Reporting Guidelines

The chapter contains the following topics:

Reporting Server Instance

The Reporting Server instance(Informix dB) must be named as **cvp**, and must not be renamed.

Allow Only Reporting Users to Query Database

In the interests of security, allow only reporting users to generate reports.

Accurate Time Stamps for Reporting and Logging

Cisco Unified Customer Voice Portal (Unified CVP) components do not themselves synchronize machine times. However, customers must provide a cross-component time synchronization mechanism, such as Network Time Protocol (NTP), to ensure accurate time stamps for reporting and logging.

CPU-Intensive Reports

Do not run CPU-intensive reports off the database while the database is receiving data.

Note Reports become more CPU intensive as the complexity associated with producing the report from the information available in the database increases. There is no sharp dividing line between intensive and non-intensive reports. The system performance must remain within the guidelines defined in the *Solution Design Guide for Cisco Unified Contact Center Enterprise*.

Database Backup and Recovery

Issues to keep in mind are:

- Managing your backup strategy
- Turning off the reporting server when doing database recovery

These issues are discussed in Database Backup and Database Recovery.

Database Retention Settings

Ensure that the database is sized conservatively so that it never needs to emergency purge.

Data Security

Ensure data is secure by the following practices:

- Unified CVP offers administrators the ability to choose not to persist sensitive ECC data in the database. Users define ECC variables in Cisco Unified Intelligent Contact Manager Enterprise (Unified ICME) and by default they are not persisted in the Unified CVP database.

  The Caller_input and FromExtVXML ECC variables are subject to many application-dependent uses. For security purposes, flag these two variables as not persistent in Unified ICME. If there is anything in them that must be stored, the routing script can copy the data to an applicable variable for storage in the database.

- Users can reduce the data generated by means of data filters (for VXML Server application detail data filtering. Either adding more exclusive filters, or using fewer inclusive filters, cuts down on the amount of data stored.

- Users can turn off logging of sensitive data containing caller's responses on a per-element basis. The caller's input, such as set of digits representing Social Security numbers or credit card numbers, can be set not to be logged, providing a security layer in case logs are compromised.

Database Sizing Issues

See the discussion in Data Categories and Data Retention. Also see the *Solution Design Guide for Cisco Unified Contact Center Enterprise*.

Report Data Filtering Before Database Storage

Users can reduce the data generated by means of data filters. Either adding more exclusive filters, or using fewer inclusive filters, can cut down on the amount of data stored.

For information on filtering reporting data, see https://www.cisco.com/en/US/products/sw/custcosw/ps1006/products_user_guide_list.html.

Inclusive and Exclusive VXML Filters for Reporting

You use Inclusive and Exclusive VXML filters to control the data that the VXML Server feeds to the Reporting Server. Data feed control is crucial for:

Procedure

- Saving space in the reporting database.
- Preserving messaging communication bandwidth.

Inclusive and Exclusive Filter Configuration

To configure inclusive and exclusive filters for a Reporting Server:

Procedure

Step 1   Choose **Device Management** > **VXML Server**.

The Find, Add, Delete, Edit VXML Servers window opens.

Step 2   You can search for a VXML Server by using the procedure in the Finding a VXML Server topic.

Step 3   From the list of matching records, choose the VXML Server that you want to edit.

Step 4   Click **Edit**.

The VXML Server Configuration window opens to the General Tab.

Step 5 Select the **Configuration Tab**, then configure VXML Server properties.

Step 6 In the **VXML Applications Details: Filters** pane, enter an inclusive filter that defines the VXML elements to include in data sent to the Reporting Server.

Step 7 Optionally, enter an exclusive filter that excludes some of the data specified by the inclusive filter.

Step 8 When you finish configuring filters, click **Save** to save the settings in the Operations Console database or click **Save & Deploy** to save and apply the changes to the VXML Server.

Step 9 Shut down and then start the VXML Server and the primary and backup Call Servers.

Related Topics

VXML Inclusive and Exclusive Filter Rules

Inclusive and exclusive filters operate using the following rules:

- Filters are case sensitive.

- By default, all items but the Start, End, Subdialog_Start and Subdialog_End elements are filtered from reporting data unless they are added to an Inclusive Filter. The Subdialog_Start and Subdialog_End elements are never filtered from reporting data unless Reporting is disabled on the VXML Server.

- The Exclusive Filter takes precedence over the Inclusive Filter. For example, if an application name is in the Exclusive Filter, then all of the items of that applications are excluded from reporting data even if a particular field or element is listed in the Inclusive filter.

- The syntax for Inclusive/Exclusive filters is:

  `Appname.ElementType.ElementName.FieldName`

  or

  `AppName.*.*.SESSION:Varname`

  ✎

  Note This syntax is used to indicate session variables.

- A semicolon (;) should be used to separate each item in a filter. For example, `ElementA ; ElementB` is valid.
- A wildcard (*) can be specified anywhere within the application name, element type, element name, or field name.
- Element types, element names, and field names can contain alphanumeric characters, underscores, and a space character.
- An application name can contain alphanumeric characters and underscores, but the space character is not allowed. For example, `A_aa.B_bb.*C_cc_DD.E_ee_F*` is valid.

Element Types and Filter Names

The following table lists the various VXML element type and their flag.

Table 1: VXML Element Type and Filter Name

Element Type	Filter Name
VXML_TYPE_START	start
VXML_TYPE_END	end
VXML_TYPE_SUBDIALOG_START	SubdialogStart
VXML_TYPE_SUBDIALOG_RETURN	SubdialogReturn
VXML_TYPE_VOICE	voice
VXML_TYPE_DECISION	decision
VXML_TYPE_ACTION	action
VXML_TYPE_VXMLINSERT	vxml
VXML_TYPE_CUSTOM	custom
VXML_TYPE_HOTLINK	hotlink
VXML_TYPE_HOTEVENT	hotevent
VXML_TYPE_ELEMENTFLAG	flag_element
VXML_TYPE_REQICMLABEL	ReqICMLabel
VXML_TYPE_RECORDINGLABEL	RecordingLabel
VXML_TYPE_ELMENT_DEF	element_def

VXML Filter Wildcard Matching Example

The following table provides examples of VXML filter wildcard matching.

Table 2: Examples of VXML Filter Wildcard Matching

Filter	What It Matches
MyApplication.voice.*.*	Matches all voice elements in MyApplication
.voice..*	Matches all Voice elements in all applications.
MyApplication.*.*.var*	Matches all fields in MyApplication that start with with the string var.
MyApplication.*.*.*3	Matches all fields in MyApplication that end with 3.
MyApplication.*.*.SESSION:Company	Matches the Company session variable in MyApplication.

Inclusive and Exclusive VXML Filters for Reporting Example

The following table provides examples of some different combinations of Inclusive and Exclusive filters and the resulting data that the VXML Server feeds to the Reporting Server.

Table 3: Examples of Inclusive and Exclusive VXML Filters for Reporting

Inclusive Filter	Exclusive Filter	Data the VXML Server Feeds To the Reporting Server
`Application1.*.*.*`	None	All Application1 data

`Application1.*.*.*`	`*.*.Element1.*;`
	
	`*.*.Element2.*`
		All Application1 data, except Element1 and Element2

`Application1.*.*.*`	`*.*.Element1.*;`
	
	`*.*.Element2.*;`
	
	`*.*.*.Field1`
		All Application1 data, except Element1, Element2, and Field1

`Application1.*.*.*`	`*.voice.*.*` which matches Element3 and Element4	All Application1 data, except Element3 and Element4

`*.*.Element1.*;`
 	
`*.*.Element2.*;`
 	
`*.*.*.Field1`
	`Application1.*.*.*`	No data for Application1. Other data for other applications, such as Application2, which contain Element1, Element2 and Field1, will be fed.

`*.voice.*.*` which matches Element1, Element2, Element3, and Element4	`*.*.Element3.*;`
	
	`*.*.Element4.*`
		Only Element1 and Element2 and all applications.

`*.voice.*.*` which matches Element1 and Element2	`*.*.*.Field1`	Element1 and Element2, except for Field1, if it exists in those elements
`*.*.Element1.*`	None	Element1
`*.*.Element1.*`	`*.*.*.Field1`	Element1, except for Field1 if it exists in Element1

`*.*.*.Field1`	`*.*.Element3.*;`
	
	`*.*.Element4.*`
		Field1 in any elements except Element3 and Element4

A good strategy for using filters is to create an Inclusive filter that includes the data you want to save in the Reporting database and then create an Exclusive filter to *exclude* portions of the data, for example, sensitive security information such as Social Security Numbers. For example, you would:

- First, create an inclusive filter to include all information:

  MyApp.voice.*.*

- Then, create an exclusive filter to remove credit card and social security numbers information:

```
MyApp.voice.*.CreditCard; MyApp.voice.*.SSN
```

Informix, Operating System Time, and Local Time

Informix displays a datetime that corresponds to the same time zone as the Informix server operating system's time zone, represented in Universal Time Coordinated (UTC).

- If you wish a datetime to be displayed for a time zone other than that of the Informix server operating system, you must use reporting tools or SQL tools (for example, Java).

- If, in your system, you are using more than one Informix server for Unified CVP reporting, it is best if all such server operating systems are set to the same time zone. This helps avoid confusion.

Cisco Unified Customer Voice Portal and SQL Server Data Joining

To join data from a SQL server database and an Informix Database you must use a reporting tool that supports the ability to join data from two heterogeneous databases.

Reporting Password Policy Adherence

Reporting passwords are subject to both the Unified CVP password policy *and* the password policy enforced by the operating system of the computer on which the reporting server resides. For each aspect of the password, the Reporting password *must* meet the requirement of the more restrictive policy.

Related Topics

[Passwords](#)

Purge and Backup Database Maintenance Tasks

The database backup and purge maintenance tasks are created as Windows Scheduled Tasks, and can be viewed in the Scheduled Tasks window (**Start** > **Programs** > **Accessories** > **System Tools** > **Scheduled Tasks**). These jobs log in as SYSTEM.

If the CVPDBNightlyPurge and CVPDBMidDayPurge tasks do not run, then the database will not be purged and will eventually become full, resulting in data loss.

If the CVPDBBackup task does not run the database will not be backed up.

Periodically, you should check the Scheduled Tasks to ensure the Last Run Time was as expected and there are no status messages.

Reporting Isolation Level

Reporting clients should never run with an isolation level of repeatable read because this could hold locks and prevent updates to the data.

Timestamp Synchronization

Call Servers, VXML Servers, and reporting servers must have their clocks synchronized to assure accurate timestamps in both the database and log files.

Since Unified CVP components do not themselves synchronize machine times, you must deploy a cross-component time synchronization mechanism, such as NTP.

Writing Efficient SQL Queries When Creating Reports

Keep these guidelines in mind:

- When writing SQL, developers must organize their WHERE clauses and put the most important join first. The most important join is the one that will reduce the size of the dataset to the least amount of rows.

- Write reports so that every field in the WHERE and ORDER BY clauses uses an indexed field.

- A subset of the data that satisfies any given query can protect the user and the database from generating massive data results. Including the word FIRST in Select statements will return only the amount of data requested. For example, `SELECT FIRST 1000 * FROM Call`.

- The second column in a composite index should never be used in a JOIN statement without the first column.

- Engineers writing database code should treat database, table, and column names as case sensitive--even though the current database is case insensitive--to ensure that the application is portable.

- Many operations hold database locks; therefore, reports should use a wait time of 30 seconds, if possible.

- It is possible to capture gigabytes of Unified CVP data in a single day. Any query against the database should target time ranges and subsets of data that will return in a reasonable time. Datetime columns are crucial selections. Sorting or grouping large quantities of data may exceed the capacity of the reporting server database as delivered.

- All sessions that connect to the reporting database should initiate with two statements:

  `SET ISOLATION DIRTY READ; SET LOCK MORE TO WAIT 30`.

  This prevents reporting queries from interfering with CallServer message persistence and improves the performance of reporting queries.

> ⚠ **Warning**   Do not ever set the Isolation level to Repeatable Read.

- The internal ID generator limits the amount of total VXML subsystems to 8,000 per deployment.

Warning   Do not ever write a SQL statement that selects into temp without specifying the 'no log' option.

Zero Duration Calls and Writing Reports

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 zero duration, report writers will know to exclude such a call from consideration in cases where it would otherwise skew metrics.