CommunityWFM

Integration Specification
Cisco UCCX
for SQL Server

For CommunityWFM Software Version 4.4+/5.0+

March 2024

3400 Waterview Parkway, Suite 325
Richardson, Texas 75080

phone 877-668-6870
web CommunityWFM.com

©2024, CommunityWFM
All Rights Reserved.

Table of Contents

About this Document 2

CommunityWFM and Cisco UCCX Integration 2

Historical Data Collection 2

Contact Service Queue Historical Collection 2

Cisco UCCX contact offered query 2

Cisco UCCX contact abandoned query 3

Cisco UCCX contact AHT query 3

Agent State Transaction Data Collection 4

Importing Configuration Data 4

Contact Service Queue Data 4

About this Document

The objective of this document is to outline the method and details of CommunityWFM integration to Cisco UCCX. The document assumes that the reader has some basic understanding of the Cisco database layout and a working understanding of SQL query syntax.

CommunityWFM and Cisco UCCX Integration

CommunityWFM integrates with the Cisco UCCX platform using direct queries against the Cisco DB_CRA database. Tables within the database satisfy the data collection requirements to fully integrate the CommunityWFM application with the Cisco UCCX platform.

Each component of data collection and the required database and tables are described below.

Historical Data Collection

CommunityWFM collects historical contact volume data from the DB_CRA database to be used in volume and handle time reports and for future forecasting. The Community Historical Data Collection Service (a .NET Windows service) executes the query for each defined data collection point every 15 minutes and loads the results into vendor-neutral tables inside the Community product database.

Contact Service Queue Historical Collection

For CSQ based data collection, the queries run against the ContactCallDetail, ContactQueueDetail and ContactServiceQueue tables and are implemented as follows:

Cisco UCCX contact offered query

```
SELECT DATEADD(minute, CONVERT(int, DATEPART(minute, startDateTime) / 15) * 15,
DATEADD(hour, DATEPART(hour, startDateTime), CONVERT(datetime,
FLOOR(CONVERT(float, startDateTime))))) AS 'Slot',CSQ.contactServiceQueueId AS
'Skill',CQD.profileID,COUNT(*) AS 'CallVolume' FROM ContactQueueDetail CQD JOIN
ContactCallDetail CCD ON CCD.sessionID = CQD.sessionID AND CCD.sessionSeqNum =
CQD.sessionSeqNum AND CCD.profileID=CQD.profileID AND CCD.nodeID=CQD.nodeID
JOIN ContactServiceQueue CSQ ON CSQ.recordId = CQD.targetID WHERE
CQD.targetType = 0 ; GROUP BY DATEADD(minute, CONVERT(int, DATEPART(minute,
startDateTime) / 15) * 15, DATEADD(hour, DATEPART(hour, startDateTime),
```

CONVERT(datetime, FLOOR(CONVERT(float, startDateTime)))),CSQ.contactServiceQueueId,CQD.profileID ORDER BY 'Slot'

Cisco UCCX contact abandoned query

SELECT DATEADD(minute, CONVERT(int, DATEPART(minute, startDateTime) / 15) * 15, DATEADD(hour, DATEPART(hour, startDateTime), CONVERT(datetime, FLOOR(CONVERT(float, startDateTime))))) AS 'Slot',CSQ.contactServiceQueueId AS 'Skill',CQD.profileID,COUNT(*) AS 'Abandoned' FROM ContactQueueDetail CQD JOIN ContactCallDetail CCD ON CCD.sessionID = CQD.sessionID AND CCD.sessionSeqNum = CQD.sessionSeqNum AND CCD.profileID=CQD.profileID AND CCD.nodeID=CQD.nodeID JOIN ContactServiceQueue CSQ ON CSQ.recordId = CQD.targetID WHERE CQD.targetType = 0 AND CQD.disposition=1 ; GROUP BY DATEADD(minute, CONVERT(int, DATEPART(minute, startDateTime) / 15) * 15, DATEADD(hour, DATEPART(hour, startDateTime), CONVERT(datetime, FLOOR(CONVERT(float, startDateTime))))),CSQ.contactServiceQueueId,CQD.profileID ORDER BY 'Slot'

Cisco UCCX contact AHT query

SELECT DATEADD(minute,CONVERT(int,DATEPART(minute,ACD.startDateTime) / 15) * 15,DATEADD(hour, DATEPART(hour, ACD.startDateTime), CONVERT(datetime, FLOOR(CONVERT(float, ACD.startDateTime))))) AS 'Slot',CSQ.contactServiceQueueId AS 'Skill',SUM(ACD.talkTime) + SUM(ACD.holdTime) + SUM(ACD.workTime) AS 'HandleTime' FROM AgentConnectionDetail ACD JOIN ContactQueueDetail CQD ON CqD.sessionID=ACD.sessionID AND CqD.sessionSeqNum = ACD.sessionSeqNum AND CqD.profileID = ACD.profileID AND CqD.nodeID = ACD.nodeID AND CqD.qIndex = ACD.qIndex JOIN ContactServiceQueue CSQ ON CSQ.recordID = CQD.targetID WHERE CQD.targetType = 0 AND CQD.disposition=2 ; GROUP BY DATEADD(minute, CONVERT(int, DATEPART(minute, ACD.startDateTime) / 15) * 15, DATEADD(hour, DATEPART(hour, ACD.startDateTime), CONVERT(datetime, FLOOR(CONVERT(float, ACD.startDateTime))))), CSQ.contactServiceQueueId ORDER BY 'Slot'

Note: The historical collection service implements replacement characters for the WHERE condition. The data collection service resolves these replacement tokens when it executes the query for individual Contact Service Queues.

Agent State Transaction Data Collection

CommunityWFM collects agent state transactions from the DB_CRA database to compare against schedule intervals in order to provide agent schedule adherence reporting. The Community Adherence Collection Service (a .NET Windows service) executes the query on a user-defined interval (typically between 5 and 30 seconds) and loads the results into vendor-neutral tables inside the CommunityWFM product database.

The state transaction collection service executes the following query against the AgentStateDetail table:

```
Select R.resourceLoginId AS 'login_id', DATEADD(minute, -1 * ASD.gmtOffset,
ASD.eventDateTime) AS 'eventDateTime', CONVERT(nvarchar, ASD.eventType, 20) +
CONVERT(nvarchar, ASD.reasonCode, 20) AS 'eventType' from AgentStateDetail ASD
JOIN Resource R ON R.resourceID = ASD.agentId AND R.profileID = ASD.profileID where
DATEADD(minute, - 1* ASD.gmtOffset, ASD.eventDateTime) > '" +
FormatDateTime(m_last_dt) + "' order by eventDateTime
```

Note: The FormatDateTime(m_last_dt) is c# code that formats the last read time into an SQL string to only pull transactions that occur after the last time the service executed.

Importing Configuration Data

CommunityWFM supports the ability to import Contact Service Queue data directly from the Cisco database into CommunityWFM. Configuration data is retrieved from the DB_CRA database using the DSN provided by the end user.

Contact Service Queue Data

```
SELECT DISTINCT CSQName AS 'Name', contactServiceQueueId AS 'ACDSkillId' FROM
ContactServiceQueue WHERE dateInactive IS NULL AND active=1 ORDER BY CSQName
```