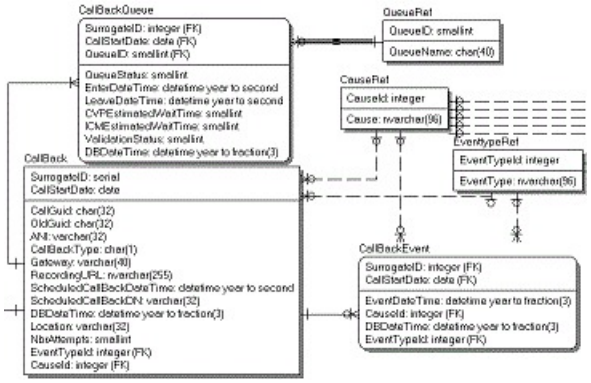


Figure 4: Summary Tables

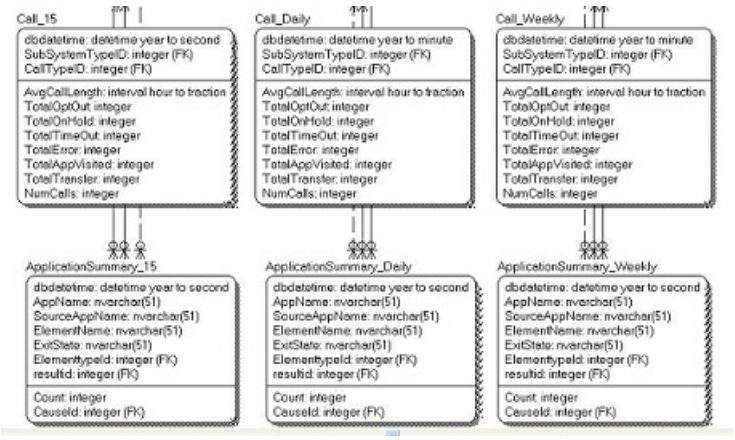
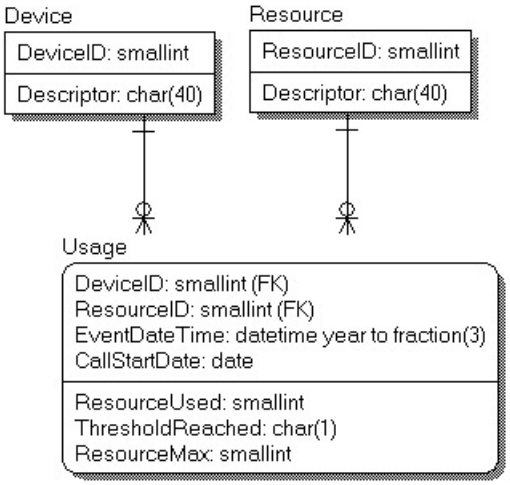


Figure 5: Trunk Group Utilization Tables



Keys

The documentation for the reporting schema lists fields as PK, FK, A, or No.

Fields are designated in this document as Primary Key (PK), Foreign Key (FK), or Alternate Key (AK) for informational purposes only. For performance reasons, the schema does not enforce Primary, Foreign, or Alternate keys. When the Index column for a field shows FK or AK, it means that a field can be looked up to return a text explanation of the code in another table.

Primary and Alternate Keys are in fact supported by an index. Major Foreign Keys (CallGUID, SessionID, ElementID) have a supporting index. Foreign Keys which refer to lookup tables are not supported by an index.

Unified Customer Voice Portal Reporting Data Model

The following section provides information on the following topics:

DateTime Columns

Most major tables have three columns to assist in managing the reporting server itself.

- **CallStartDate** - This column is used for partitioning and purging data.

  This is the date and time the call started and is meant to ensure that detail data, which may cross a date boundary, are tied to the original call and can all be stored and removed together.

- **EventDateTime** - This is the date and time that the recorded event transpired. This is recorded in UTC time.

  The Call Table table has two EventDateTime fields, recorded as *StartDateTime* and *EndDateTime*.

- **DBDateTime** - This is the date and time that the recorded event was written to the database. It is meant to contrast with the EventDateTime. A marked difference between these values indicates a delay in the data arriving at the reporting server. This delay should either be allowed for or investigated.

Informix Dates and Times

The Informix engine that hosts the Unified CVP reporting database supports three concepts of time:

- Dates
- DateTimes
- Intervals

Dates

A date (for example CallStartDate) has no time element to it. It is specified between single quotes as 'MM/DD/YYYY'.

```
SELECT count(*)
  FROM Call
 WHERE CallStartDate='05/31/2012';
```

This date format can be modified to suit the locale with the DBDATE environment variable: in this case DBDATE=MDY4/, or Month/Day/Year(4) with a forward slash separator. These can be arranged in any order (DMY4-, or DMY2/ or Y4MD/) by modifying the DBDate enumeration variable.

Date also supports: key words such as 'TODAY' and date arithmetic.

For example, this returns a count of calls received yesterday:

```
SELECT count(*)
  FROM Call
 WHERE CallStartDate=TODAY-1;
```

Functions such as YEAR(), MONTH() and WEEKDAY().

```
SELECT count(*)
  From Call
 WHERE WEEKDAY(CallStartDate)=1
```

Note Days of the week are numbered from 0 through 6 where 0 is Sunday and 6 is Saturday.

DateTimes

DateTimes include a time component and use the ANSI standard: 'YYYY-MM-DD HH:MM:SS.FFF' where FFF are fractions of seconds. For example, this returns a count of calls received in a given 48 hours:

```
SELECT count(*)
  FROM Call
 WHERE Call.StartDateTime between '2009-05-01 00:00:00' AND '2009-05-3
       23:59:59';
```

These support the same YEAR(), MONTH() and WEEKDAY() functions as the Date datatype. The Current date and time is specified as 'CURRENT YEAR TO SECOND' and also supports date arithmetic.

```
SELECT count(*)
  FROM Call
 WHERE Call.StartDateTime > CURRENT YEAR TO SECOND - 2 UNITS DAY;
```

Unified CVP DateTimes are all recorded as UTC time, with the exception of *dbdatetime* which is recorded as a local time. Localtimezoneoffset is a column in the Call table that contains the number of minutes offset from UTC to derive the Local Time. This can be used as an interval. (In the example below, localtimezoneoffset is -240 minutes).

```
select first 10 enddatetime, enddatetime + localtimezoneoffset units
minute as LocalTime from call;

enddatetime             localtime

2010-02-09 15:03:54.453 2010-02-09 11:03:54.453
2010-02-09 15:03:54.453 2010-02-09 11:03:54.453
2010-02-09 15:03:54.469 2010-02-09 11:03:54.469
2010-02-09 15:01:23.125 2010-02-09 11:01:23.125
2010-02-09 15:03:54.469 2010-02-09 11:03:54.469
2010-02-09 15:01:23.141 2010-02-09 11:01:23.141
2010-02-09 15:03:54.500 2010-02-09 11:03:54.500
2010-02-09 15:01:23.156 2010-02-09 11:01:23.156
2010-02-09 15:01:23.156 2010-02-09 11:01:23.156
```

```
2010-02-09 15:01:23.156 2010-02-09 11:01:23.156
```

An aggregation function lastperiod(datetime, *Period*) is supported. *Period* can be: 15, 30, 60, DD, WW, or MM. This will convert the datetime into the date and time at which the current period started. Hence:

Lastperiod(2009-10-14 12:46:56,*15*) returns *2009-10-14 12:45:00*

Lastperiod(2009-10-14 12:46:56, *30*) returns *2009-10-14 12:30:00*

Lastperiod(2009-10-14 12:46:56, *60*) returns *2009-10-14 12:00:00*

Lastperiod(2009-10-14 12:46:56, *DD*) returns *2009-10-14 00:00:00*

Lastperiod(2009-10-14 12:46:56, *WW*) returns *2009-10-11 00:00:00 (Sunday)*

Lastperiod(2009-10-14 12:46:56, *MM*) returns *2009-10-1 00:00:00 (1st day of the month)*

Intervals

An Interval is a span of time and can be specified as *n UNITS period* where *period* can be:

- YEAR
- MONTH
- DAY
- HOUR
- MINUTE
- SECOND

A database query with an interval must be sent in the preceding format. When returned from the database, the interval will look like a datetime (YYYY-MM-DD HH:MM:SS.FFF). The components that are returned depend on the interval definition. It is unlikely that a DAY component will be returned from Unified CVP intervals; instead, expect a format like HH:MM:SS.FFF.

For a full discussion of Informix, refer to the Informix Guide to SQL: Reference Manual.

SIP Calls

SIP calls are recorded in the Call Table along with VXML calls.

They can be distinguished from VXML calls with the CallTypeID column. (Contains "4". Refer to the CallTypeRef Table, on page 33, where 4 is a SIP call.)

Events for these calls (such as start and end) are recorded in the CallEvent Table.

Sample Query and SIP Calls

Details for a SIP call could be retrieved using the following query:

```
SELECT Call.*, CallEvent.*
  FROM Call, CallEvent
 WHERE Call.CallGUID=CallEvent.CallGuid
   AND Call.CallGuid='CallGuid';
```

where CallGuid is replaced by the value of the CallGuid for which information is desired.

Trunk Utilization

Trunk utilization is a record of state messages from various devices linked to the reporting server and their current status. The frequency in which these messages are written is controlled by the IOS Gateway (Gateway Utilization). This data captures a point-in-time over time. It is laid out in a fact table (Usage Table) with three dimensions - Resource, Device, and Time.

Because time is not likely to be consistent across all devices, the Usage table has not been codified as an official dimension table, but rather as a date and time. Queries for usage should aggregate from this table.

Sample Queries, Trunk Utilization

Query for average CPU across all devices for the month of May:

```
SELECT avg(ResourceUsed)
  FROM Usage, Resource
 WHERE Resource.ResourceID=Usage.ResourceID
   AND Resource= 'CPU'
   AND Usage.EventDateTime between '2009-05-01 00:00:00' AND '2009-05-31
23:59:59';
```

Note that BETWEEN is inclusive. This query can also be written as:

```
AND Usage.EventDateTime >=  '2009-05-01 00:00:00' AND Usage.EventDateTime <=
  '2009-05-31
23:59:59';
```

Query for a list of devices and a count of the number of times they exceeded a threshold during the month of May:

```
SELECT Device, Resource, count(*)
  FROM Device, Resource, Usage
 WHERE Resource.ResourceID=Usage.ResourceID
   AND Device.DeviceID=Usage.DeviceID
   AND Usage.ThresholdReached= 'Y'
   AND month(Usage.EventDateTime) = 5
GROUP BY Device, Resource;
```

Note the use of the Month() function in *AND month (Usage.EventDateTime) = 5* .

Cisco Unified Customer Voice Portal Database Tables

This section lists the Unified CVP tables that hold reporting data.

Tables are categorized as follows:

Call Tables

The following Call tables are described in this section:

Call Table

This table is the primary record of a call and contains the basic metrics for each call. It contains one record per call.

Any drill into a specific call should start here to obtain the proper CallGUID.

On occasion, messages are dropped, even for an otherwise successful call. In such cases, EndDateTime is set to the same value as StartDateTime. Thus, if a call appears to be of 0 duration, report writers will know to exclude such a call from consideration in cases where it would otherwise skew metrics.

Table 1: Call Table

Field	Type	Null	Index	Description
CallGUID	char(32) for new installations char(35) for upgrades	No	PK (Composite CallGUID,CallStartDate)	The global unique id of a call.
CallStartDate	date	No	PK (Composite CallGUID,CallStartDate)	The date of the call, for data purging purposes.
StartDateTime	datetime YEAR to FRACTION(3)	No	Yes	EventDateTime for the date and time a call was made.
EndDateTime	datetime YEAR to FRACTION(3)	Yes	Yes	EventDateTime for the date and time a call ended with hang-up or disconnect.
ANI	varchar(32)	Yes	No	The ANI of the caller sent by telephony provider
DNIS	varchar(32)	No	No	The DNIS of a call sent by telephony provider.
UUI	varchar(100)	Yes	No	The UUI of the originating caller sent by telephony provider.

Field	Type	Null	Index	Description
IIdigits	varchar(100)	Yes	No	The IIDIGITS of the originating caller sent by telephony provider
UID	varchar(50)	Yes	No	The external UID of the caller if the call is associated with a user.
Numoptout (deprecated)	int	No	No	The number of times that the call is opted out to an agent.
NumTimeOut	int	No	No	The number of times the call timed out.
NumError	int	No	No	The number of errors that occurred during the call.
NumOnHold	int	No	No	The number on hold within a VXML application.
NumAppVisited	int	No	No	The number of applications visited during the life of the call.
TotalTransfer	int	No	No	The total number of times the call is transferred out. A transfer includes transfers to agents as well as a transfer to the VRU leg.
DBDateTime	datetime YEAR to FRACTION (3)	No	Yes	The date and time of the database operation (when the record was inserted).
CallTypeID	smallint *Formerly char(1)*	No	Non-Indexed FK	The type of call. See CallTypeRef Table.
SubSystemTypeID	int	No	Non-Indexed KX	The type of Unified CVP Service, such as SIP, IVR, VXML.
LocalTime ZoneOffset	smallint		No	The offset in minutes of the local timezone from UTC timezone. *Replaces LocalTimeZone.*

CallEvent Table

This table tracks each event that occurs within a call.

This table is populated for SIP calls. VXML calls will be recorded in the analogous VXMLSession Table.

Table 2: CallEvent Table

Field	Type	Null	Index	Description
CallGUID	char(32) for new installations char(35) for upgrades	No	Indexed FK (Composite CallGUID, CallStartDate)	The global unique id of a call
CallStartDate	date	No	FK (Composite CallGUID, CallStartDate)	The date of the call, for data purging purposes
CallLegID	varchar(255,43) *Formerly Varchar(43)*	Yes	No	A call id assigned by a Service
MessageBusName	varchar(42)	No	No	The name of the Call Server (its message adapter name) with which the event is associated
EventDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date and time of the event
EventTypeID	int	No	Non-Indexed FK	The mechanism used to generate the call event. See EventTypeRef Table.
CauseID	int	No	Non-Indexed FK	The reason that the call event was generated. See CauseRef Table.
DBDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date and time of the database operation (when the record was inserted).
TransferTypeID	integer	No	Non-Indexed FK	A unique id of the transfer type. See TransferTypeRef Table.
SubSystemTypeID	int	No	Non-Indexed FK	The type of the Service. See SubSystemTypeRef Table.
SubSystemName	varchar(41)	No	No	The name of the Service the event originated from
MediaFileName	nvarchar(255)	Yes	No	*This is always null.*

Field	Type	Null	Index	Description
TransferLabel	varchar(32)	Yes	No	This is the destination to which CVP transfers the call. The label is received from ICM via the TEMPORARY_CONNECT or CONNECT message
LocalTimeZoneOffset	smallint	NULL	No	The offset in minutes of this the local timezone from UTC timezone.

CallICMInfo Table

This table contains information to associate a Unified CVP call to ICM. It stores the ICM Call RouteCallKey, RouterCallKeyDay and RouterCallSequenceNumber for a call.

The CallICMInfo table is populated when the call is on the switch leg. This table is populated by SIP or VXML subsystems.

Note Currently the system does not capture the VRU leg of the call; thus if you have a Capture element and multiple Termination Call Detail (TCD) records are cut, the RouterCallKeySequenceNumber will increment in Historical Data Server (HDS) but will not be captured in the Unified CVP database. This is a known limitation.

Refer to the *Configuration Guide for Cisco Unified Customer Voice Portal* for further explanation about using the ReqICMLabel element to pass data to a Unified ICME script.

Table 3: CallICMInfo Table

Field	Type	Null	Index	Description
CallGUID	char(32) for new installations char(35) for upgrades	No	Indexed FK (Composite CallGUID, CallStartDate)	The global unique id of a call
CallStartDate	date	No	FK (Composite CallGUID, CallStartDate)	The date of the call, for data purging purposes

Field	Type	Null	Index	Description
RouterCallKey	Integer	No	AK (Composite index RouterCallKey, RouterCallKey Day)	ICM Router CallKey - single value per call. This value does not increment if the call is transferred from switch leg to VRU leg or if the call is transferred to an agent. If the call is a consult or conference, then Unified CVP will see two different callguids for the same call in its database. • The first callguid is the incoming callguid when the call is established. • The second callguid is for an agent originated/consult call. The RouterCallKey and RouterCallKey Day will act as a binder/glue between the two callguids for that single call as these values will not change between the two legs of the call.

Field	Type	Null	Index	Description
RouterCallKeyDay	Integer	No	AK(Composite index RouterCallKey, RouterCallKey Day)	ICM RouterCallKeyDay Typically this number changes on the switch and VRU leg of a call. You will see 0 for the switch leg of the call and 1 for the VRU leg of the call. This number usually does not change for basic CVP calls, but will increment if customers are using the capture node in their ICM script or when there is a transfer to an agent on the switch leg. In this scenario, Unified CVP sends a new call to Cisco Unified Communications Manager (Unified CM). This comes back via JTAPI and is on a separate Peripheral Gateway (PG). As the new call shows on a separate PG, Unified ICM cuts a new TCD record when the call ends. The RouterCallKeySequenceNumber increments on that switch leg.
RouterCallKey SequenceNumber	int	Yes	Yes	ICM RouterCallKeySequenceNumber.
EventDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date and time of the event.
DBDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date and time of the database operation (when the record was inserted).

VXML Tables

The following VXML tables are described in this section:

- VXMLElementFlag Table, on page 18
- VXMLError Table, on page 18
- VXMLHotEvent Table, on page 19
- VXMLHotLink Table, on page 20
- VXMLSession Table, on page 21
- VXMLSessionVariable Table, on page 22
- VXMLVoiceInteractDetail Table, on page 24

The data for VXML treatment is much richer than that which is available for SIP calls. Events can be captured from VXML for anything that occurs inside of the VXML script. These calls start at the Call Table, on page 9 and are linked to the VXMLSession Table, on page 21 using the CallGUID column.

The VXMLSession is made up of a series of elements that are visited within the context of an application. Each element may have multiple ancillary attributes such as flags that can be set in an element. Values for these flags may be found in the VXMLElementFlag Table, on page 18 and are linked to using the ElementID.

VXMLElementFlags information for a call can be retrieved using the following query:

```
SELECT VXMLElementFlag.Name
  FROM Call, VXMLSession, VXMLElement, VXMLElementFlag
 WHERE Call.CallGuid= CallGuid
AND Call.CallGuid=VXMLSession.CallGuid
AND VXMLSession.SessionID=VXMLElement.SessionID
AND VXMLElement.ElementID=VXMLElementFlag.ElementID;
```

where CallGuid is replaced by the value of the CallGuid for which information is desired.

VXMLCustomContent Table

This table contains one record for each VXML custom event. This event occurs if a custom component programmatically calls the AddToLog method of the Session API. The event will also occur when an element whose configuration contains entries in the Add To Log table in the General tab is run.

Table 4: VXMLCustomContent Table

Field	Type	Null	Index	Description
ElementID	int8	No	Indexed FK (Composite index ElementID, CallStartDate)	The unique ID of a visited element.
CallStartDate	date	No	FK (Composite index ElementID, CallStartDate)	The date of the call, for data purging purposes.
VarName	nvarchar(51)	No	No	The name of the custom event variable.

Field	Type	Null	Index	Description
VarValue	nvarchar(255)	Yes	No	The value of the custom event variable.
EventDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date and time when the variable is changed.
DBDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date and time of the database operation (when the record was inserted). This is useful for debugging purposes to determine lags between when the event occurred versus when it was written to the database (for example, a long lag may indicate problems with the reporting server).

VXMLElement Table

This table contains one record for each VXML script element visited by a call. For example, if the same element is visited twice in an application script during a call, there will be two separate element records.

Table 5: VXMLElement Table

Field	Type	Null	Index	Description
ElementID	int8	No	PK (Composite ElementID, CallStartDate)	The unique id of a visited element.
CallStartDate	date	No	PK (Composite ElementID, CallStartDate)	The date of the call, for data purging purposes.
SessionID	int8	No	Indexed FK	The unique id of a VXML application session.
CallGUID	char(32) for new installations char(35) for upgrades	No	FK	The global unique id of a call.
ElementName	nvarchar(51)	No	No	The name of an element.
ElementTypeID	int	No	Non-Indexed FK	The type of element.
EnterDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date and time when the element was entered.

Field	Type	Null	Index	Description
ExitDateTime	datetime YEAR to FRACTION(3)	Yes	No	Date and time when the element was exited.
ExitState	nvarchar(51)	Yes	No	The exit state of the element.
NumberOfInteractions	int	Yes	No	The number of interactions while the user visited this element.
ResultID	int	Yes	Non-Indexed FK	Indicates how an element ended.
DBDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date and time of the database operation (when the record was inserted). This is useful for debugging purposes to determine lags between when the event occurred versus when it was written to the database (for example, a long lag may indicate problems with the reporting server).

VXMLElementDetail Table

This table contains one detail record for each script element variable. VarValue holds the String value of the variable and VarDataTypeID specifies the data type of the variable to which the String value can be converted.

Table 6: VXMLElementDetail Table

Field	Type	Null	Index	Description
ElementID	int8	No	Indexed FK (Composite index ElementID, CallStartDate)	The unique id of an element.
CallStartDate	date	No	Yes (Composite index ElementID, CallStartDate)	The date of the call, for data purging purposes.
VarName	nvarchar(51)	No	No	The name of the element variable.
VarValue	nvarchar(255)	Yes	No	The String value of the element variable.

Field	Type	Null	Index	Description
VarDataTypeID	int	No	Non-Indexed FK	The data type of the element variable, such as String, Integer, Boolean.
ActionTypeID	int	No	Non-Indexed FK	The type of action for an element that changes data.
EventDateTime	datetime YEAR to FRACTION(3)	No	Yes	Date and time when the variable was changed.
DBDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date and time of the database operation (when the record was inserted).

VXMLElementFlag Table

This table contains one record for each element in which a flag was activated. The Name field holds the name of the flag.

Table 7: VXMLElementFlag Table

Field	Type	Null	Index	Description
ElementID	int8	No	Indexed FK (Composite index ElementID, CallStartDate)	The unique id for the element in which the flag activated.
CallStartDate	date	No	Yes (Composite index ElementID, CallStartDate)	The date of the call, for data purging purposes.
EventDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date and time when the flag activated.
Name	nvarchar(51)	No	No	The flag name.
PreviousElementName	nvarchar(51)	Yes	No	The name of the previous application element.
DBDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date and time of the database operation (when the record was inserted).

VXMLError Table

This table contains VXML errors that occurred during the life of the VXML application session. The table contains one record for each element in which an error occurred. The ErrorName field holds the name of the error.

Table 8: VXMLError Table

Field	Type	Null	Index	Description
ElementID	int8	No	Indexed FK (Composite index ElementID, CallStartDate)	The unique id for the element in which the error occurs.
CallStartDate	date	No	FK (Composite index ElementID, CallStartDate)	The date of the call, for data purging purposes.
ErrorName	varchar(12)	No	No	Name of an error.
EventDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date and time when the error occurred.
Description	nvarchar(255)	No	No	The detailed error message.
DBDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date and time of the database operation (when the record was inserted). This is useful for debugging purposes to determine lags between when the event occurred versus when it was written to the database (for example, a long lag may indicate problems with the reporting server).

VXMLHotEvent Table

HotEvent is a global event that when caught, executes developer-specified actions. This table contains information (HotEvent name, HotEvent DateTime and the ElementID) about the HotEvent occurred in an element.

Table 9: VXMLHotEvent Table

Field	Type	Null	Index	Description
ElementID	int8	No	Indexed FK (Composite index ElementID, CallStartDate)	The unique id for the element in which the HotEvent occurred.
CallStartDate	date	No	Yes (Composite index ElementID, CallStartDate)	The date of the call, for data purging purposes.

Field	Type	Null	Index	Description
EventDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date and time when HotEvent occurred.
Name	nvarchar(51)	No	No	The name of the HotEvent.
DBDateTime	datetime YEAR to FRACTION(3)	No	No	The date and time of the database operation (when the record was inserted). This is useful for debugging purposes to determine lags between when the event occurred versus when it was written to the database (for example, a long lag may indicate problems with the reporting server).

VXMLHotLink Table

Hotlink is a globally accessible utterance key press that immediately brings the call to a specific part of the call flow or throws an event. This table contains information (HotLink name, HotLink DateTime and the ElementID) about the HotLink that occurred in an element.

Table 10: VXMLHotLink Table

Field	Type	Null	Index	Description
ElementID	int8	No	Indexed FK (Composite index ElementID, CallStartDate)	The unique id for the element in which the hotlink activated.
CallStartDate	date	No	FK (Composite index ElementID, CallStartDate),	The unique id for the element in which the HotLink activated.
EventDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date of the call, for data purging purposes
Name	nvarchar(51)	No	No	The name of the HotLink.

Field	Type	Null	Index	Description
DBDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date and time of the database operation (when the record was inserted). This is useful for debugging purposes to determine lags between when the event occurred versus when it was written to the database (for example, a long lag may indicate problems with the reporting server).

VXMLSession Table

This table contains one record for each application visited by a VXML call. For example, if a call has transferred from one application to another one, the call with the same CallGUID will have two session records.

SIP calls are recorded in the CallEvent Table, on page 10.

Table 11: VXMLSession Table

Field	Type	Null	Index	Description
SessionID	int8	No	PK (Composite SessionID, CallStartDate)	The unique ID of a VXML application session.
CallStartDate	date	No	PK (second field in PK and Composite indexes)	The date of the call, for data purging purposes.
SessionName	nvarchar(96)	No	No	The name of the session assigned by VXML Server.
CallGUID	char(32) for new installations char(35) for upgrades	No	Indexed FK (Composite index CallGUID, CallStartDate);	The global unique id of a call.
StartDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date and time when session starts.
AppName	nvarchar(51)	No	Yes	The name of the VXML application.
EventTypeID	int	No	Non-Indexed FK	The mechanism used to end the application visit.

Field	Type	Null	Index	Description
CauseID	int	No	Non-Indexed FK	The reason that the application visit ended.
EndDateTime	datetime YEAR to FRACTION(3)	Yes	No	The end date and time of the session.
SourceAppName	nvarchar(51)	Yes	No	The name of the application that transferred to this one.
SubSystemName	varchar(41)	No	No	The name of the VXML Service.
MessageBusName	varchar(42)	No	No	The name of the message bus that delivers the VXML data feed message.
DBDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date and time of the database operation (when the record was inserted). This is useful for debugging purposes to determine lags between when the event occurred versus when it was written to the database (for example, a long lag may indicate problems with the reporting server).
Duration	int	Null	No	The length of the session.
LocalTimeZoneOffset	smallint	No	No	The offset in minutes of this the local timezone from UTC timezone. *Replaces LocalTimeZone.*

VXMLSessionVariable Table

This table contains one record for each instance of a session variable. For example, if the same session variable was modified once in an application script during a call, there will be two separate records, one for its initial value when it was created and another for the updated value.

Table 12: VXMLSessionVariable Table

Field	Type	Null	Index	Description
ElementID	int8	No	Indexed FK (Composite index ElementID, CallStartDate)	The identifier of the element in which the session variable changes.
SessionID	int8	No	Indexed FK (Composite index ElementID, CallStartDate)	The unique ID of an IVR application session.
CallStartDate	date	No	Yes (second field in Composite indexes)	The date of the call, for data purging purposes.
VarName	nvarchar(51)	No	No	The name of the session variable that was exited.
VarValue	nvarchar(255)	Yes	No	The value of the session variable.
ActionTypeID	int	No	Non-Indexed FK	The type of action for a session variable that changes data.
EventDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date and time when the session variable changed.
VarDataTypeID	int	No	Non-Indexed FK	The data type of the session variable, such as Integer, String, Boolean.
DBDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date and time of the database operation (when the record was inserted). This is useful for debugging purposes to determine lags between when the event occurred versus when it was written to the database (for example, a long lag may indicate problems with the reporting server).

Field	Type	Null	Index	Description
FromICM	Boolean	No	No	Indicates whether this session variable change originated from Unified ICME; Informix stores these values as 1 or 0, but represents these values as "t" or "f"

VXMLVoiceInteractDetail Table

This table has one record for each Voice Interaction with the caller.

Table 13: VXMLVoiceInteractDetail Table

Field	Type	Null	Index	Description
ElementID	int8	No	Indexed FK (Composite index ElementID, CallStartDate)	The unique ID of a visited element.
CallStartDate	date	No	FK (Composite index ElementID, CallStartDate);	The date of the call, for data purging purposes.
ElaspedTimeMillisecond	int	No	No	The time since the last interaction.
VoiceActionTypeID	int	No	Non-Indexed FK	The type of interaction.
Value	nvarchar(255)	Yes	No	The value of interaction.
DBDateTime	datetime YEAR to FRACTION(3)	No	No	The date and time of the database operation (when the record was inserted).

Summary / Aggregate Tables

The Summary / Aggregate tables are described in the following section:

- ApplicationSummary_15 Table
- ApplicationSummary_Daily Table
- ApplicationSummary_Weekly Table
- ApplicationSummary_Monthly Table, on page 28
- Call_15 Table

- Call_Daily Table
- Call_Weekly Table

Unified CVP reporting server includes a summary process that aggregates data from the Call and VXMLElement tables into new summary tables. These six tables hold summary data on Call metrics and on elements visited in Unified CVP applications.

These metrics include:

- The datetime of the beginning of the summary period.
- The average call length for the calls in this period
- Various totals, including the total number of opt outs, timeouts, and on holds for the calls in this period; the total number of transfers; and the total number of applications visited for the calls in this period

Summary tables use a star schema. Each summary table has a collection of non-numeric attributes and one or more numeric attributes that can be aggregated according to their type. Adding or removing an attribute from a query in a report definition allows a drill up or drill down into the data presented.

For example: the Application Summary tables have the following non-numeric attributes: Dbdatetime | Appname | Sourceappname | Elementname | Elementtypeid | Resultid | Causeid | Exitstate.

The numeric data available to report on those dimensions are: Avg_elapsed and Count.

`Select Appname, avg(avg_elapsed), sum(count)` will yield the average elapsed time and number of occurrences for an application. Adding ElementName to the Select clause (`Select Appname, ElementName, avg(avg_elapsed), sum(count)`) will further elaborate on where time was spent within the application. This can be further qualified by checking for specific Results, Causes, or Exit states.

These summary tables are not pure fact tables in cases where the dimensions are not always ID columns which refer to dimension or lookup tables.

In an upgrade situation, the summary process will start aggregation at the earliest data date within the Call and VXMLelement tables. At most, once every 15 minutes, the summary process will aggregate one day's worth of data from historical records to avoid overtaxing the system by attempting to process too much data.

This means that in a single 24-hour period, the system can summarize 96 days of data at most.

- _Daily tables will be populated one day behind the _15 minute tables.
- _Weekly tables will be populated from _Daily tables once those have been fully populated for the week in question.
- _Monthly tables will be populated from _Weekly tables once those have been fully populated for the month in question.

Retention for summary tables is hardcoded to 60 days for 15 minute summaries, 18 months for daily summaries, 10 years for weekly data, and 40 years for monthly aggregation.

Note

- Take into consideration that it can take some time to collect aggregate-level data from the reporting server.
- Summary tables are built in 15-minute increments using the local time of the reporting server. Latency of source data is not guaranteed. In the event of a failover situation, data may arrive hours after it was initially created. For this reason all summary time periods reflect the time that the source data arrived at the database, which will generally be close to the time that it was created.

ApplicationSummary_15 Table

The ApplicationSummary_15 table is a 15-minute summary of Application/element data, useful for Dominant Path analysis.

Table 14: ApplicationSummary_15 Table

Field	Type	Null	Index	Description
dbdatetime	datetime year to second	Yes	PK	The start of the time period for this row.
AppName	nvarchar(51)	Yes	PK	The name of the VXML application.
SourceAppName	nvarchar(51)	Yes	PK	The name of the application that transferred to this one.
ElementName	nvarchar(51)	No	PK	The name of the element.
ExitState	nvarchar(51)	Yes	PK	The exit state of the element.
ElementTypeID	integer	Yes	PK; FK	The unique ID of an element type.
ResultID	integer	Yes	PK, FK	The unique ID of a result
Count	integer	Yes	No	The number of occurrences for this time period.
CauseId	int	Yes	FK	The unique ID of a cause.
Avg_elapsed	int	yes	No	The average elapsed time for this element.

ApplicationSummary_Daily Table

The ApplicationSummary_Daily table provides a daily summary of Application/element data, useful for Dominant Path analysis.

Table 15: ApplicationSummary_Daily Table

Field	Type	Null	Index	Description
dbdatetime	datetime year to second	Yes	PK	The start of the time period for this row.
AppName	nvarchar(51)	Yes	PK	The name of the VXML application.
SourceAppName	nvarchar(51)	Yes	PK	The name of the application that transferred to this one.
ElementName	nvarchar(51)	Yes	PK	The name of the element.
ExitState	nvarchar(51)	Yes	PK	The exit state of the element.
ElementTypeID	integer	Yes	PK, FK	The unique ID of an element type.
ResultID	integer	Yes	PK, FK	The unique ID of a result.
Count	integer	Yes	No	The number of occurrences for this time period.
CauseId	int	Yes	FK	The unique ID of a cause.
Avg_elapsed	int	yes	No	The average elapsed time for this element.

ApplicationSummary_Weekly Table

A weekly summary of Application/element data, useful for Dominant Path analysis.

Field	Type	Null	Index	Description
dbdatetime	datetime year to second	Yes	PK	The start of the time period for this row
AppName	nvarchar(51)	Yes	PK	The name of the VXML application
SourceAppName	nvarchar(51)	Yes	PK	The name of the application that transferred to this one
ElementName	nvarchar(51)	Yes	PK	The name of the element
ExitState	nvarchar(51)	Yes	PK	The exit state of the element

Field	Type	Null	Index	Description
ElementTypeID	integer	Yes	PK, FK	The unique id of an element type
ResultID	integer	Yes	PK, FK	The unique id of a result
Count	integer	Yes	No	The number of occurrences for this time period
CauseId	int	Yes	FK	The unique id of a cause
Avg_elapsed	int	yes	No	The average elapsed time for this element

ApplicationSummary_Monthly Table

The ApplicationSummary_Monthly table displays a monthly summary of Application/element data, useful for Dominant Path analysis.

Table 16: ApplicationSummary_Monthly Table

Field	Type	Null	Index	Description
dbdatetime	datetime year to second	Yes	PK	The start of the time period for this row.
AppName	nvarchar(51)	Yes	PK	The name of the VXML application.
SourceAppName	nvarchar(51)	Yes	PK	The name of the application that transferred to this one.
ElementName	nvarchar(51)	Yes	PK	The name of the element
ExitState	nvarchar(51)	Yes	PK	The exit state of the element.
ElementTypeID	integer	Yes	PK, FK	The unique ID of an element type.
ResultID	integer	Yes	PK, FK	The unique ID of a result.
Count	integer	Yes	No	The number of occurrences for this time period.
CauseId	int	Yes	FK	The unique id of a cause.
Avg_elapsed	int	yes	No	The average elapsed time for this element.

Call_15 Table

The Call_15 table dispalys a15-minute summary of call activity by SubSystemType and CallType.

Table 17: Call_15 Table

Field	Type	Null	Index	Description
dbdatetime	datetime year to second	Yes	PK	time in 15-minute increments.
SubSystemTypeID	integer	Yes	PK, FK	The unique ID of a Service type.
CallTypeID	smallint	No	PK, FK	The unique ID of a Call type.
AvgCallLength	interval HOUR (3) to FRACTION (3)	Yes	No	The average call length for this period.
TotalOptOut	integer	Yes	No	The total number of Opt Outs in this period.
TotalOnHold	integer	Yes	No	The total number of Holds in this period.
TotalTimeOut	integer	Yes	No	The total number of Time Outs in this period.
TotalError	integer	Yes	No	The total number of errors in this period.
TotalAppVisited	integer	Yes	No	The total number of applications visited in this period.
TotalTransfer	integer	Yes	No	The total number of transfers in this period.
NumCalls	integer	Yes	No	The total number of calls in this period.

Call_Daily Table

The Call_Daily table displays a daily summary of call activity by SubSystemType and CallType

Table 18: Call_Daily Table

Field	Type	Null	Index	Description
dbdatetime	datetime year to minute	Yes	PK	The time in daily increments.

Field	Type	Null	Index	Description
SubSystemTypeID	integer	Yes	PK, FK	The unique ID of a Service type.
CallTypeID	smallint	Yes	PK, FK	The unique ID of a Call type.
AvgCallLength	interval HOUR (3) to FRACTION (3)	Yes	No	The average call length for this period.
TotalOptOut	integer	Yes	No	The total number of Opt Outs in this period.
TotalOnHold	integer	Yes	No	The total number of Holds in this period.
TotalTimeOut	integer	Yes	No	The total number of Time Outs in this period.
TotalError	integer	Yes	No	The total number of errors in this period.
TotalAppVisited	integer	Yes	No	The total number of applications visited in this period.
TotalTransfer	integer	Yes	No	The total number of transfers in this period.
NumCalls	integer	Yes	No	The total number of calls in this period.

Call_Weekly Table

The Call_Weekly table displays a weekly summary of call activity by SubSystemType and CallType.

Table 19: Call_Weekly Table

Field	Type	Null	Index	Description
dbdatetime	datetime year to minute	Yes	PK	The time in weekly increments.
SubSystemTypeID	integer	Yes	PK, FK	The unique ID of a Service type.
CallTypeID	small int	Yes	PK, FK	The unique ID of a Call type.
AvgCallLength	interval HOUR (3) to FRACTION (3)	Yes	No	The average call length for this period.

Field	Type	Null	Index	Description
TotalOptOut	integer	Yes	No	The total number of Opt Outs in this period.
TotalOnHold	integer	Yes	No	The total number of Holds in this period.
TotalTimeOut	integer	Yes	No	The total number of Time Outs in this period.
TotalError	integer	Yes	No	The total number of errors in this period.
TotalAppVisited	integer	Yes	No	The total number of applications visited in this period.
TotalTransfer	integer	Yes	No	The total number of transfers in this period.
NumCalls	integer	Yes	No	The total number of calls in this period.

Call_Monthly Table

The Call_Monthly table displays a monthly summary of call activity by SubSystemType and CallType.

Table 20: Call_Monthly Table

Field	Type	Null	Index	Description
dbdatetime	datetime year to minute	Yes	PK	The time in weekly increments.
SubSystemTypeID	integer	Yes	PK, FK	The unique ID of a Service type.
CallTypeID	small int	Yes	PK, FK	The unique ID of a Call type.
AvgCallLength	interval HOUR (3) to FRACTION (3)	Yes	No	The average call length for this period.
TotalOptOut	integer	Yes	No	The total number of Opt Outs in this period.
TotalOnHold	integer	Yes	No	The total number of Holds in this period.
TotalTimeOut	integer	Yes	No	The total number of Time Outs in this period.

Field	Type	Null	Index	Description
TotalError	integer	Yes	No	The total number of errors in this period.
TotalAppVisited	integer	Yes	No	The total number of applications visited in this period.
TotalTransfer	integer	Yes	No	The total number of transfers in this period.
NumCalls	integer	Yes	No	The total number of calls in this period.

Lookup and Reference Tables

The Lookup and Reference tables are discussed in the following sections:

ActionTypeRef Table

This is a reference table that resolves an ActionTypeID to the text value for an element that changes data.

Table 21: ActionTypeRef Table

Field	Type	Null	Index	Description
ActionTypeID	int	No	PK	The unique id of an action type.
ActionType *Formerly Name*	nvarchar(96)	No	No	The name of the action type.

Table Values (ID, Action Name):

1, "Initialize"

2, "Update"

3, "Return"

CallTypeRef Table

This is a reference table that resolves CallTypeID to a text value.

Table 22: CallTypeRef Table

Field	Type	Null	Index	Description
CallTypeID	small int	No	PK	The unique ID of a call type reference.
CallType	char(32)	No	No	The Call Type.

Table Values (ID, CallType):

1, "Legacy Audio"

2, "Legacy Video"

4, "SIP"

5, "VRU"

6, "VXML"

7, "Basic Video"

8, "Full Video"

CauseRef Table

This table maps a CauseID to the text value for the cause.

Table 23: CauseRef Table

Field	Type	Null	Index	Description
CauseID	int	No	PK	The unique ID of a call event cause.
Cause	nvarchar(96)	No	No	The cause of the event. *Formerly Name*

Table Values (ID, Cause):

0 = "None"

1, "Normal Completion"

2, "Call Abandon"

3, "Call Transferred"

4, "New Transaction"

5, "Busy"

6, "No Answer"

7, "Maintenance"

8, "Net Congestion"

9, "Net Not Obtainable"

10, "Reorder Tone"

11, "Resources Not Available"

12, "Trunks Busy"

13, "Called Party Disconnected"

14, "Max Ports"

15, "Suspended"

16, "Time Out"

17, "Invalidated"

18, "Error"

19, "Video Answered"

20, "Post Call Answer"

21, "Invalid"

22, "Failure"

23, "Audio Recording Start"

24, "Audio Recording Stop"

25, "No Response"

26, "Invalid Number"

27, "Connected"

28, "Caller Canceled"

29, "Whisper Start"

30, "Whisper Done"

31, "Whisper Setup Failed"

32, "Abandon In Whisper"

33, "Whisper Media Error"

1001, "Hang Up"

1002, "Network"

1003, "System"

1004, "Script Type"

1005, "Unknown UApp"

1006, "Script Name"

1007, "Config Param"

1008, "Misconfig Ecc"

1009, "Media File"

1010, "Semantic"

1011, "VXML Format"

1012, "VXML Element"

1013, "Variable Data"

1014, "No Var Data"

1015, "Format"

1016, "Entry Invalid"

1017, "No Entry"

1018, "Media Resource Video" [Unable to perform video-related request due to resource limitations]

1019, "Recording Failed"

1020, "Data Range"

1021, "Timed Out"

1022, "Called Hung Up" [Agent, VRU, or other endpoint hung up on caller; that is, the caller did not hang up first]

1023, "No Answer"

1024, "Busy"

1025, "Transfer"

1026, "Invalid Extn"

1027, "Hang Up Forced"

1028, "After Trans Estab"

1030, "Unsupported Language"

1031, "Media Resource ASR

1032, "Media Resource TTS"

1033, "General ASR TTS"

1034, "Unknown Error

1035, "Missing Configuration"

1036, "Duplicate Record"

1037, "Not in Queue"

1039, "Unknown Callguid"

1040, "CVP System Unavailable"

1041, "CVP App Error"

1042, "CVP App Hang Up"

1043, "Error CVP App Suspended"

1044, "Error CVP No Session Error"

1045, "Error CVP Bad Fetch"

1046, "No Streaming Media Resource TTS"

Device Table

The device for which this resource is measured. This is an IP Address.

Table 24: Device Table

Field	Type	Null	Index	Description
DeviceID	smallint	No	PK	Unique identifier of this device.
Descriptor	char(40)	Yes	No	The IP address of this device.

ElementtypeRef Table

This table maps an ElementTypeID to a text value for the VXML element type.

Table 25: ElementtypeRef Table

Field	Type	Null	Index	Description
ElementTypeID	int	No	PK	The unique id of an element type.

Field	Type	Null	Index	Description
ElementType	nvarchar(96)	No	Yes	The name of the element type . *Formerly Name*

Table Values (ID, ElementType):

0, Start"

1, "End"

2, "Subdialog_Start"

3, "Subdialog_Return"

4, "Decision"

5, "Action"

6, "Custom"

7, "HotLink"

8, "HotEvent"

9, "ElementFlag"

10, "Voice"

11, "VXMLInsert"

12, "ReqICMLabel"

13, "Genera"l

EventTypeRef Table

This is the table to map an EventID to the text value for its name (event type).

Field	Type	Null	Index	Description
EventTypeID	int	No	PK	The unique id of a call event type
EventType	nvarchar(96)	No	No	The name of the event type *Formerly Name*

Table Values (ID, EventType):

0, "New Call"

1, "Connect Failure"

2, "Busy"

3, "No Answer"

4, "Answer"

5, "Abandon"

6, "Disconnect"

7, "Hang Up"

8, "App Transfer"

9, "App Session Complete"

10, "Call Transfer"

11, "Run Script"

12, "Agent Recording"

13, "ICM Recording"

14, "Agent Video"

15, "ICM Video"

16, "Bridge Transfer"

17, "Blind Transfer"

18, "ReqICMLabel"

19, "Audio Recording"

20, "Callback Canceled"

21, "Callback Pending"

22, "Callback In Progress"

23, "Callback Tentative"

24, "Callback Complete"

25, "Callback Recover"

26, "Callback Created"

29, "Max allowed callbacks to this ANI exceeded"

OutgoingECCVariable Table

This table stores the ECC Variables that are returned from Unified CVP to an ICM script.

At present, this table is populated by the courtesy callback studio application element and when the ReqICMLabel element is used in a Call Studio script. Refer to the *CVP Administration and Configuration Guide* for further explanation about using the ReqICMLabel element to pass data to a Unified ICME script.

Table 26: OutgoingECCVariable Table

Field	Type	Null	Index	Description
CallGUID	char(32) for new installations char(35) for upgrades	No	Indexed FK (Composite index CallGUID, CallStartDate)	The global unique id of a call.

Field	Type	Null	Index	Description
CallStartDate	date	No	FK (Composite CallGUID, CallStartDate)	The date of the call, for data purging purposes.
SessionID	int8	No	Yes	The identifier of the session in which the ECC variable changes.
ElementID	int8	No	Yes	The identifier of the element in which the ECC variable changes.
ECCVarName	char(12)	No	No	The name of session variable that was exited.
ECCVarValue	nvarchar(255)	No	No	The value of session variable.
EventDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date and time when the ECC variable changed.
DBDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date and time of the database operation (when the record was inserted). This is useful for debugging purposes to determine lags between when the event occurred versus when it was written to the database (for example, a long lag may indicate problems with the reporting server).

QueueRef Table

QueueRef is a callback lookup table. This table maps QueueID to a text value for the queue in which a callback is waiting. The QueueName stores whatever you decide to call the queues.

Table 27: QueueRef Table

Field	Type	Null	Index	Description
QueueID	smallint	No	PK	The unique ID of a queue.
QueueName	char(40)	Yes	No	The name of the queue.

Resource Table

The resources that are measured include Memory, CPU, DSO and DSP.

Table 28: Resource Table

Field	Type	Null	Index	Description
ResourceID	smallint	No	PK	Unique Identifier.
Descriptor	char(40)	Yes	No	The name of the resource we are measuring (CPU, Memory, DSP, DS0, System).

ResultRef Table

This table maps a ResultID to a text value for a result.

Table 29: ResultRef Table

Field	Type	Null	Index	Description
ResultID	int	No	PK	The unique ID of a result.
Result	nvarchar(96)	No	No	The name of the element result. *Formerly Name*

Table Values (ID, Result):

1, "Normal"

2, "Invalidated"

3, "HotEvent"

4, "HotLink"

5, "Hang Up"

6, "Error"

7, "Transfer"

SubSystemTypeRef Table

This table maps a SubSystemTypeID to a Unified CVP Service type.

Table 30: SubSystemTypeRef Table

Field	Type	Null	Index	Description
SubSystemTypeID	int	No	PK	The unique ID of a Service type.
SubSystem	nvarchar(96)	No	No	The name of the Service type.

Table Values (ID, Name):

0, "SIP"

1, "IVR"

2, "VXML"

3, "OAMP" [Operate, Administer, Maintain, Provision = Operations Console]

4, "Controller"

5, "RPT"

6, "ICM"

7, "ORM" [Element with Unified CVP components that allows the Operations Console to manage the components]

8, "System"

TransferTypeRef Table

This is a reference table to resolve TransferTypeID to a text value.

Table 31: TransferTypeRef Table

Field	Type	Null	Index	Description
TransferTypeID	integer	No	PK	A unique ID of the transfer type.
TransferType	varchar(30)	Yes	No	The type of transfer performed.

Usage Table

This is a fact table of device/resource measurements.

Field	Type	Null	Index	Description
DeviceID	smallint	No	FK	Unique identifier of this device
ResourceID	smallint	No	FK	Unique Identifier

Field	Type	Null	Index	Description
EventDateTime	datetime year to fraction(3)	No	PK	Date and time of this measurement
CallStartDate	date	No	PK	The date of this measurement for purge purposes
ResourceUsed	smallint	Yes	No	The amount of resouce used
ThresholdReached	char(1)	Yes	No	True/False. Was the maximum threshold for this resource reached?
ResourceMax	smallint	Yes	No	The amount of this resource available on this device

UserInputModeRef Table

This table maps a UserInputModeID to the name of the user input mode.

Table 32: UserInputModeRef Table

Field	Type	Null	Index	Description
UserInputModeID	int	No	PK	The unique ID of a user input mode.
UserInputMode	nvarchar(96)	No	No	The name of the user input mode. *Formerly Name*

Table Values (ID, Name):

1, "DTMF"

2, "Voice"

3, "DTMF Voice"

VarDataTypeRef Table

This table maps a VarDataTypeID to the data type of a variable.

Table 33: VarDataTypeRef Table

Field	Type	Null	Index	Description
VarDataTypeID	int	No	PK	The unique ID of a variable data type.

Field	Type	Null	Index	Description
VarDataType	nvarchar(96)	No	No	The name of the variable data type. *Formerly Name*

Table Values (ID, Name):

0, "String"

1, "Int"

2, "Float"

3, "Boolean"

VoiceActionTypeRef Table

This table maps a VoiceActionTypeID to a text value.

Field	Type	Null	Index	Description
VoiceActionTypeID	int	No	PK	The unique ID of a VoiceActionTypeRef.
VoiceActionType	nvarchar(96)	No	No	The name of the call state. *Formerly Name*

Table Values (ID, Name):

1, "No Match"

2, "No Input"

3, "Audio Group"

4, "Input Mode"

5, "Utterance"

6, "Interpretation"

7, "Confidence"

Courtesy CallBack Tables

The following sections describe the Courtesy CallBack tables:

- CallBack Table
- CallBackEvent Table
- CallBackQueue Table

These tables support Courtesy Callback functionality.

Since this data is of an online-transaction-processing (OLTP) nature, it is retained in its own database, the callback database. When the caller registers a request for a callback, that request is stored in the CallBack Table.

A row is placed into the CallBackQueue Table for the call to manage timing and sequencing of calls.

Events that occur during the callback are registered in the CallBackQueue Table. This information can be retrieved using the following query:

```
SELECT Callback.*, CallBackEvent.*,
       CallBackQueue.*
  FROM Callback, CallBackEvent,
       CallBackQueue
 WHERE CallBack.CallGuid= CallGuid
   AND CallBack.SurrogateID=CallBackEvent.SurrogateID
   AND CallBack.SurrogateID=CallBackQueue.SurrogateID;
```

Where CallGuid is replaced by the value of the CallGuid for which information is desired.

Query for number of callbacks currently pending:

```
SELECT count(*)
  FROM CallBack, EventTypeRef
 WHERE CallBack.EventTypeID=EventTypeRef.EventTypeID
   AND EventType in (Callback Pending);
```

Query for a list of failed callbacks with telephone number and failure reason code:

```
SELECT CallGuid, ANI, NbrAttempts, Cause
  FROM CallBack, CauseRef, EventTypeRef
 WHERE CallBack.CauseID=CauseRef.CauseID
   AND CallBack.EventTypeID=EventTypeRef.EventTypeID
   AND EventType in (Callback Canceled);
```

CallBack Table

The callback table is a view of two tables: Callback_Current and Callback_Historical. The two tables are identical; every 30 minutes, data for completed calls is pulled from Callback_Current and moved to Callback_Historical.

One row is generated in this table for each callback that is made.

Table 34: CallBack Table

Field	Type	Null	Index	Description
SurrogateID	serial	No	PK	A unique generated key to replace the CallGuid throughout the Callback schema.
CallStartDate	date	No	PK	The date of the callback for data purging purposes.

Field	Type	Null	Index	Description
OldGuid	char(32)	Yes	No	The Guid for the original scheduled callback. Used by the DB servlet to retrieve information about the old scheduled callback in order to create a new record for the pre-emptive callback.
ANI	varchar(32)	Yes	No	The number to which the callback will be placed.
CallBackType	char(1)	Yes	No	P = preemptive. S = scheduled
Gateway	varchar(40)	Yes	No	The identifier for the gateway. Can be an IP address or other string identifier.
RecordingURL	nvarchar(255)	Yes	No	The URL that points to a wav file of the recorded name of the caller.
ScheduledCallBack DateTime	datetime year to second	Yes	No	The Datetime (including timezone) for a scheduled callback. Not included for preemptive callbacks.
ScheduledCallBackDN	varchar(32)	Yes	No	The DN to which the scheduled callback will be placed. This will invoke an ICM script with this DN.
DBDateTime	datetime year to fraction(3)	Yes	No	The date and time of the database operation.
Location	varchar(32)	Yes	No	The location name assigned to a set of gateways. Used in scheduled callback to select applicable egress gateways for the callback.
NbrAttempts	smallint	Yes	No	The number of attempts made to call back.
EventTypeId	integer	Yes	FK	The unique ID of a event type.
CauseId	integer	Yes	FK	The unique ID of a cause.

CallBackEvent Table

This table holds a record of each callback event that occurs for the call.

This table holds seven days worth of data. Purge daily.

Table 35: CallBackEvent Table

Field	Type	Null	Index	Description
SurrogateID	integer	No	PK, FK	A unique generated key to replace the CallGuid throughout the Callback schema.
CallStartDate	date	No	PK	The date of the callback for data purging purposes
EventDateTime	datetime year to fraction(3)	No	No	The date and time of the event
CauseId	integer	Yes	Yes	See Values below. CauseRef Table
DBDateTime	datetime year to fraction(3)	No	No	The date and time of the database operation.
EventTypeId	integer	No	FK	The unique ID of an event type. See ElementtypeRef Table.

Table Values (ID, CauseID)

0, "busy"

1, "noanswer"

2, "noresponse"

3, "invalid_number"

4, "connected"

5, "caller_canceled"

6, "trunksbusy"

7, "error"

CallBackQueue Table

This table holds data for the queue in which the call sits until its scheduled time or until a slot becomes available.

Table 36: CallBackQueue Table

Field	Type	Null	Index	Description
SurrogateID	integer (serial)	No	PK, FK	
CallStartDate	date	No	PK	The date of the callback for data purging purposes.
CallGuid	char(32) for new installations char(35) for upgrades	Yes	No	Unique ID for the call.
QueueID	smallint	No	FK	
QueueStatus	smallint	No	No	Status in queue: 0 = in_queue, 1 = not_in_queue 2 = Zombie
EnterDateTime	datetime year to second	No	No	The Datetime entered queue.
LeaveDateTime	datetime year to second	Yes	No	The Datetime left queue.
CVPEstimatedWaitTime	smallint	No	No	The CVP-calculated estimated wait time (in seconds) Since enterdatetime. This is generated during the insert, it will not be maintained.
ICMEstimatedWaitTime	smallint	No	No	Unified ICM-calculated estimated wait time (in seconds) Since enterdatetime. This is generated during the insert, it will not be maintained.
ValidationStatus	smallint	No	No	The bitmask result obtained from the Validation method. See sample code that follows this table.
DBDateTime	datetime year to fraction(3)	No	No	The date and time of the database operation.

Validation Method sample code.

This is an example of a bitmask result obtained from the Validation method:

- TOD, Time of Day Error, meaning the callback was scheduled for a time of day when the queue is not open.
- EWT, Estimated Wait Time, indicates if the agent wait time for an agent is long enough to warrant a callback.

```
00000000 00000001 OK
00000000 00000010 ICM_NO_SCHEDULED_ALLOWED
00000000 00000100 ICM_NO_PREEMPTIVE_ALLOWED
00000000 00001000 NOT_IN_QUEUE
00000000 00010000 TOD
00000000 00100000 EWT
00000000 01000000 PROBE_FAILED_NO_RESPONSE
00000000 10000000 PROBE_FAILED_NO_CONFIG
00000001 00000000 EXCEED_CAPACITY_GW
00000010 00000000 EXCEED_CAPACITY_QUEUE
```