



Cisco Unified Workforce Optimization

Workforce Management GIS API Reference Guide Release 10.5

First Published: June 18, 2014

Last Updated: March 18, 2015

Americas Headquarters

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
<http://www.cisco.com>
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

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 NONINFRINGEMENT 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. 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.

Workforce Management GIS API Reference Guide

© 2014, 2015 Cisco Systems, Inc. All rights reserved.

Contents

- Introduction** **5**
 - Other Data Import Tools 5
- Historical Data Capture** **7**
 - Requirements 7
 - Text File Locations 8
 - ACD Numbering 8
 - Text File Details 9
 - Data Type Definitions 9
 - AgentProductivity.AGENT File 11
 - File Name Format 11
 - File Header 12
 - Column Header 13
 - Data Lines 13
 - Required Columns 13
 - Optional Columns 15
 - Example 19
 - ServiceHistorical.SERVICE File 20
 - File Name Format 20
 - File Header 21
 - Column Header 22
 - Data Lines 22
 - Required Columns 23
 - Optional Columns 25
 - Missing Columns 25
 - Example 26

AgentState.EVENT File	27
File Name Format	27
File Header	27
Column Header	28
Data Lines	28
Required Columns	29
Optional Columns	29
Agent State Values	29
Example	30
Real-Time Data Capture API	33
API Definition	33
Establish Session	33
Submit Agent State Information	34
Examples	36
Destroy Session	37

Introduction

The Generic Interface Services (GIS) API is used to integrate historical and real-time data from any automatic call distributor (ACD) to the Workforce Management (WFM) database at specified intervals. WFM uses this data to generate reports and statistics.

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

Other Data Import Tools

There is another tool available that can be used to import data. The Workforce Management Historical Import Tool (WHIT) enables you to convert existing historical data for use by a newly installed WFM system.

For more information about WHIT, see the *Calabrio Workforce Management Historical Import Tool Reference Guide*.

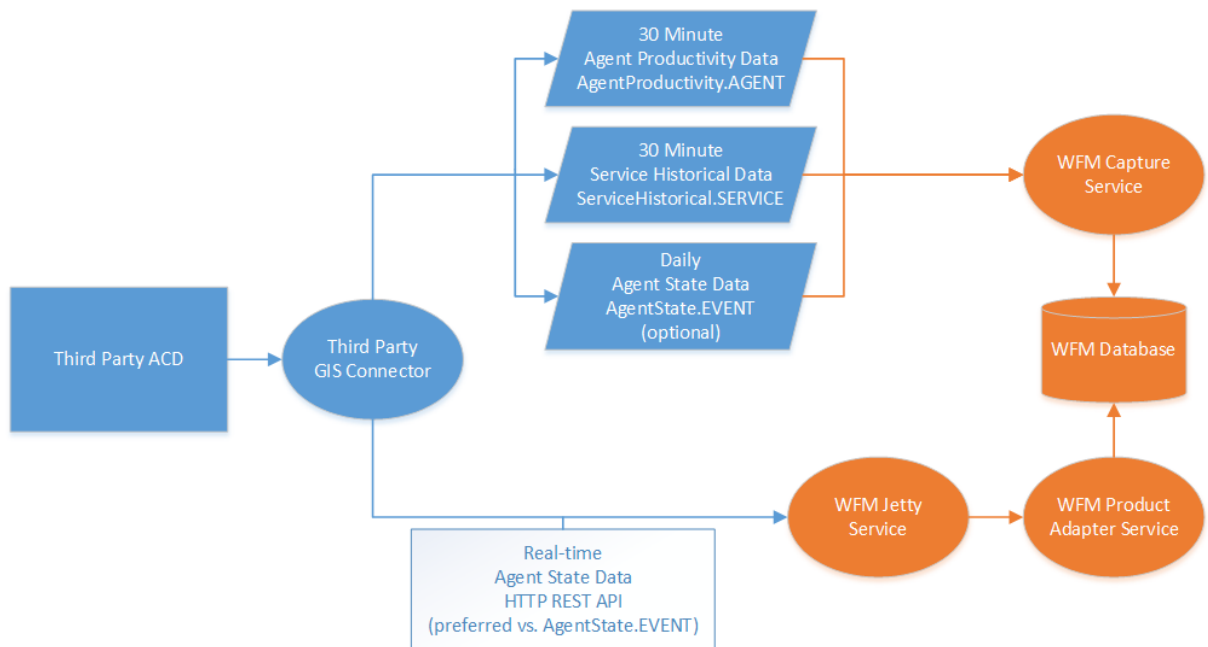
Historical Data Capture

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 data capture.

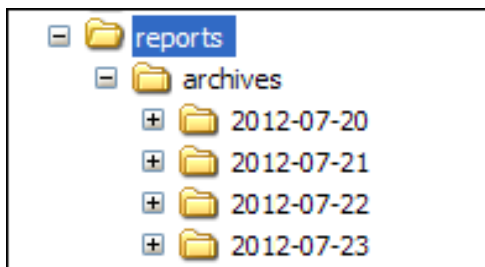
- Service historical data is calculated from the summation of agent service productivity data. As a result, both the AgentProductivity.AGENT and ServiceHistorical.SERVICE files must be available for a given period for capture to take place. If both files are found, but data is missing from one, certain 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.

Text File Locations

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

```
...\WFO_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 (see Figure 3). 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 (Log ID in WFM) and service queue identifiers (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.

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.

Data Type	Definition
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.
time	<p>A time expressed in any of the following formats:</p> <ul style="list-style-type: none"> ■ hh:mm ■ hh:mm:ss ■ hh:mm:ss.sss

Data Type	Definition
timestamp	<p>A time stamp consisting of the following data types:</p> <p><date> <time> <tzoffset></p> <p>If the tzoffset is omitted, the date and time are assumed to be in the default time zone of the WFM server.</p>
tzoffset	<p>A time zone offset expressed in +HHMM, -HHMM, HHMM format or the literal values GMT or UTC.</p>

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

<date/time>	<p>The date and time the file was generated, in YYYYMMDDHHMM format.</p> <p>Example: 201301151430 (14:30 on January 15, 2013). Time is in 24-hour format.</p>
<tzoffset>	<p>The time zone (where the ACD is located) date/time offset from GMT in AHHMM format, where:</p> <ul style="list-style-type: none"> ■ A is either P (positive) or N (negative), characterizing the offset from GMT ■ HH is the number of offset hours ■ MM is the number of offset minutes <p>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.</p>

<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#>

<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.

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.

Required columns

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.
totalPeriodHandle TimeSeconds	nonnegative double	<p>The total time the agent spent handling contacts (talk, hold, work) for the service queue that occurred within the boundaries of the period.</p> <p>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).</p>

Required columns (continued)

Column	Data Type	Description
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.

Optional columns

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.

Optional columns (continued)

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 out-bound 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.

Optional columns (continued)

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>

Optional columns (continued)

Column	Data Type	Description
		<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-service (Talk, Hold, After Work, Ready/Waiting, and Reserved) time.</p>
totalProratedNotReady 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>

Optional columns (continued)

Column	Data Type	Description
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>
totalAnswerDelaySeconds	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: 0

acdAgentId,acdServiceId,contactsHandled,contactsAnswered,contactsTransferredOut,contactsExternalIn,contactsExternalOut,totalTalkSeconds,totalHoldSeconds,totalAfterContactWorkSeconds,totalPeriodHandleTimeSeconds,totalUnproratedReadyWaitingSeconds,totalUnproratedNotReadyBusySeconds,totalUnproratedInSessionSeconds

5009,5236,7,8,0,0,0,933.728,0.000,228.718,1215.857,12.000,102.460,1800.000

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

<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>	<p>The time zone (where the ACD is located) date/time offset from GMT in AHHMM format, where:</p> <ul style="list-style-type: none"> ■ A is either P (positive) or N (negative), characterizing the offset from GMT ■ HH is the number of offset hours ■ MM is the number of offset minutes <p>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.</p>
<ACD#>	<p>The ACD number. Can be zero (0) or a positive number with as many digits as desired.</p>

File Header

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

<date>	<p>The date the file was generated, in YYYY-MM-DD format.</p>
<time>	<p>The beginning of the 30-minute interval covered by the data, in HH:MM 24-hour format.</p>
<tzoffset>	<p>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).</p>
<ACD#>	<p>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.</p>

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).

Required columns

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.

Required columns (continued)

Column	Data Type	Description
<p>contacts Answered</p>	<p>nonnegative double</p>	<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>contacts Abandoned</p>	<p>nonnegative double</p>	<p>The number of contacts for the service queue abandoned by callers during the period.</p>
<p>totalTalk Seconds</p>	<p>nonnegative double</p>	<p>The total talk time on contacts for the service queue handled by agents during the period.</p>
<p>totalHold Seconds</p>	<p>nonnegative double</p>	<p>The total hold time on contacts for the service queue handled by agents during the period.</p>
<p>totalAfter ContactWork Seconds</p>	<p>nonnegative double</p>	<p>The total after contact work time on contacts for the service queue handled by agents during the period.</p>
<p>totalAnswer Delay Seconds</p>	<p>nonnegative double</p>	<p>The total delay in answering contacts for the service for contacts the agent answered during the period</p>
<p>serviceLevel Percent</p>	<p>nonnegative double ≤ 100</p>	<p>The percent of contacts for the service during the period that were answered within the service level threshold.</p>

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.

Optional columns

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.

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 add 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
acdServiceId, contactsOffered, contactsHandled, contactsAnswered, contactsAbandoned, totalTalkSeconds, totalHoldSeconds, totalAfterContactWorkSeconds, totalAnswerDelaySeconds, serviceLevelPercent
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

<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.
<ACD#>	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.

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.

Required columns

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.

Optional columns

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 states as associated codes

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

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 8.9. 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 TAC 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 ACD ID in WFM (Application Management > People > Agents).
gisStateIdentifier	<p>The identifier for the agent state.</p> <ul style="list-style-type: none"> ■ 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. ■ 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. ■ 3—Ready. The agent is ready to accept contacts from the ACD. ■ 4—Talking. The agent is on a contact. ■ 5—Work Not Ready. The agent is engaged in after contact work and will enter a Not Ready state when finished. ■ 6—Work Ready. The agent is engaged in after contact work and will enter a Ready state when finished. ■ 10—Hold. The agent has placed the contact on hold. <div style="border: 1px solid black; background-color: #e1f5fe; padding: 5px; margin-top: 10px;"> <p>Note: 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.

Field	Descriptions
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.

Index

A

ACD numbering 8
AgentProductivity.AGENT file 11
AgentState.EVENT file 27

D

Data capture requirements 7
Data type definitions 9

F

File locations 8

R

Real-time data capture API 33
 definition 33
Requirements 7

S

ServiceHistorical.SERVICE file 20

T

Text files
 AgentProductivity.AGENT 11

AgentState.EVENT 27
 details 9
 locations 8
ServiceHistorical.SERVICE 20

