



# Cisco Unified Workforce Optimization

Historical and Real-Time Data Import Reference Guide Release 11.5

First Published: July 28, 2016

Last Updated: January 5, 2018

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: [www.cisco.com/go/trademarks](http://www.cisco.com/go/trademarks). Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Historical and Real-Time Data Import Reference Guide

© 2016, 2018 Cisco Systems, Inc. All rights reserved.

# Contents

---

- Introduction** ..... 5
- GIS API** ..... 7
  - Historical Data Capture ..... 7
    - Requirements ..... 8
    - File Usage ..... 8
    - Text File Locations ..... 8
    - ACD Numbering ..... 9
    - Text File Details ..... 9
    - Data Type Definitions ..... 10
  - AgentProductivity.AGENT File ..... 11
    - File Name Format ..... 11
    - File Header ..... 12
    - Column Header ..... 12
    - Data Lines ..... 13
    - Required Columns ..... 13
    - Optional Columns ..... 14
    - Example ..... 17
  - ServiceHistorical.SERVICE File ..... 18
    - File Name Format ..... 18
    - File Header ..... 18
    - Column Header ..... 19
    - Data Lines ..... 20
    - Required Columns ..... 20
    - Optional Columns ..... 21
    - Missing Columns ..... 22
    - Example ..... 23
  - AgentState.EVENT File ..... 23

File Name Format .....	23
File Header .....	24
Column Header .....	24
Data Lines .....	24
Required Columns .....	25
Optional Columns .....	25
Agent State Values .....	25
Example .....	26
Integrating Your HRMS with WFM .....	27
Import File Details .....	28
Export File Details .....	29
Real-Time Data Capture API .....	30
API Definition .....	30
Establish Session .....	30
Submit Agent State Information .....	31
Examples .....	32
Destroy Session .....	34
<b>WFM Historical Import Template (WHIT) .....</b>	<b>35</b>
Software Requirements .....	35
WHIT Components .....	35
WFM Historical Import Template .....	35
WHIT Executable .....	36
Importing Data with WHIT .....	36
Installing WHIT .....	36
Preparing the WHIT Spreadsheet .....	37
Running WHIT .....	38

# Introduction

---

You can import and integrate historical and real-time data from any automatic call distributor (ACD) to the Workforce Management (WFM) database using one of the following tools.

- **GIS API.** The GIS API is used to import both historical and real-time data to the WFM database at specified intervals. WFM uses this data to generate reports and statistics.
- **Workforce Management Historical Import Tool (WHIT).** WHIT is used to convert existing historical data from a non-WFM system into data files that can be loaded into WFM. You might want to use WHIT if you are installing a new WFM system.



# GIS API

The GIS API is part of WFM and requires no separate installation or executable to function.

**Note:** File paths shown in this document are the default file path. WFM can be installed to a custom location, so your file path might differ.

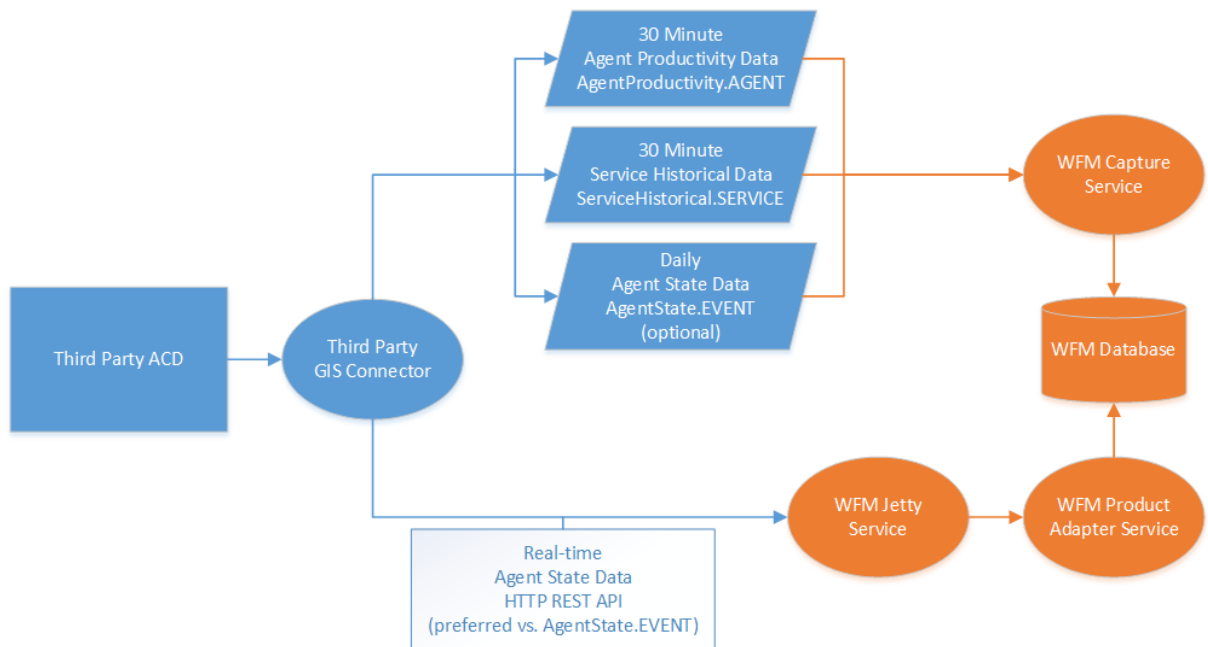
## Historical Data Capture

The following tasks are required to import historical data from an ACD to the WFM database:

1. Write scripts that retrieve the historical data from the ACD.
2. Create a batch process to run the scripts at specified intervals to retrieve the historical data and convert it to the required comma-delimited text file format.
3. Place the comma-delimited text files in a predefined folder on the WFM server using FTP or some other tool.

The GIS API processes the comma-delimited text files and writes the historical data to the WFM database.

The following graphic is a high-level depiction of the historical data integration process.



## Requirements

The following are conditions that must be met in order to assure successful call and non-call data capture.

- Service historical data is calculated from the summation of agent service productivity data. As a result, both the AgentProductivity.AGENT and ServiceHistorical.SERVICE files must be available for a given period for capture to take place. If both files are found, but data is missing from one, certain values might be taken to be zero (the sum of zero items equals zero). It is also important that the agent productivity and service historical data are consistent in terms of which agents and services have data.
- Only one file of each type should be written per historical period. A second file of the same type for the same period will cause data from the first file to be overwritten.
- The same ACD number should be used in the file name and file headers for all files.

## File Usage

Use the three capture files as follows.

File	Use
.SERVICE	This file is required and used for service queue statistics.
.AGENT	This file is required in order to process the files for the given intervals. It is not necessary for the file to contain any data rows, only the header row. However, if you want to track individual agent productivity, then it should include data rows.
.EVENT	<p>This file is optional. It can be used as one method to capture real-time events to calculate agent conformance and adherence.</p> <p>The GIS API processes this file as soon as it drops into the Reports folder. The frequency of new files dropping into the Reports folder determines how close to real time the data is.</p> <div style="background-color: #e0f0e0; padding: 10px; border: 1px solid #ccc;"> <p><b>Note:</b> The preferred method of capturing real-time events is through the use of the Real-Time Data Capture API. See <a href="#">Real-Time Data Capture API</a> for more information.</p> </div>

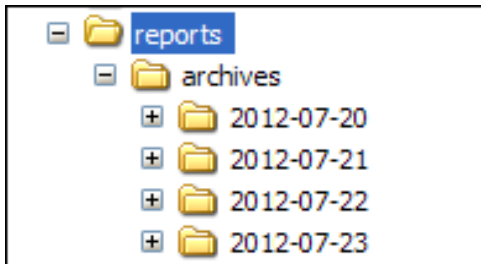
## Text File Locations

The comma-delimited text files must be placed in the following location on the server that hosts the WFM Capture service:



C:\Program Files (x86)\Common Files\WFM\reports\

When the text files are processed by WFM, they are automatically moved to the archives folder in a subfolder named for the date on which they were processed, as shown in the figure below. The format of the archive folder name is YYYY-MM-DD.



## ACD Numbering

ACDs from which data is imported are assigned numbers to identify them as the data source. The ACD numbers are assigned by the author of the scripts that extract the data from the ACD and write it to the comma-delimited text files.

Even if there is only one ACD, that ACD must be assigned a number. In the examples in this document, the ACD is assigned the number 0. At this time, the ACD number is not used by the GIS API, but it might be used in the future. It is currently a best practice to use 0 in GIS file names and file headers.

Additionally, the agent identifiers (ACD ID in WFM) and service queue identifiers (Service Queue ID in WFM) in GIS text files must be unique across the WFM installation. In a multiple-ACD environment, this can be accomplished by having the scripts prepend the ACD number and a delimiter character to the agent's or service queue's identifier in the GIS text files, and setting up agents/service queues accordingly in WFM.

## Text File Details

This section provides the details of the three comma-delimited text files used by the GIS API. These files are the following:

- AgentProductivity.AGENT
- ServiceHistorical.SERVICE
- AgentState.EVENT

The data in the files is by 30-minute intervals. This interval is not configurable.

**Note:** The date format in the text files (YYYY-MM-DD) might be changed by a spreadsheet application such as Microsoft Excel to the format that is set in the application as the default. Be

aware of this possibility and make sure that the date columns are configured so that the date format is correct.

## Data Type Definitions

The data types noted in this section have specific definitions. These definitions are detailed in the following table.

Data Type	Definition
date	A date expressed in YYYY-MM-DD format.
double	A floating point value expressed in a manner that can be parsed as a signed double-precision floating point value. Any value allowed as a long is also allowed as a double. The + sign should be omitted.
integer	An integer expressed as a signed decimal integer value, without leading zeros, whose value is at least -0x8000 and at most 0x7fff. The + sign should be omitted.
long	An integer expressed as a signed decimal integer value, without leading zeros, whose value is at least -0x80000000 and at most 0x7fffffff. The + sign should be omitted.
nonnegative double	A floating point value expressed in a manner that can be parsed as a signed double-precision floating point value, whose value is at least 0x00000000. Any value allowed as a non-negative long is also allowed as a non-negative double. The + sign should be omitted.
nonnegative long	An integer expressed as a signed decimal integer value, without leading zeros, whose value is at least 0x00000000 and at most 0x7fffffff. The + sign should be omitted.
null	The literal value NULL (case insensitive).
positive double	A floating point value expressed in a manner that can be parsed as a positive signed double-precision floating point value. Any value allowed as a positive long is also allowed as a positive double. The + sign should be omitted.
positive long	An integer expressed as a signed decimal integer value without leading zeros, whose value is at least 0x00000001 and at most 0x7fffffff. The + sign should be omitted.

Data Type	Definition
time	A time expressed in any of the following formats: <ul style="list-style-type: none"> <li>• hh:mm</li> <li>• hh:mm:ss</li> <li>• hh:mm:ss.sss</li> </ul>
timestamp	A time stamp consisting of the following data types: <date> <time> <tzoffset> If the tzoffset is omitted, the date and time are assumed to be in the default time zone of the WFM server.
tzoffset	A time zone offset expressed in +HHMM, -HHMM, HHMM format or the literal values GMT or UTC.

## AgentProductivity.AGENT File

The AgentProductivity.AGENT file contains agent productivity data by 30-minute intervals from 00:00 to 23:59.

### File Name Format

<date/time><tzoffset>\_<ACD#>\_AgentProductivity.AGENT

Element	Description
<date/time>	The date and time the file was generated, in YYYYMMDDHHMM format. Example: 201301151430 (14:30 on January 15, 2013). Time is in 24-hour format.
<tzoffset>	The time zone (where the ACD is located) date/time offset from GMT in AHHMM format, where: <ul style="list-style-type: none"> <li>• A is either P (positive) or N (negative), characterizing the offset from GMT</li> <li>• HH is the number of offset hours</li> <li>• MM is the number of offset minutes</li> </ul> Example: If the ACD is in GMT, then the <tzoffset> is P0000. If the ACD is in EST (GMT-0500), then the <tzoffset> is N0500.
<ACD#>	The ACD number. Can be zero (0) or a positive number with as many digits as desired.

## File Header

AGENT DATE: <date> INTERVAL: <time> TZOFFSET: <tzoffset> ACD: <ACD#>

Element	Description
<date>	The date the file was generated, in YYYY-MM-DD format.
<time>	The beginning of the 30-minute interval covered by the data, in HH:MM 24-hour format.
<tzoffset>	The time zone (where the ACD is located) offset from GMT, in +HHMM, -HHMM, or HHMM format. If plus or minus is not specified, the offset is assumed to be plus (positive).
<ACD#>	The number of the ACD that is the source of the data. Currently this value is not used, but there must be a number. Use 0 as a default value if no other number is valid. It is recommended that it be the same as the ACD# used in the file name.

The file header must be the first non-blank line in the file.

**Note:** The AGENT DATE, INTERVAL, and TZOFFSET values must represent a period start time that matches those in the file name as well as a historical period supported by WFM. It is not required that the individual values match, but only to mean the same point in time. For example, if the file is named:

```
201301231800N0500_0_AgentProductivity.AGENT
```

and the header is:

```
AGENT DATE: 2013-01-23 INTERVAL: 23:00 TZOFFSET: 0000 ACD: 0
```

there is no error, because each is expressing the same point in time in different time zones. The time 18:00 in the time zone N0500 (a 5-hour negative offset from GMT) is the same as the time 23:00 in the GMT time zone.

## Column Header

<columnname1>,<columnname2>,<columnname3>...,<columnnameN>

Where <columnname1> through <columnnameN> are the names of columns represented in the file.

Column names are not restricted to required and optional columns. You can add additional unrecognized columns and associated data (for example, agent names or service names) to make the GIS files more human-readable. The GIS API ignores these columns and their associated data.

Column order is not specified. The only requirement for column order is that the column names are in the same order as the data in each line.

Column names cannot contain commas, are case sensitive, and cannot start or end with a space (such spaces are automatically trimmed when the file is parsed). Column names must be unique. Duplicate column names (after space trimming) result in an error.

The column header must be the second non-blank line in the file.

### Data Lines

```
<columnvalue1>,<columnvalue2>,<columnvalue3>...,<columnvalueN>
```

Where <columnvalue1> through <columnvalueN> are the values of <columnname1> through <columnnameN> for one row of data.

Each line of data corresponds to one data item within the file, with one value for each column, in the same order as the column header. The values are separated by commas. No value can contain a comma. Data values are trimmed of leading and trailing white space when parsed.

The third and subsequent non-blank lines in the file must be data lines. It is possible for a file to contain no data lines. This means there were no data items for that period.

### Required Columns

The following table describes the columns that are required to be in the AgentProductivity.AGENT file.

Column	Data Type	Description
acdAgentId	string	The ID of the agent in the ACD.
acdServiceId	positive long	The numerical ID of the service queue in the ACD.
contacts Handled	nonnegative double	The number of contacts for the service queue handled by the agent during the period.
totalTalkSeconds	nonnegative double	The total talk time on contacts for the service queue handled by the agent during the period. A contact's talk time can start in the previous period; the entire talk time is counted.
totalHold Seconds	nonnegative double	The total hold time on contacts for the service queue handled by the agent during the period. A contact's talk time can start in the previous period; the entire talk time is counted.
totalAfterContact WorkSeconds	nonnegative double	The total after contact work time on contacts for the service queue handled by the agent during the period. A contact's talk time can start in the previous period; the entire talk time is counted.

Column	Data Type	Description
totalPeriodHandle TimeSeconds	nonnegative double	The total time the agent spent handling contacts (talk, hold, work) for the service queue that occurred within the boundaries of the period.  This metric (A) is differentiated from the sum of totalTalkSeconds + totalHoldSeconds + totalAfterContactWorkSeconds (B) in that it does not include time that exceeds the period boundary. For example, if a call is answered by the agent at 09:29 and the call ends at 9:33 with no after-contact work, and if the ACD considers the call to be handled during the 09:30-10:00 period, then the call contributes 4 minutes to B (09:29-09:33) but only 3 minutes to A (09:30-09:33).
totalUnprorated ReadyWaiting Seconds	nonnegative double	The total Ready/Waiting time for the agent during the period, across all service queues.
totalUnprorated NotReadyBusy Seconds	nonnegative double	The total Not Ready/Busy time for the agent during the period, across all service queues (not counting time while the agent is handling a contact or reported as Ready/Waiting on another service queue).
totalUnprorated InSession Seconds	nonnegative double	The total in session time (logged in time) for the agent during the period, across all service queues.

### Optional Columns

The following table describes the columns that are optional in the AgentProductivity.AGENT file. If not included, the default value is used for the metric.

Column	Data Type	Description
periodStart	timestamp	A timestamp representing the start of the interval period.  If included, this field must refer to the same point in time (although not necessarily expressed in the same time zone) as the period start time in the file name and in the file header. If it does not match, an error occurs.
contacts TransferredOut	nonnegative double	Default value = 0. The number of contacts transferred out by the agent during the period.

---

Column	Data Type	Description
contacts TransferredIn	nonnegative double	The number of contacts transferred in to the agent during the period.  This field is reserved for future use. Currently the value is ignored.
contactsExternalIn	nonnegative double	Default value = 0. The number of external inbound contacts to that agent during the period.
contactsExternalOut	nonnegative double	Default value = 0. The number of external outbound contacts from the agent during the period.
totalReserved Seconds	nonnegative double	Default = 0. The total reserved time for contacts for the service queue for the agent during the period. If non-zero, the reserved time is not counted as Ready/Waiting time.

Column	Data Type	Description
totalProrated WaitingSeconds	nonnegative double	<p>The total prorated Ready/Waiting time for the agent and service queue.</p> <p>When totaled across all service queues, this must be equal to the required field totalUnproratedReadyWaitingSeconds.</p> <p>If the prorated values are omitted or set to zero, the Capture service divides the unprorated values equally among the service queues for which the agent has agent-service queue productivity data during the period.</p> <p>If the prorated values are specified, and add up to the unprorated values, then the prorated values are used for the agent-service queue combination.</p> <p>If the prorated values are specified and do not add up to the unprorated values, then for each agent-service queue entry, the Capture service recalculates the prorated value as the unprorated value weighted by the specified prorated value divided by the sum of the specified prorated values for all of the service queues for which the agent has agent-service queue data in the period.</p> <p>This is most useful for ACDs that assign Ready/Waiting time to an agent for one service queue and Not Ready/Busy time for other service queues. It is important that unprorated and prorated Not Ready/Busy times do not include times when the agent is accruing Ready/Waiting or other time for other service queues.</p> <p>For ACDs that report Not Ready/Busy time for service queues when the agent is handling calls or Ready/Waiting time in another service queue, it is important to remove all of the duplicate time and report Not Ready/Busy time only when an agent is Not Ready/Busy in all service queues simultaneously. For this type of ACD, the total unprorated Ready/Waiting time is the sum of each service queue's Ready/Waiting time and the total unprorated Not Ready/Busy time is computed from the total in-session (logged-in) time less the total in-</p>



Column	Data Type	Description
		service (Talk, Hold, After Work, Ready/Waiting, and Reserved) time.
totalProrated NotReady BusySeconds	nonnegative double	<p>The total prorated Not Ready/Busy time for the agent and service queue.</p> <p>When totaled across all service queues, this is equal to the required field totalUnproratedNotReadyBusySeconds.</p> <p>For a detailed explanation of this value, see the description for the field totalProratedReadyWaitingSeconds.</p>
contactsAnswered	nonnegative double	<p>The number of contacts for the service queue answered by the agent during the period.</p> <p>The difference between this field and the required field contactsHandled is up to the implementer. In WFM, contactsHandled is used for things such as forecasting numbers of contacts and computing average handle times, while contactsAnswered is used for computing average speed of answer. Some ACDs make a distinction between the two fields; for example, in some ACDs a handled contact is a contact that ends during the reporting interval, while an answered contact is a contact that is answered during the reporting interval.</p> <p>This field is reserved for future use. Currently the value is ignored.</p>
totalAnswer DelaySeconds	nonnegative double	<p>The total delay in answering contacts for the service queue for contacts the agent answered during the period.</p> <p>This field is reserved for future use. Currently the value is ignored.</p>

### Example

File name: 201301151330N0600\_0\_AgentProductivity.AGENT

AGENT DATE: 2013-01-15 INTERVAL: 13:30 TZOFFSET: -0600 ACD: 0acdA-  
gentId,acdSer-

viceId,contactsHandled,contactsAnswered,contactsTransferredOut,contactsExternal

```

5009,5236,7,8,0,0,0,933.728,0.000,228.718,1215.857,12.0
102.460,1800.000
5009,5240,4,4,0,0,0,365.039,0.000,104.644,469.683,12.000,
102.460,1800.000
5073,5236,1,1,0,0,0,380.596,0.000,9.510,215.077,1.000,
1583.923,1800.000
5073,5240,0,0,0,0,0,0.000,0.000,0.000,0.000,1.000,1583.923,
1800.000

```

## ServiceHistorical.SERVICE File

The ServiceHistorical.SERVICE file contains service queue data by 30-minute intervals from 00:00 to 23:59 for both interactive and non-interactive service queue types.

- Interactive service queue types consist of contacts in which agents and customers have real-time communication, such as voice (call) or chat.
- Non-interactive service queue types consist of contacts in which agents and customers do not have real-time communication, such as email, fax, and social media, as well as contact activities like stuffing envelopes.

### File Name Format

<date/time><tzoffset>\_<ACD#>\_ServiceHistorical.SERVICE

Element	Description
<date/time>	The date and time the file was generated, in YYYYMMDDHHMM format. Example: 201301151430 (14:30 on January 15, 2013). Time is in 24-hour format.
<tzoffset>	The time zone (where the ACD is located) date/time offset from GMT in AHHMM format, where: <ul style="list-style-type: none"> <li>• A is either P (positive) or N (negative), characterizing the offset from GMT</li> <li>• HH is the number of offset hours</li> <li>• MM is the number of offset minutes</li> </ul> Example: If the ACD is in GMT, then the <tzoffset> is P0000. If the ACD is in EST (GMT-0500), then the <tzoffset> is N0500.
<ACD#>	The ACD number. Can be zero (0) or a positive number with as many digits as desired.

### File Header

SERVICE DATE: <date> INTERVAL: <time> TZOFFSET: <tzoffset> ACD: <ACD#>

Element	Description
<date>	The date the file was generated, in YYYY-MM-DD format.
<time>	The beginning of the 30-minute interval covered by the data, in HH:MM 24-hour format.
<tzoffset>	The time zone (where the ACD is located) offset from GMT, in +HHMM, -HHMM, or HHMM format. If plus or minus is not specified, the offset is assumed to be plus (positive).
<ACD#>	The number of the ACD that is the source of the data. Currently this value is not used, but there must be a number. Use 0 as a default value if no other number is valid. It is recommended that it be the same as the ACD# used in the file name.

The file header must be the first non-blank line in the file.

**Note:** The SERVICE DATE, INTERVAL, and TZOFFSET values must represent a period start time that matches those in the file name as well as a historical period supported by WFM. It is not required that the individual values match, but only to mean the same point in time. For example, if the file is named:

```
201301231800N0500_0_ServiceHistorical.SERVICE
```

and the header is:

```
SERVICE DATE: 2013-01-23 INTERVAL: 23:00 TZOFFSET: 0000 ACD: 0
```

there is no error, because each is expressing the same point in time in different time zones. The time 18:00 in the time zone N0500 (a 5-hour negative offset from GMT) is the same as the time 23:00 in the GMT time zone.

## Column Header

```
<columnname1>,<columnname2>,<columnname3>...,<columnnameN>
```

Where <columnname1> through <columnnameN> are the names of columns represented in the file.

Column names are not restricted to required and optional columns. You can add additional unrecognized columns and associated data (for example, agent names or service names) to make the GIS files more human-readable. The GIS API ignores these columns and their associated data.

Column order is not specified. The only requirement for column order is that the column names are in the same order as the data in each line.

Column names cannot contain commas, are case sensitive, and cannot start or end with a space (such spaces are automatically trimmed when the file is parsed). Column names must be unique. Duplicate column names (after space trimming) result in an error.

The column header must be the second non-blank line in the file.

### Data Lines

<columnvalue1>,<columnvalue2>,<columnvalue3>...,<columnvalueN>

Where <columnvalue1> through <columnvalueN> are the values of <columnname1> through <columnnameN> for one row of data.

Each line of data corresponds to one data item within the file, with one value for each column, in the same order as the column header. The values are separated by commas. No value can contain a comma. Data values are trimmed of leading and trailing white space when parsed.

The third and subsequent non-blank lines in the file must be data lines. It is possible for a file to contain no data lines. This means there were no data items for that period.

### Required Columns

The following table describes the columns that are required to be in the ServiceHistorical.SERVICE file for both interactive and non-interactive service queue types.

The column names can be interpreted differently for each service queue type. For example, contactsHandled can be thought of as the number of chat contacts handled or the number of email contacts handled.

For non-interactive service queues, if columns such as contactsAbandoned, totalHoldSeconds, and totalAnswerDelaySeconds do not apply, then set the values in these columns to 0 (zero).

Column	Data Type	Description
acdServiceId	positive long	The numerical ID of the service queue in the ACD.
contacts Offered	nonnegative double	The number of contacts for the service queue offered to agents during the period.
contacts Handled	nonnegative double	The number of contacts for the service queue handled by agents during the period.

Column	Data Type	Description
contacts Answered	nonnegative double	The number of contacts for the service queue answered by the agent during the period.  The difference between this field and the required field contactsHandled is up to the implementer. In WFM, contactsHandled is used for things such as forecasting numbers of contacts and computing average handle times, while contactsAnswered is used for computing average speed of answer. Some ACDs make a distinction between the two fields; for example, in some ACDs a handled contact is a contact that ends during the reporting interval, while an answered contact is a contact that is answered during the reporting interval.
contacts Abandoned	nonnegative double	The number of contacts for the service queue abandoned by callers during the period.
totalTalk Seconds	nonnegative double	The total talk time on contacts for the service queue handled by agents during the period.
totalHold Seconds	nonnegative double	The total hold time on contacts for the service queue handled by agents during the period.
totalAfter ContactWork Seconds	nonnegative double	The total after contact work time on contacts for the service queue handled by agents during the period.
totalAnswer Delay Seconds	nonnegative double	The total delay in answering contacts for the service for contacts the agent answered during the period
serviceLevel Percent	nonnegative double £ 100	The total delay in answering contacts for the service for contacts the agent answered during the period.

### Optional Columns

The following table describes the columns that are optional in ServiceHistorical.SERVICE file. If not included, the default value is used for the metric.

Columns	Data Type	Description
period Start	timestamp	A timestamp representing the start of the interval period.  If included, this field must refer to the same point in time (although not necessarily expressed in the same time zone) as the period start time in the file name and in the file header. If it does not match, an error occurs.
contacts Blocked	nonnegative double	Default value = 0. The number of contacts blocked for the service queue during the period.
service Level Seconds	positive double	The service level threshold in seconds for the service queue during the period.  This field is reserved for future use. Currently the value is ignored.
contactsIn Queue	nonnegative double	The maximum number of contacts in queue for the interval. Used only with non-interactive queues. For interactive queues, this value will always be zero no matter what value is entered in this column.

### Missing Columns

Some columns, such as `totalInServiceSeconds` and `totalInSessionSeconds`, might appear to be missing. In fact, these are values that need to be prorated; they are computed by combining other historical service queue data with agent productivity data. This is done so that per-service queue values from historical service queue data add up to the same totals as the per-agent-per-service queue values from agent productivity data.

For example, in most ACDs, time in a Ready/Waiting state is not associated with a service queue. Rather, the agent is just in that state. In the `AgentProductivity.AGENT` file, this total is provided in each agent-service queue entry for the agent as `totalUnproratedReadyWaitingTime`. From that, we compute a `proratedReadyWaitingTime` for each service queue (the total unprorated value divided by the number of service queues for which the agent has agent-service queue productivity data during the period).

A similar approach is taken for `totalUnproratedNotReadyBusySeconds`, which is converted to `proratedNotReadyBusySeconds`. We use the prorated values to compute totals for a service queue. As a result, we don't count the same time twice (or more for multiple service queues) and the metrics are zero-sum.

If the `totalProratedReadyWaitingTime` and/or `totalProratedNotReadyBusyTime` values are specified, we use these as a guide to prorating the unprorated times. If they add up correctly, the specified prorated values are used. If not, we use them as a weight (using the specified prorated value divided by the sum of the specified prorated values across all the service queues for the agent during the interval) for computing the prorated values.

Implementers who want to have more control over how time is prorated, other than dividing by the number of service queues, can add the columns `totalProratedReadyWaitingSeconds` and `totalProratedNotReadyBusySeconds` to the `AgentProductivity.AGENT` file.

It is strongly recommended that you ensure that the sum of the `totalProratedReadyWaitingSeconds` values for each agent-service queue line for a given agent during a period adds up to the `totalUnproratedReadyWaitingSeconds` value in each agent-service queue line for that agent, and likewise for Not Ready/Busy time. It is also strongly recommended that you ensure that Not Ready/Busy time is never counted for a service queue if that time is concurrent with time in another state for other service queues.

For example, for ACDs that report Not Ready/Busy time for one service queue while an agent is handling calls in another service queue, the Not Ready/Busy time concurrent with call handling should not be included in the specified unprorated or prorated values. Likewise, if an ACD counts Ready time for one service queue while also counting Not Ready/Busy time for another service queue, that Not Ready/Busy time should not be included in the specified unprorated or prorated values. Only the total unprorated values `totalUnproratedReadyWaitingTime` and `totalUnproratedNotReadyBusyTime` should be double counted, and only as time an agent is Ready/Waiting for some service queue, and time an agent is Not Ready/Busy for all service queues, respectively.

### Example

File name: `201301151330N0600_0_ServiceHistorical.SERVICE`

```
SERVICE DATE: 2013-01-15 INTERVAL: 13:30 TZOFFSET: -0600 ACD: 0
acdSer-
viceId, -
contactsOffered, contactsHandled, contactsAnswered, contactsAbandoned, totalTalkSe
5236, 7, 8, 9, 0, 1314.324, 0.000, 238.228, 80929.903, 0.000
5240, 4, 4, 4, 0, 365.039, 0.000, 104.644, 35600.210, 0.000
```

### AgentState.EVENT File

The `AgentState.EVENT` file contains agent state data by 30-minute intervals from 00:00 to 23:59.

#### File Name Format

`<date/time>_<ACD#>_AgentState.EVENT`

Element	Description
<code>&lt;date/time&gt;</code>	The date and time the file was generated, in YYYYMMDDHHMM format.  Example: 201301151430 (14:30 on January 15, 2013). Time is in 24-hour format.
<code>&lt;ACD#&gt;</code>	The ACD number. Can be zero (0) or a positive number with as many digits as desired.

## File Header

EVENT DATE: <date> ACD: <ACD#>

<date> The date the file was generated, in YYYY-MM-DD format.

<ACD#> The number of the ACD that is the source of the data. Currently this value is not used, but there must be a number. Use 0 as a default value if no other number is valid. It is recommended that it be the same as the ACD# used in the file name.

The file header must be the first non-blank line in the file.

**Note:** The EVENT DATE values must represent the server date that matches both that in the file name and the date of all agent events in the file. For example, if a file is named:

20130123\_0\_AgentState.EVENT

then the file must contain only events for January 23, 2013 and no other date in the server time zone. This does not mean that the individual events must be expressed in the server time zone, only that after the events are converted to the server time zone by the Capture service, the date of the converted event timestamp must match the date in the file name and the file header.

## Column Header

<columnname1>,<columnname2>,<columnname3>...,<columnnameN>

Where <columnname1> through <columnnameN> are the names of columns represented in the file.

Column names are not restricted to required and optional columns. You can add additional unrecognized columns and associated data (for example, agent names or service names) to make the GIS files more human-readable. The GIS API ignores these columns and their associated data.

Column order is not specified. The only requirement for column order is that the column names are in the same order as the data in each line.

Column names cannot contain commas, are case sensitive, and cannot start or end with a space (such spaces are automatically trimmed when the file is parsed). Column names must be unique. Duplicate column names (after space trimming) result in an error.

The column header must be the second non-blank line in the file.

## Data Lines

<columnvalue1>,<columnvalue2>,<columnvalue3>...,<columnvalueN>

Where <columnvalue1> through <columnvalueN> are the values of <columnname1> through <columnnameN> for one row of data.



Each line of data corresponds to one data item within the file, with one value for each column, in the same order as the column header. The values are separated by commas. No value can contain a comma. Data values are trimmed of leading and trailing white space when parsed.

The third and subsequent non-blank lines in the file must be data lines. It is possible for a file to contain no data lines. This means there were no data items for that period.

### Required Columns

The following table describes the columns that are required to be in the AgentState.EVENT file.

Column	Data Type	Description
acdAgentId	string	The ID of the agent in the ACD.
eventDateTime	timestamp	A timestamp representing the point in time at which the agent state event took place, consisting of the following data types:  <date> <time> <tzoffset>  If the tzoffset is omitted, the date and time are assumed to be in GMT time.
agentState	string, long, or null	A code representing the agent state. See Agent State Values for more information.

### Optional Columns

The following table describes the columns that are optional in the AgentState.EVENT file. If not included, the default value is used for the metric.

Column	Data Type	Description
reasonCode	long or null	Default value = NULL. The reason code associated with the agent state change.

### Agent State Values

The agentState field accepts only certain values. If an event does not match one of these agent states, no line for that event should be written to the file. Each state has an associated integer or string code that can be used as a value. The following table describes the agent states and their associated codes.

Agent State	Code (Integer)	Code (String)	Description
Logging Out/ Out of Service	1	LO	The agent logs out of the ACD

Agent State	Code (Integer)	Code (String)	Description
Ready/Waiting	2	RE	The agent is ready to handle a contact
Talking/In Contact	3	TK	The agent is talking to a contact
Work After Contact	4	WK	The agent is performing after-contact work
On Hold	5	OH	The agent is on hold
Not Ready/Busy	6	NR	The agent is not ready to receive contacts

Only one event should be written per agent per timestamp. If multiple events are written, it is likely (but not guaranteed) that only the last event in the file for the agent for the same timestamp will be captured.

For ACDs that report an agent as Ready/Waiting or Not Ready/Busy for specific service queues, and use one of these values in one service queue while the agent is in another state on another service queue, the events must be converted to be non-service queue specific. For example, if an ACD has an agent as Talking in one service queue and Not Ready/Busy in other service queues, a single TK or 3 agentState should be written.

In general, when an agent is in multiple states for different service queues, and only one event per agent per timestamp is written, priorities are established as follows:

- Every other event takes priority over Logged Out
- Every event except Logged Out takes priority over Not Ready/Busy
- Every event except Logged Out and Not Ready/Busy takes priority over Ready/Waiting

### Example

File name: 20130115\_0\_AgentState.EVENT

```
EVENT DATE: 2013-01-15 ACD: 0
acdAgentId,eventDateTime,agentState,reasonCode
```

[lines omitted for brevity]

```
5009,2013-01-15 09:00:08.977,6,3
5073,2013-01-15 09:00:08.977,3,NULL
5073,2013-01-15 09:00:25.983,4,NULL
5073,2013-01-15 09:01:13.367,2,NULL
5073,2013-01-15 09:01:14.367,3,NULL
```

[lines omitted for brevity]

```
5009,2013-01-15 16:51:24.247,3,NULL
5009,2013-01-15 16:55:43.720,5,NULL
5009,2013-01-15 16:56:04.553,3,NULL
5009,2013-01-15 17:01:51.483,4,NULL
5009,2013-01-15 17:02:02.313,1,NULL
5073,2013-01-15 17:04:55.210,4,NULL
5073,2013-01-15 17:05:20.987,1,NULL
```

## Integrating Your HRMS with WFM

If HRMS (Human Resources Management System) integration is enabled in WFM Configuration Setup, WFM will import vacation data from your HRMS and export data on vacation hours used to a file for use by your HRMS.

**IMPORTANT:** The agent's username as configured in WFM is the identifier for vacation hours data. It must be identical in WFM and the HRMS for the data to be written to the correct record in each.

WFM monitors the following folder for the text file sent from the HRMS:

```
C:\Program Files\Common Files\WFM\vacationreports
```

As soon as the file appears, data from the file is imported into WFM using the GIS API. The imported data overwrites any existing vacation data in the WFM database.

Data is exported from WFM daily. It is up to the customer to determine the frequency at which data will be imported from the HRMS to WFM. Best practice is to import the data at the interval at which earned vacation hours are calculated in the HRMS.

At the time of day configured in WFM Configuration Setup (HRMS Integration step), WFM outputs a report file that contains the number of hours used by agents for each day for the last seven days. This file also contains pending and approved hours as of the current date. Hours are considered used for schedules ending the previous day. Any schedules in progress are not counted as used.

The import files are archived for future reference. When the text files are processed by the GIS API, they are automatically moved to the archives folder in a subfolder named for the date on which they were processed.

For example:

```
...\WFM\vacationreports\archives\2013-05-02\VacationHours_From_
HRMS
```

```

... \WFM\vacationreports\archives\2013-05-03
... \WFM\vacationreports\archives\2013-05-04

```

Vacation import files are deleted according to the schedule configured or vacation report file on the WFM Configuration Setup utility's Data Retention Periods step.

Vacation export files are not automatically deleted. It is up to the customer to determine when and if the export files are to be deleted.

## Import File Details

The following table displays the details of the import file.

Element	Description
File Location	C:\Program Files (x86)\Common Files\WFM\vacationreports
File Name	VacationHours_From_HRMS
File Header	VACATION FROM HRMS DATE: <date>
Column Headers	<p>login (required)</p> <p>vacationTypeLabel (required)</p> <p>availableHours (required)</p> <p>totalEarnedHours (optional)</p> <div style="background-color: #e6f2e6; padding: 5px; margin-top: 10px;"> <p><b>Note:</b> If the optional totalEarnedHours field is not included in the file, WFM defaults its values to zero and overwrites the existing values. This can result in a negative value when calculating the hours used for an agent. If the field is included in the file, its value cannot exceed 9999.</p> </div> <div style="background-color: #e6f2e6; padding: 5px; margin-top: 10px;"> <p><b>Note:</b> The data value entered for vacationTypeLabel is all rows must match exactly the vacation type name as configured in WFM (Application Management &gt; Vacation Planning &gt; Vacation Types).</p> </div>

The following is an example of an import file.

```

VACATION FROM HRMS DATE: 2016-04-15
login,vacationTypeLabel,availableHours,totalEarnedHours
smithj,Days Off,8,32

```

```
jonest,Vacation,16,24
smithj,Vacation,24,24
```

**Note:** Data elements are limited to 4 digits. Errors in records are logged in the log file.

**Note:** Commas that are part of the HRMS import data must be escaped with a backslash (for example, "3\,000") in order to be treated as literals and not file delimiters.

## Export File Details

The following table displays the details of the import file.

Element	Description
File Location	C:\Program Files\Common Files\WFM\vacationreports
File Name	vacation_<date>_WFM
File Header	VACATION FROM WFM DATE: <date>
Column Headers	login date vacationTypeLabel usedHours requestedHours approvedHours employeeId acclId

The output export file contains the following for each vacation type:

- Used Hours: The hours used for the date on the row.
- Requested Hours: The total hours that are pending and not yet approved at the current time. This is displayed only for the current day; previous days show a dash (-).
- Approved Hours: The total approved hours at the current time. This is displayed only for the current day; previous days show a dash (-).

The following is an example of an export file.

```
VACATION FROM WFM DATE: 2016-04-20
```

```
login,d-  
ate,-  
vacationTypeLabel,usedHours,requestedHours,approvedHours,employeeId,acdId  
smithj,2016-04-17,Days Off,-,-,8.0,john.smith,2  
smithj,2016-04-17,Chris!,-,-,-,john.smith,2  
jonest,2016-04-19,Vacation,8.0,-,8.0,teri.jones,1  
jonest,2016-04-19,Personal Days,8.0,-,-23.98,teri.jones,1  
jonest,2016-04-19,Chris!,8.0,-,-,teri.jones,1
```

## Real-Time Data Capture API

The real-time data capture API can be used to notify WFM of real-time agent state information for any ACD. This real-time data is then used by WFM to calculate agent adherence and conformance.

The real-time API is an HTTP REST API.

**Note:** This API is a replacement for a similar API that existed prior to WFM 9.0. If you are using the older API, you must update your application to use this API instead.

**Note:** HTTPS can be used with the API as well as HTTP.

### API Definition

Applications using this API must do the following:

1. On application startup, establish an HTTP session with the WFM Jetty service.
2. For the life of the application:
  - a. Monitor agent state changes using an ACD-specific integration method.
  - b. Submit agent state changes to the WFM server.
3. On application shutdown, destroy the HTTP session.

### Establish Session

Before providing agent state data to WFM, you must first establish an HTTP session (log in).

To log in, issue an HTTP POST request to:

```
http://<Calabrio ONE container host>/api/rest/authorize
```

The Calabrio ONE container host is set during installation. See the *Workforce Management Installation Guide* for more information.

The body of the request must be in the following format:

```
[
  {
    "id": "scheduling",
    "data": {"wfm.service": true},
    "userId": "serviceUser",
    "password": "s3rv1c3P@ssw0rd!",
    "locale": "en"
  }
]
```

A response of **200 OK** indicates success.

**Note:** The serviceUser password is not user configurable. Contact Calabrio Support Services if you need to change it.

### Submit Agent State Information

You can submit each agent state change as a separate API request, or bundle several agent state changes into a single request. In general, sending multiple agent state changes in a single request results in higher performance.

To submit agent state information, issue an HTTP POST request to:

```
http://<Calabrio ONE container host>/api/rest/scheduling/
gis/realTimeAgentState
```

The body of the request must be in the following format:

```
[
  {
    "acdAgentId": string,
    "gisStateIdentifier": string,
    "timestamp": number,
    "reasonCode": number
  }, ...
]
```

The fields in this request are defined in the following table.

Field	Descriptions
acdAgentId	The ACD identifier for the agent. This is the agent's ACD ID in WFM (Application Management > People > Agents).
gisStateIdentifier	<p>The identifier for the agent state.</p> <ul style="list-style-type: none"> <li>• 1–Logout. The agent has logged out of the ACD. In many ACDs, a numeric reason code might accompany this state to indicate the reason for the state change.</li> <li>• 2–Not Ready. The agent is not accepting contacts from the ACD. In many ACDs, a numeric reason code might accompany this state to indicate the reason for the state change.</li> <li>• 3–Ready. The agent is ready to accept contacts from the ACD.</li> <li>• 4–Talking. The agent is on a contact.</li> <li>• 5–Work Not Ready. The agent is engaged in after contact work and will enter a Not Ready state when finished.</li> <li>• 6–Work Ready. The agent is engaged in after contact work and will enter a Ready state when finished.</li> <li>• 10–Hold. The agent has placed the contact on hold.</li> </ul> <div style="border: 1px solid black; background-color: #e6f2e6; padding: 5px; margin-top: 10px;"> <p><b>Note:</b> If the ACD has an after contact work state, but makes no distinction between Work Not Ready and Work Ready, then either state can be used.</p> </div>
timestamp	The time that the agent state change took place, expressed in milliseconds since epoch.
reasonCode	An optional reason code. This can be null or blank. Valid values are 1-65535.

### Examples

The following is an example of an agent state change sent as a single API request.

- Agent 5009 goes into the hold state at Wed, 27 Jun 2012 18:03:27 GMT.

```
[
  {
```



```
        "acdAgentId":"5009",
        "gisStateIdentifier":"10",
        "timestamp":1340820207000,
        "reasonCode":null
    }
]
```

The following is an example of multiple agent state changes bundled together into a single request.

- Agent 5009 goes Not Ready with reason code 9 at Wed, 27 Jun 2012 18:03:39 GMT.
- Agent 5009 goes Ready at Wed, 27 Jun 2012 18:03:40 GMT.
- Agent 5002 goes Not Ready with reason code 2 at Wed, 27 Jun 2012 18:03:39 GMT

```
[
  {
    "acdAgentId":"5009",
    "gisStateIdentifier":"2",
    "timestamp":1340820219000,
    "reasonCode":9
  },
  {
    "acdAgentId":"5009",
    "gisStateIdentifier":"3",
    "timestamp":1340820220000,
    "reasonCode":null
  },
  {
    "acdAgentId":"5002",
    "gisStateIdentifier":"2",
    "timestamp":1340820219000,
```

```
    "reasonCode":2  
  }
```

### **Destroy Session**

On application shutdown, you should destroy the HTTP session (log out).

To log out, issue an HTTP DELETE request to:

```
http://<Calabrio ONE container host>/api/rest/authorize
```

A response of **200 OK** indicates success.

# WFM Historical Import Template (WHIT)

---

WHIT is not included with your WFM installation. You must request the components from Customer Support Services and install them according to the instructions in this section.

**Note:** File paths shown here are the default file paths. WFM can be installed to a custom location, so your file path might differ.

## Software Requirements

The software requirements for running the WHIT tool are as follows:

- WFM 11.0 or newer must be installed and running.
- Java 7 or later must be installed on the WFM server that hosts the WFM Capture service.

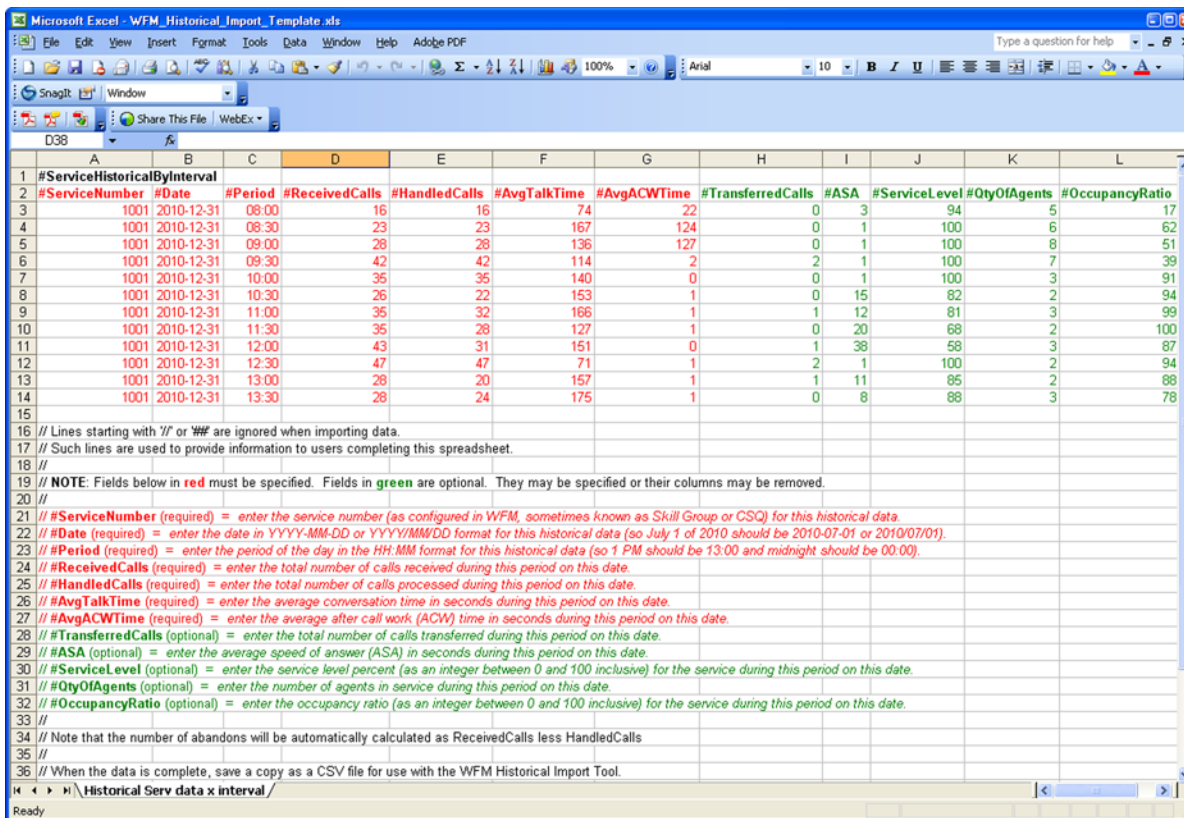
## WHIT Components

WHIT consists of two components:

- WFM Historical Import Template spreadsheet
- WHIT executable

## WFM Historical Import Template

The WFM Historical Import Template is an Excel spreadsheet named `WFM_Historical_Import_Template.xls`. Your historical data is entered in this spreadsheet and then saved in CSV (comma separated value) format.



## WHIT Executable

The WHIT executable (WHIT.exe) converts your historical data from the CSV file you create from the Historical Import Template to a set of files in the format expected by the Generic Interface Services (GIS) API, a feature that comes with WFM. The GIS API, in turn, imports the files into the WFM database.

## Importing Data with WHIT

The process of importing data consists of the following tasks:

1. Install WHIT.
2. Insert your data into the WHIT spreadsheet and save it in CSV format.
3. Run WHIT.

## Installing WHIT

To install WHIT on your WFM server:

1. Log in to the WFM server that hosts the WFM Capture service.
2. Copy the WHIT executable (WHIT.exe) and the WFM\_Historical\_Import\_Template.xls to any folder on the WFM server.

## Preparing the WHIT Spreadsheet

The fields that you complete in the WFM Historical Import Template spreadsheet depend on the type of service queue for which you are importing data.

There are two service queue types:

- **Interactive:** Interactive service queue types consist of contacts in which agents and customers have real-time communication, such as voice (call) or chat.
- **Non-Interactive:** Non-interactive service queue types consist of contacts in which agents and customers do not have real-time communication, such as email, fax, and social media, as well as contact activities like stuffing envelopes.

The field names can be interpreted differently for each service queue type. For example, ReceivedCalls can be thought of as the number of chat contacts received or the number of email contacts received. Complete the spreadsheet in a manner that works best for your contact center.

### *To prepare the WHIT spreadsheet:*

1. Open the WFM\_Historical\_Import\_Template.xls spreadsheet.
2. Insert your historical data into the spreadsheet. A description of what goes in each field is included in the spreadsheet.

To import historical data for interactive service queues, complete the required fields in red. You can also complete optional fields in green if desired.

**Note:** The #Period column data must be 30-minute intervals.

To import historical data for non-interactive service queues, complete the required fields in red. It is recommended that you also complete the ServiceLevel, QtyOfAgents, and OccupancyRatio fields.

Set the values in fields to 0 (zero) for data you do not want to specify.

**Note:** If those fields are optional, you can remove those columns.

3. When you have added all your historical data, save the spreadsheet in comma separated value (CSV) format.
4. Copy the CSV file to the same location on the WFM Capture service server where you copied the WHIT executable.

## Running WHIT

Perform this task on the WFM server that hosts the WFM Capture service.

### *To run WHIT:*

1. Double-click WHIT.exe and follow the instructions in the application.
2. Click Import. WHIT generates GIS files and places them in the following location:

C:\Program Files (x86)\Common Files\WFM\reports

**Note:** This is the default location. Your instance of WFM might be installed on a different drive.

The GIS API then processes the GIS files and imports the data into the WFM database.

3. When the GIS files are processed, the GIS API removes the files from the reports folder and WHIT displays the following message:

Capture status: All files are captured. Historical import is complete.