



# Integration Specification NEC Business ConneCT (BCT)

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](https://CommunityWFM.com)

## Table of Contents

About this Document.....	2
CommunityWFM and NEC BCT Integration.....	2
Historical Data Collection .....	2
Agent State Transaction Data Collection .....	4
Importing Configuration Data.....	5
Queue Data Collection .....	5
Agent Information.....	5

## About this Document

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

## CommunityWFM and NEC BCT Integration

CommunityWFM integrates with the NEC BCT platform using direct queries against the BCT United database via a .net native SQL provider. Tables within the database satisfy the data collection requirements to fully integrate the CommunityWFM application with the NEC BCT 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 NEC BCT 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 RawHistData\_RouterCallDataView and RawHisData\_ACWDataView views and is implemented as follows:

## BCT default contact historical query

```

SELECT DATEADD(MINUTE, DATEDIFF(minute, 0, DateTimeCallArrivedAtThisModule) / 15 * 15, 0)
AS 'Slot'

        ,RCDV.RouterName AS 'Skill'

        ,COUNT(*) AS 'ContactsOffered'

        ,MAX(ISNULL(ABANS.NotHandled, 0)) AS 'NotHandled'

        ,SUM(DATEDIFF(SECOND, DateTimeCallAnsweredAtUserPhone,
DateTimeCallEndedAtUserPhone)) AS 'TalkTime'

        ,MAX(ISNULL(ACW.ACW, 0)) AS 'ACW'

        ,SUM(DATEDIFF(SECOND, DateTimeCallAnsweredAtUserPhone,
DateTimeCallEndedAtUserPhone)) + MAX(ISNULL(ACW.ACW, 0)) AS 'TotalHandlingTime'
FROM   dbo.RawHistData_RouterCallDataView RCDV
LEFT   JOIN ( SELECT COUNT(*) AS 'NotHandled'

                ,RouterName

                ,DATEADD(MINUTE, DATEDIFF(minute, 0,
DateTimeCallArrivedAtThisModule) / 15 * 15, 0) AS 'Slot'

                FROM   dbo.RawHistData_RouterCallDataView ABS

                WHERE UPPER(CallResult) IN ('ABANDONED', 'ERROR',
'NOAGENT', 'TIMEOUT')

                GROUP BY RouterName

                ,DATEADD(MINUTE, DATEDIFF(minute, 0,
DateTimeCallArrivedAtThisModule) / 15 * 15, 0)) ABANS ON ABANS.Slot = DATEADD(MINUTE,
DATEDIFF(minute, 0, DateTimeCallArrivedAtThisModule) / 15 * 15, 0) AND ABANS.RouterName =
RCDV.RouterName
LEFT   JOIN ( SELECT SUM(DATEDIFF(second, ACWStartTime, ACWEndTime)) AS 'ACW'

                ,RouterName

                ,DATEADD(MINUTE, DATEDIFF(minute, 0,
ACWStartTime) / 15 * 15, 0) AS 'Slot'

                FROM   dbo.RawHistData_ACWDataView

                GROUP BY DATEADD(MINUTE, DATEDIFF(minute, 0,
ACWStartTime) / 15 * 15, 0)

```

```

,RouterName) ACW ON ACW.Slot =
DATEADD(MINUTE, DATEDIFF(minute, 0, DateTimeCallArrivedAtThisModule) / 15 * 15, 0) AND
ACW.RouterName = RCDV.RouterName
WHERE UPPER(CallResult) IN ('ANSWERED', 'ABANDONED')
GROUP BY RCDV.RouterName
,DATEADD(MINUTE, DATEDIFF(minute, 0, DateTimeCallArrivedAtThisModule) / 15
* 15, 0)
ORDER BY RCDV.RouterName, Slot

```

**Note:** The historical collection service implements replacement characters for the WHERE condition (not included above). 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 NEC BCT database to compare against schedule intervals 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 RawHistData\_AgentGlobalStatusView view:

```

SELECT UserLoginName
,DateTimeUserStateStarted
,CONVERT(nvarchar, StateId) + ISNULL(UserNotReadyReason, '') AS 'State'
FROM dbo.RawHisData_AgentGlobalStatusView
ORDER BY UserLoginName, DateTimeUserStateStarted

```

**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 queue information and agents directly from the NEC BCT database into CommunityWFM. Configuration data is retrieved from the database using the database credentials provided by the end user.

### Queue Data Collection

```
SELECT rtr_Description AS ACDSkillId, rtr_Description AS Name FROM RouterView
```

### Agent Information

```
SELECT LoginName AS 'Employee_ID', FirstName AS First_Name, LastName AS  
Last_Name FROM RawData_UserNameAndRolesView ORDER BY Last_Name ASC
```