



Integration Specification

I3 CIC

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

Table of Contents

About this Document.....	2
CommunityWFM and I3 CIC Integration.....	2
Historical Data Collection	2
Contact Queue Historical Collection.....	2
I3 CIC default contact historical query.....	2
Agent State Transaction Data Collection	3
Importing Configuration Data.....	3
Contact Queue Data	3
Agent Information.....	4

About this Document

The objective of this document is to outline the method and details of the CommunityWFM integration to I3 Customer Interaction Center (CIC). The document assumes that the reader has some basic understanding of the I3 CIC database layout and a working understanding of SQL query syntax.

CommunityWFM and I3 CIC Integration

CommunityWFM integrates with the I3 CIC platform using direct queries against the CIC I3_IC database via a .net native SQL provider. Tables within the database satisfy the data collection requirements to fully integrate the CommunityWFM application with the I3 CIC 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 I3_IC 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 CommunityWFM product database.

Contact Queue Historical Collection

For queue-based data collection, the query runs against the IWrkgrpQueueStats, table and is implemented as follows:

I3 CIC default contact historical query

```
SELECT dIntervalStart AS 'Slot', nDuration AS 'Duration', cName AS 'Skill', nAnsweredAcad + nAbandonedAcad + nFlowOutAcad AS 'CallVolume', nAbandonedAcad + nFlowOutAcad AS 'Abandoned', tAgentOnAcadCall + tAgentInAcw AS 'HandleTime' FROM IWrkgrpQueueStats WHERE dIntervalStart > '#DtStart#' AND dIntervalStart <= '#DtEnd#' AND cName = '#AcadSkill#' #SiteID# AND cReportGroup='*' ORDER BY Slot ASC, Skill ASC
```

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 queues.

Agent State Transaction Data Collection

CommunityWFM collects agent state transactions from the I3_IC 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 AgentActivityLog table:

```
select I3TimeStampGMT, UserId, StatusKey, StatusDateTimeGMT, EndDateTimeGMT,
CASE WHEN StateDuration = 0 THEN 1 ELSE 0 END AS 'IsCurrentTransaction' FROM
AgentActivityLog WHERE I3TimeStampGMT > '' + FormatDateTime(m_last_dt) + '' order
by StatusDateTimeGMT
```

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 was executed.

Importing Configuration Data

CommunityWFM supports the ability to import contact queue information and agents directly from the I3 database into CommunityWFM. Configuration data is retrieved from the I3_IC database using the database credentials provided by the end user.

Contact Queue Data

```
SELECT ObjectName AS 'ACDSkillId', ObjectName AS 'Name' from Object where
ObjectType IN (1,2,3,4,5,6,7) AND IsReal=1
```

Agent Information

```
SELECT lastname AS 'Last_Name', firstname AS 'First_Name', ICUserId AS 'Employee_Id'  
FROM Individual WHERE IndivTypeId=2 AND DATALENGTH(LastName) > 2 AND  
DATALENGTH(FirstName) > 2 ORDER BY LastName, FirstName
```