



# Integration Specification Intermedia

For CommunityWFM Software Version 5.x+

June 2025

3400 Waterview Parkway, Suite 325  
Richardson, Texas 75080

phone 877-668-6870  
web [CommunityWFM.com](http://CommunityWFM.com)

## Table of Contents

About This Document.....	2
CommunityWFM and Intermedia Integration.....	2
Thrio Configuration Page.....	3
Historical Data Collection .....	4
Agent State Transaction Data Collection .....	4
Importing Configuration Data.....	5

## About This Document

The objective of this document is to outline the method and details of the CommunityWFM integration to an Intermedia system. The document assumes that the reader has some basic understanding of the Intermedia contact center platform.

## CommunityWFM and Intermedia Integration

CommunityWFM integrates with the Intermedia platform using a hybrid, of their REST APIs written for WFM integrations, along with Community Cloud Connect (CCC) to receive a push real-time agent state data. The Intermedia API being used is the "Contact Center API," which uses HTTP requests to communicate with the contact center platform. Communication to Intermedia consists of sending in a username and password and an account ID for authentication (provided by Intermedia and the customer). CommunityWFM will supply the customer with the base URL and API key of the CCC for pushing real-time data. All times are returned in Unix time UTC.

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

## Thrio Configuration Page

**CommunityWFM**

### Intermedia

Establish connections to data sources

#### Real-time adherence integration

Server Name:	<input type="text" value="LOCALSERVER\SQLEXPRESS"/>
Database Name:	<input type="text" value="WFMSG.Staging"/>
User Id:	<input type="text" value="WFMConnection"/>
Password:	<input type="text" value="CommunityWFM"/>
Transaction Collection Interval (seconds):	<input type="text" value="45"/>
Last Adapter Read Time:	23 Jun 2025 01:34:47 AM

#### Historical contact volume data collection

Base URL:	<input type="text" value="https://pop1-apps.mycontactcenter.net/API/v3"/>
Account ID:	<input type="text" value="CUSTOMERID"/>
UserName:	<input type="text" value="CustomerUserName"/>
Password:	<input type="text" value="*****"/>
ACD Historical Reporting Interval:	<input type="text" value="15"/>

Use the Data Source Type Default Query for data collection  
 Use the custom query for the associated Data Source for data collection

with temp\_table as (SELECT [ivr\_id],dbo.fnDateTimeToSlotTimeEx(qh\_queuingdateutc, 15) as [SlotTime],[qh\_queueid] as [Skill],[qh\_queuingdateutc],[ah\_answereddateutc],[ah\_wrapupendutc], DATEDIFF(second, [ah\_answereddateutc],[ah\_wrapupendutc]) as [Duration],[abandoned] FROM [t\_adapter\_call\_data\_intermedia] where qh\_queueid = '#Skill#' and qh\_queuingdateutc >= dateadd(hour, -2, '#StartDateTime#') and qh\_queuingdateutc < '#EndDateTime#') SELECT SlotTime as [Slot],Skill,COUNT(slottime) as [CallVolume], SUM(abandoned) as [Abandoned], SUM(Duration) as [HandleTime] FROM temp\_table GROUP BY Skill, SlotTime ORDER BY SlotTime asc;

## Historical Data Collection

CommunityWFM collects individual call record data from the Intermedia REST API 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 request every 15 minutes and summarizes the results into vendor-neutral tables inside the CommunityWFM product database.

The following query is used to summarize the retrieved data:

```
with temp_table as (SELECT [ivr_id],dbo.fnDateTimeToSlotTimeEx(qh_queuingdateutc, 15)
as [SlotTime],[qh_queueid] as [Skill],[qh_queuingdateutc], [ah_answereddateutc],
[ah_wrapupendutc], DATEDIFF(second, [ah_answereddateutc],[ah_wrapupendutc]) as
[Duration],[abandoned] FROM [t_adapter_call_data_intermedia] where qh_queueid =
'#Skill#' and qh_queuingdateutc >= dateadd(hour, -2, '#StartDateTime#') and
qh_queuingdateutc < '#EndDateTime#') SELECT SlotTime as [Slot],Skill,COUNT(slottime)
as [CallVolume], SUM(abandoned) as [Abandoned], SUM(Duration) as [HandleTime]
FROM temp_table GROUP BY Skill, SlotTime ORDER BY SlotTime asc;
```

## Agent State Transaction Data Collection

CommunityWFM collects agent state transactions from the CCC database to compare against scheduled intervals to provide agent schedule adherence reporting. The Community Adherence Collection Service (a .NET Windows service) executes the request based upon the interval time configured in the UI. The adapter then loads the results into vendor-neutral tables inside the CommunityWFM product database.

The data retrieved from this call (realTime) includes the following columns:

- agent-id – Unique identifier for the agent
- Activity – Agent state
  - Possible states include:
    - Available
    - Logged out
    - Logged on
    - On Call
    - Wrap-up
    - Lunch

- Dial Out
- (Any custom states created by a customer)
- Reason Code
  - This comes back as numbers like
    - 16
    - 32
    - 33
    - 1
    - etc

## Importing Configuration Data

CommunityWFM supports the ability to import queue information and agents directly from the Intermedia system into CommunityWFM. Configuration data is retrieved by making RESTful API calls to pull back the data.