



USER MANUAL

Version 1.0.0b



INDEX

What is WP CallCenter Inbound?	4
Logging on to WP CallCenter Inbound	5
License information	7
Running a report	7
Quick activity reports	7
Agent report	9
Custom reports	9
Understanding results: Common header	11
Understanding results: Answered calls	11
Agents on queue	11
Service level agreement	12
Disconnection causes	12
Transfers	12
Answered calls by queue	13
Detail of answered calls	13
Understanding results: Unanswered calls	13
Disconnection causes	15
Unanswered calls, by queue	15
Details of unanswered calls	15
Understanding results: Call distribution	15
Call distribution per day	16
Call distribution per hour	17
Call distribution per day of week	18
Understanding results: Agent activity	18
Agent availability	21
Session and pause duration	21
Answered calls	21
Answered calls by service groups	21
Session details	21

Version: 19/01/06



Showing call details	22
Detail of answered calls	22
Listening to answered calls	23
Detail of unanswered calls	24
Area code analysis	25
The real-time status panel	26
Top status panel	29
Calls being processed	29
Agents currently logged in	30
The real-time agent page	31
Storing queue data on MySQL	35
Who should use MySQL storage?	36
Understanding MySQL storage	36
Uploading data to MySQL	37
Loading data in WP CallCenter Inbound	38
Editing WP CallCenter Inbound settings	40
Configuring users	41
Editing user classes	44
Configuring queues	45
Configuring agents	46
Configuring system preferences	49
Configuring WP ePhone for WP CallCenter Inbound	50
Configuring queues to report exit status	50
Configuring URLs to be launched by the agent real-time page	51
Listening to calls using WP CallCenter Inbound	51
Using AddQueueMember for dynamic agents	52
For more information	52
Appendix I: Default users	52
Appendix II: Security keys	53
Appendix III: Glossary	54



What is WP CallCenter Inbound?

WP CallCenter Inbound is a versatile call center monitoring system dedicated to call centres based on the WP ePhone PBX.

WP CallCenter Inbound lets you...

- Run reports on call center activity, divided by queue and filtered by agent and time period, that show what happened (e.g. taken calls, lost calls, agents logging on and off...) during the specified period. Such reports can be run while WP ePhone is running, so that you have no delay in seeing what's going on.
- See the details of call center activity, like each single call that was handled or lost, and listen to it through your web browser.
- Have a single real-time panel showing call center activity; you'll see calls being processed by queues and agent activity in the very moment it's happening.
- Give your agents a web-based interface panel that lets them see their own calls while they're being handled and optionally launch an external web-app (like a third party CRM module) as the calls come in.

To meet these goals, WP CallCenter Inbound processes a file called queue_log, i.e. the log file where WP ePhone writes signalling events on call queues. WP CallCenter Inbound is preconfigured with the standard WP ePhone installation paths so it will work out-of-the-box for most installations.

WP CallCenter Inbound is meant to be highly customizable; you can alter much of its behaviour to fine-tune it to your own needs (and display your company's logo....).

WP CallCenter Inbound is an intranet application as is designed to be used through a web browser. There is no software to install on the client machines. You can access it from anywhere, as long as you have the correct credentials.



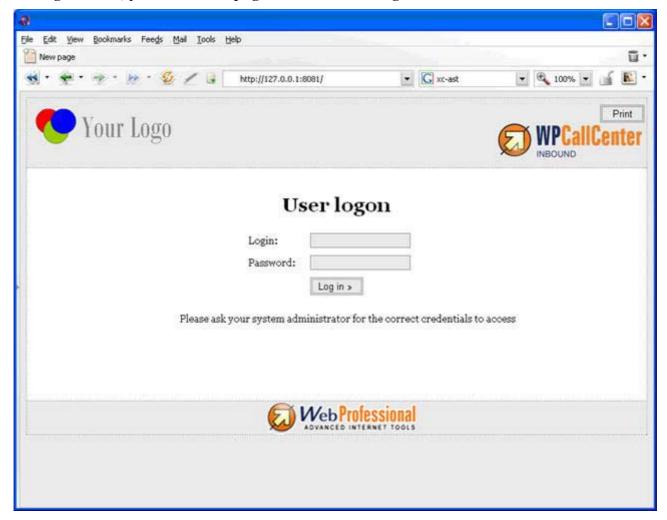
Logging on to WP CallCenter Inbound

To log on to WP CallCenter Inbound, you have to point your browser to the address of the server where you installed WP CallCenter Inbound. As servlet containers are often installed on ports different than the standard HTTP one, it might be necessary to specify the port address.

For example, if you install Tomcat 5 on the same server you're accessing WP CallCenter Inbound from, you may end up pointing your browser to: http://localhost:8080/wpccin

Ask your system administrator for the correct web address of your instance of WP CallCenter Inbound.

If all goes well, you will see a page like the following one:

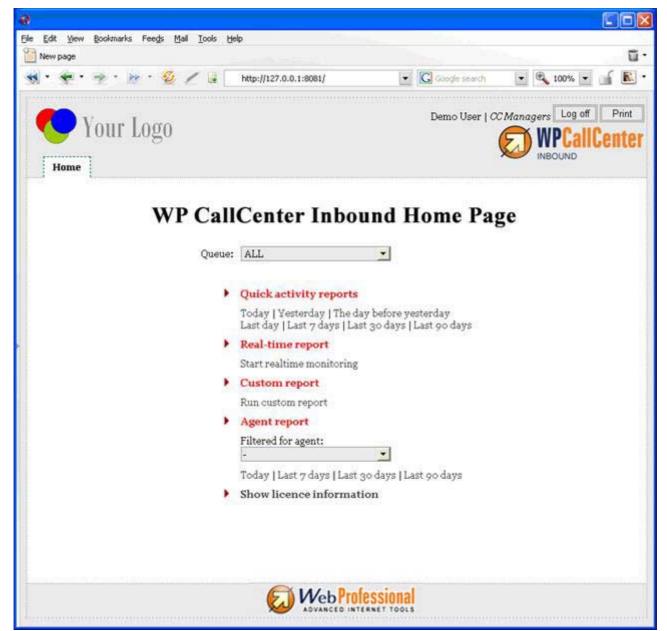


This and the following screenshots are taken using Opera 8 on Windows; other environments may present minor discrepancies from what is shown here.



If your system administrator has already configured WP CallCenter Inbound, you might see you firm's logo on the top left part of the screen and a different welcome message.

To enter the system as a user, enter the standard credentials **demouser** with password **demo** and click on the "Log in" button, or use the credentials your administrator has provided.



The user is presented with the Home Page, that is the starting point of WP CallCenter Inbound. The name of the user and the current class for the user are shown on the topright corner of the window.

To end the current session, you have to press the "Log off" key or close the browser window.



To print the current page in a printer-friendly format, you just press the "Print" button.

License information

Pressing the "Licence information" label, a page like the one below is shown.



This page shows the current release of the software and the current license information.

If you are running a free demo version, you will see that the maximum number of licensed agents is 2 and an additional text will remind you on how to register.

Running a report

To successfully run a report, your system administrator must have configured the correct queues in use on your system. You will find them in the drop-down menu on top of the page. See page 45 for details on how to do it.



Quick activity reports

The quickest way to obtain an analysis is by selecting the queue you want to analyze and then click on the appropriate time frame below the "Quick activity reports" title on the home page.

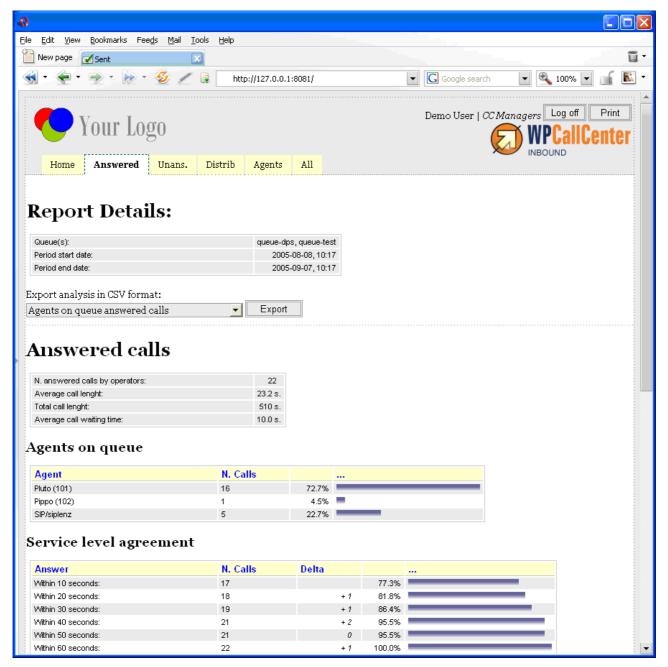
The defined time frames are the following:

- Today, Yesterday, The day before yesterday

 The day in question, starting from midnight to midnight
- Last day, Last 7 days, Last 30 days, Last 90 days
 The exact time period, starting from the current hour backwards.

The system will then show the "Answered calls" page, like here below.





On top of the page, you can see a multi-tab menu; by clicking on it you can select which part of the report you are going to see. To go back to the home page, click on the "Home" tab. You can also see all the analyses at once by clicking on the "All" label (this is mostly useful when printing the results to paper).

Agent report

If the user has the appropriate grants, s/he can restrict the analysis to a single agent. This way one can see exactly what one agent did.

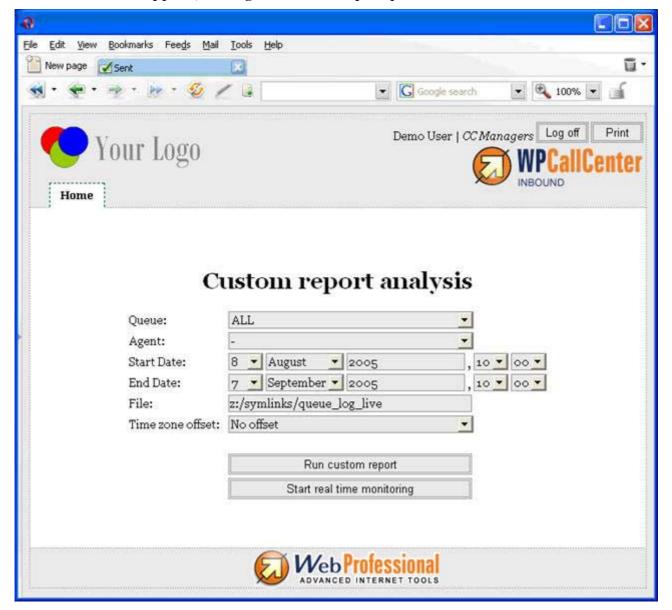
To use this feature, select the agent you want to filter by and click on the desired time period in the "Agent report" section of the Home Page.



Custom reports

Custom reports are available by clicking on "Run custom report" from the Home Page.

A new menu will appear, asking for custom report parameters:



The meaning is as follows:

- *Queue* is the queue or composite queue you want to analyze;
- *Agent* is an agent to be used as filter, or "-" as no filter;
- *Start* and *end date* let you select the period you want to analyze, with five-minute resolution;



- *File* is the queue_log file you want to analyze. You may want to change it to run reports on a different WP ePhone server or on an older archived version of your queue_log. If you run WP CallCenter Inbound on the same machine as WP ePhone, the file name should be already correct. Make sure the file is readable to your servlet container.
- *Time zone offset* is to be set if the WP ePhone server that created the queue_log file was in a different time zone from the one you are using.

By clicking on the "Run custom report" button, you can run the analysis, which output is the same as the "Quick activity report" and will be explained below.

Understanding results: Common header

On the top of each report, a box will be shown showing:

- Which queue or queues were considered for the analysis
- The time period the analysis refers to
- Whether the report is about the whole of the queue or a specific agent

There is also a box showing a number of analyses you can export in CSV format.

When running in report mode, WP CallCenter Inbound considers an event (say a call or an agent session) as soon as it's over; i.e. you will not see calls that are currently in progress. If you need to see calls in progress or whether an agent is logged in, you have to run the Real-time panels.

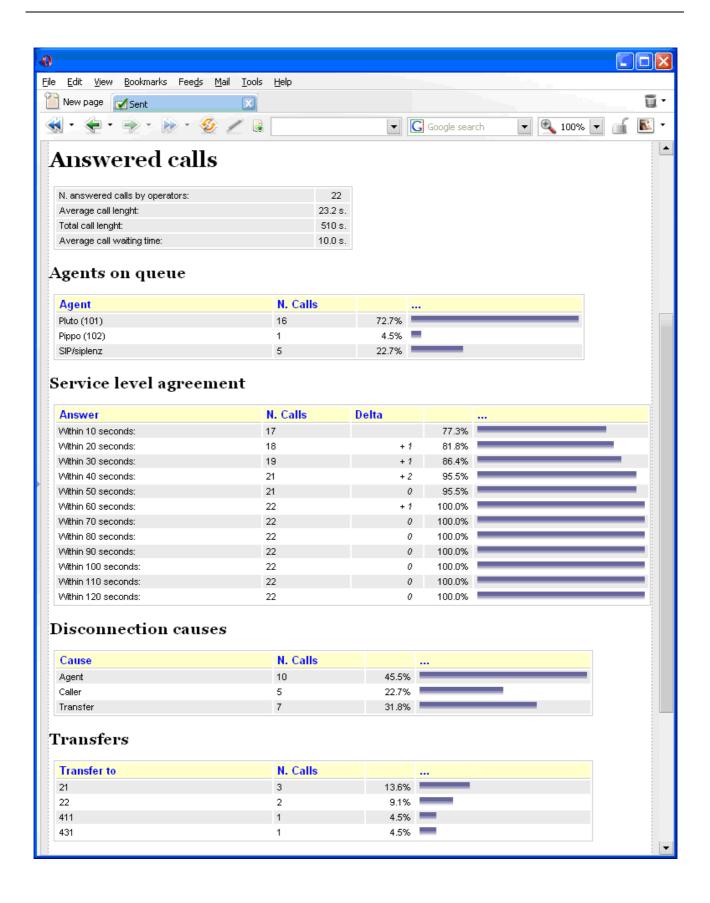
Understanding results: Answered calls

The answered calls section deals with calls that were correctly handled by agents.

The top panel shows:

- How many calls were handled;
- The average call length (i.e. time the caller spends talking to an operator);
- The total call length (for all calls on all operators);
- The average call waiting time (i.e. the time a caller was waiting on a queue before being connected to an operator).









Agents on queue

This report shows which agents have been available for the given queue, how many calls each one handled and the percentage of all calls that each one handled.

If calls are connected directly to a phone terminal, WP CallCenter Inbound tries its best to show the corresponding terminal, usually in the format used by WP ePhone, like "SIP/303" to signify a SIP phone whose number is 303.

If you connect to H.323 telephones via the OH323 module, the recorded channel names have no meaning and do not refer to a specific terminal; that's why all OH323 calls are grouped together under the label "OH323/-".

Service level agreement

This report shows the distribution of call waiting times. It shows how many calls were answered within a given time frame, usually 120 seconds in 10 second increments (the time frame and increment can be modified by the administrator, if needed – see page 49).

You get a percentage of how many calls were answered within X seconds; the percentage includes calls answered in a shorter time frame and therefore grows with time.

The "delta" value you see is the absolute increment, expressed in number of calls, between each time frame.

Disconnection causes

This report shows the reason why calls were terminated; this means that:

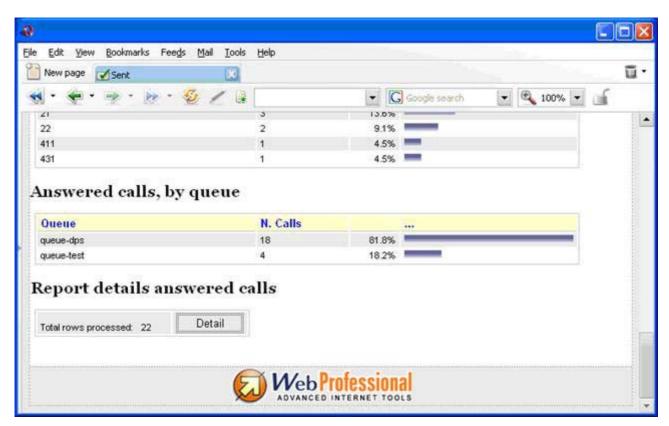
- The agent hung up, or
- The caller hung up, or
- The call was transferred outside the queue and the agent was freed again

Transfers

This graph shows how many calls were transferred to each extension in the given time frame. This lets you know who is handling exception calls.

Note: when a call is transferred outside the queue system, its length is no more recorded by the queue subsystem; therefore you only get to see the length of the call while the agent was on line.





Answered calls by queue

If more than one queue is in use for the report, this graph shows the relative magnitude of each queue.

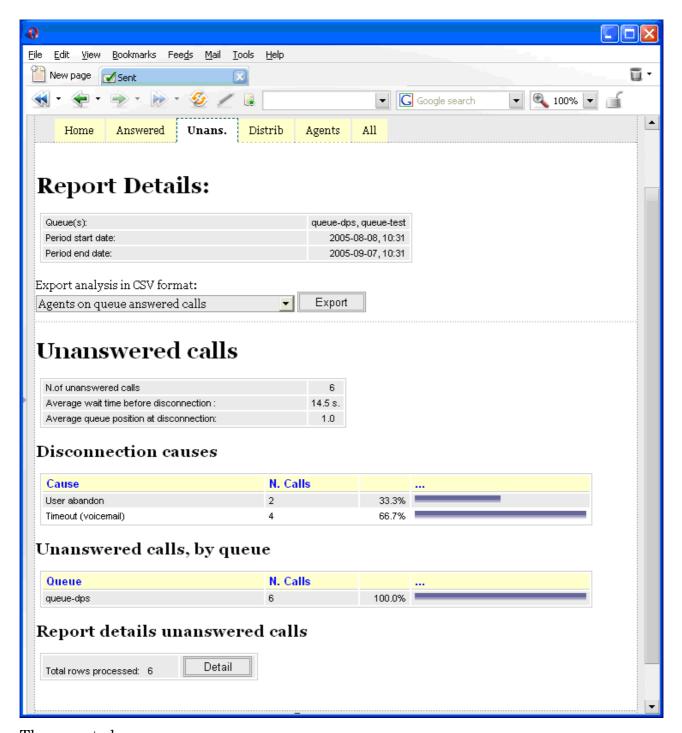
Detail of answered calls

This page shows the detail of answered calls (see page 23).

Understanding results: Unanswered calls

Unanswered calls are calls that were lost, i.e. the caller could not connect to an agent. This usually means that either the caller hung up, fed up with waiting, or the queue system decided to discharge the caller, maybe sending him to voicemail or another queue.





The report shows:

- How many calls were lost;
- The average waiting time before disconnection;
- The average queue position at disconnection (i.e. how many calls the queue had to dispatch before connecting the caller to an operator).



Disconnection causes

This report shows the relative magnitude of disconnection causes, that are:

- The caller hung up, or
- The queue timed out and discharged the caller (if this feature is enabled by the queue configuration see page 50), or
- The caller exited the queue by pressing a key (if this feature is enabled by the queue configuration).

Unanswered calls, by queue

If more than one queue is in use for the report, this graph shows the relative magnitude of each queue.

Details of unanswered calls

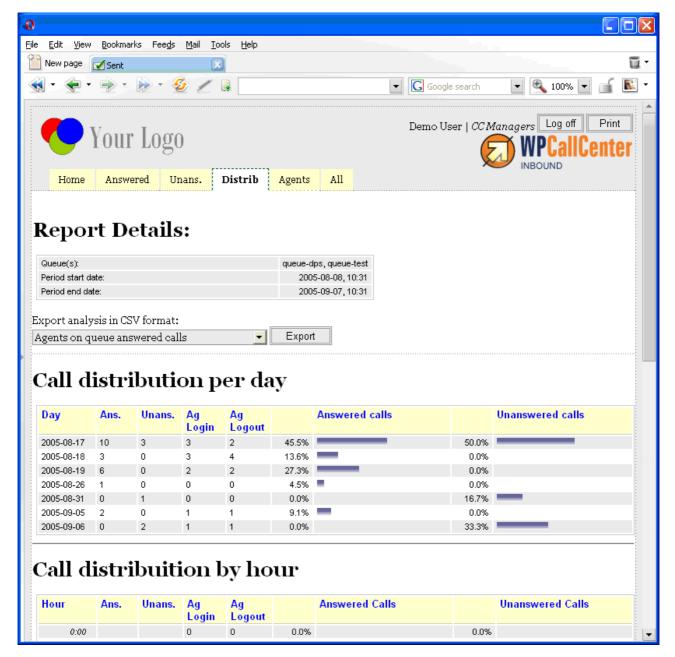
This page shows full details of unanswered calls (see page 25).

Understanding results: Call distribution

The call distribution report shows when calls were handled, when calls were lost and when agents logged in and out of your system.

All percentages are calculated on the call class they belong to, i.e. a 50% of "Unanswered calls" on one day means that 50% of all unanswered calls for the period happened during that day, not that 50% of calls were lost.

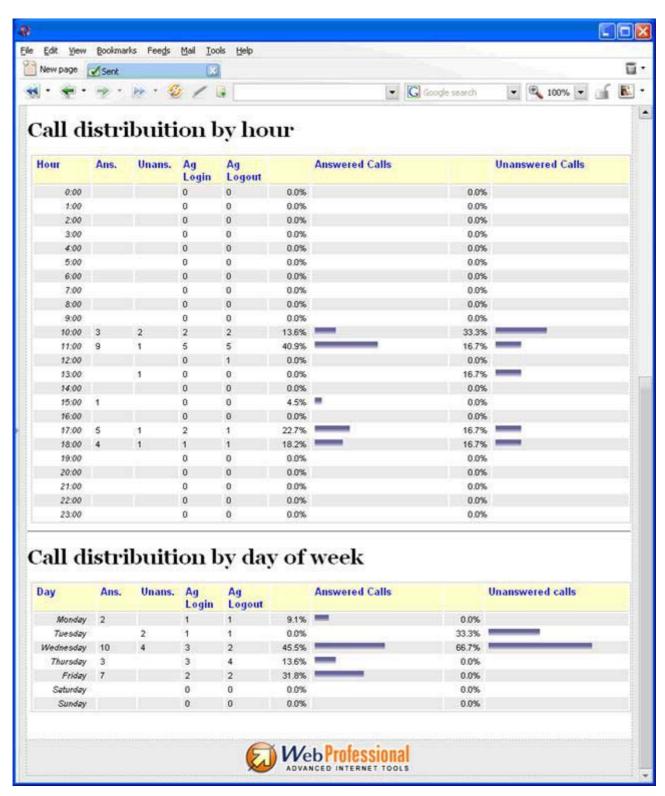




Call distribution per day

Call and agent events are shown per specific day. Days with no events are not shown.





Call distribution per hour

Events are shown on a 24-hour distribution. If this graph appears to be incorrect, you have to run a "Custom report" setting the time zone accordingly (see page 10).



Call distribution per day of week

This report shows the weekly behaviour of your queues. The longer the analysis period, the more significant its results will be.

Understanding results: Agent activity

Agent activity refers to the behaviour of WP ePhone defined agents. If you connect you queues straight to telephone terminals, this section will always be empty.

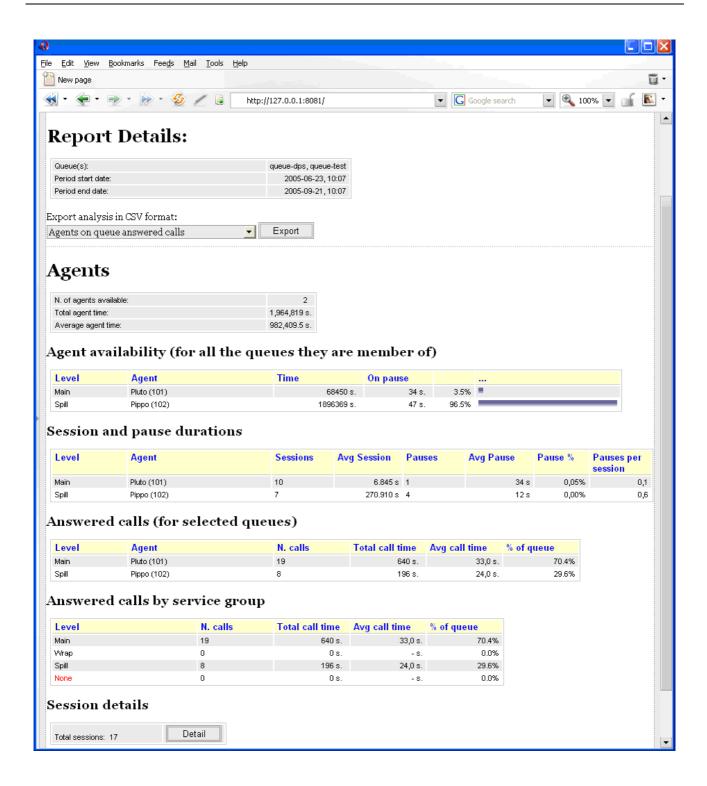
Each agent may be flagged as being a member of four priority groups:

- *Main*: the agents usually answering the queue
- *Spill*: the agents answering the queue if all "Main" agents are busy or unavailable
- *Wrap*: the agents answering the queue if all "Main" and "Spill" agents are busy or unavailable
- *None*: this agent is not a member of any priority group for this queue

This feature is useful if priority groups are used in the queue configuration. If they are not used, just assign all agents to "Main" for each queue.

If an unknown agent appears on a queue, it will be marked as "None", written in red.





The report shows:

- How many agents were available for the queue
- How much time all agents have been available
- The average agent available time



Agent availability

This graph shows which agents were available during the specified time frame and the percentage of agents' available time each one cumulated.

This time is calculated per all queues any agent is a member of, as the act of logging on is in general for the whole system and not specific to one single queue.

For each agent, the total time on pause – if any – is computed.

Session and pause duration

For each agent, the total number of sessions and pauses is computed (session time is already deducted of pause time). For both sessions and pauses, an average length is computed.

The "Pause percentage" is how much time an agent was on pause compared to available time.

The "Pauses per session" computes how many pauses – on average – each agent makes for each log-in session.

These metrics should be considered according to your call center rules on pauses and time-out.

Answered calls

This graph shows who of your agents answered calls for the queues you selected. The number of calls, together with total and average call durations are computed accordingly.

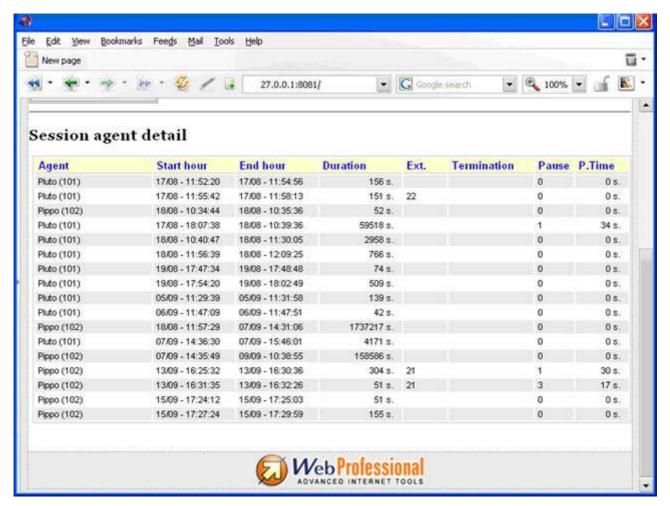
Answered calls by service groups

This graph show which priority levels handled calls for your queue. This shows whether your main line is staffed enough to handle the load of incoming calls.

Session details

By clicking on the "Detail" button, a new page is shown, detailing each agent session that was recorded.





For each agent session, the start and end times are recorded, together with the total duration in seconds.

If the agent logs on via the call back function, the designated call back extension is shown.

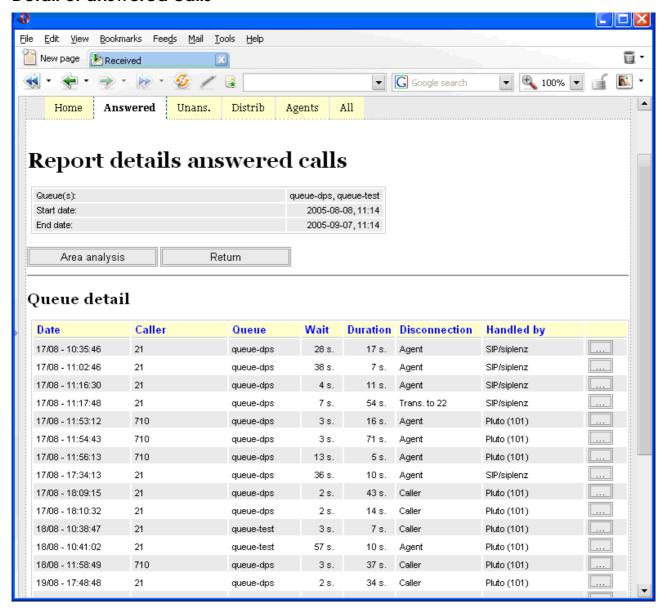
The number of pauses and the total pause time in seconds is shown.



Showing call details

As shown above, WP CallCenter Inbound lets you see the very detail of calls handled by WP ePhone.

Detail of answered calls



For each answered call, the following information is shown:

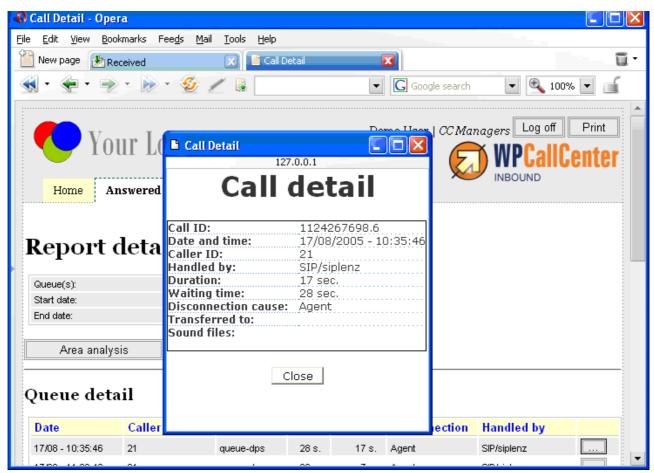
- Date and time for the call;
- The Caller*ID, if available (the Caller*ID format may differ according to your local telco in some countries it include the full name of the caller, in other it might be a number and in others it may be unavailable);



- The queue that handled the call;
- The total waiting time before the agent was connected;
- The duration of the call, talking to an agent;
- The cause of disconnection:
- Which agent or terminal handled the call.

Listening to answered calls

Clicking on the button with three dots near to a call opens a detail popup, like the one below:



For each call, the recorded pieces of information are shown.

If the call was monitored, i.e. recorded to disk, a number of sound files may be shown. By clicking on a sound file you can listen to it straight from your browser.

Please note that:

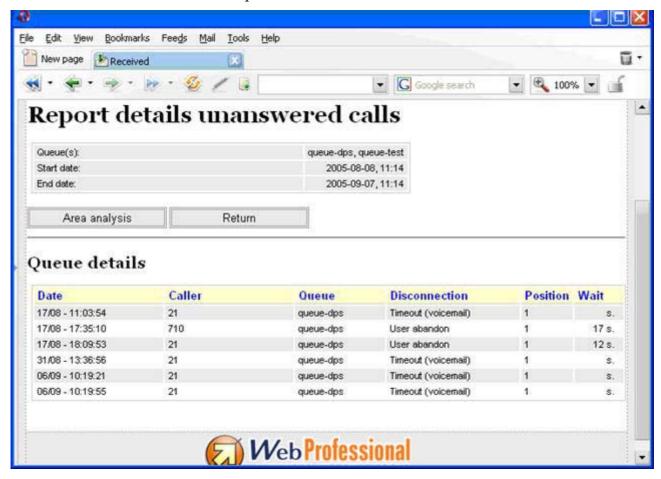
• The recorded file name must contain the WP ePhone Call ID for WP CallCenter Inbound to relate it to the call – see page 51 for tips on how to configure WP ePhone correctly to implement this feature;



- The audio storage on the WP ePhone server must be readable by the servlet container;
- You must have the correct sound codecs to listen to the sound file on your PC. WAV files usually work out of the box but are comparatively quite big, while GSM files require an additional codec pack on most Windows machines but consume disk storage much more efficiently.
- WP ePhone will usually record two different sound files one for the caller and the other for the agent and will then mix them together at the end of the call. If this does not happen automatically, you might find two different sound files, named "-in" and "-out", each of which contains the voice of one of the parties.

Detail of unanswered calls

The unanswered calls detail is quite similar to that of answered calls.



The following data are shown:

- Date and time of the lost call;
- Caller*ID:
- Queue that handled the call;



- Disconnection cause;
- Position at disconnection, if available;
- Waiting time before disconnection, if available.

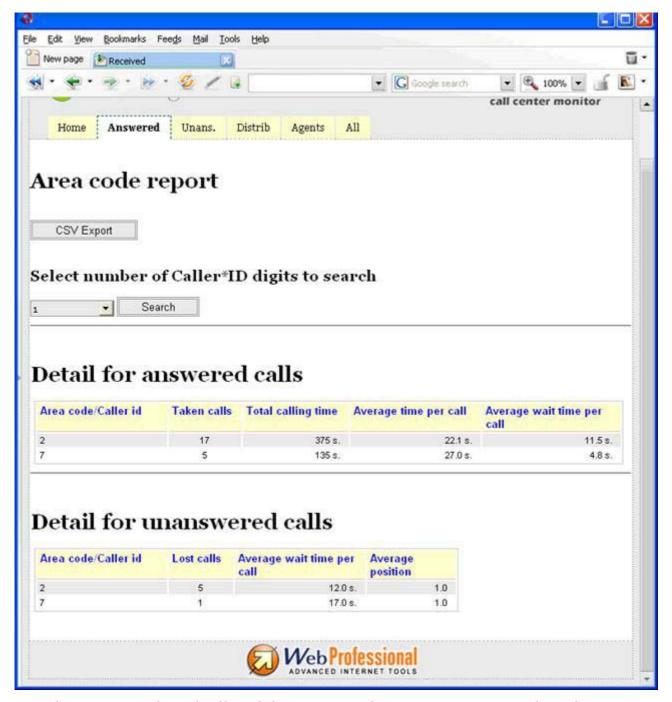
Please note that on a queue timeout, WP ePhone will not report the waiting time, as it is fixed and same as the queue timeout.

Area code analysis

If the Caller*ID is present, it is possible to break down both answered and unanswered calls to specific area codes by clicking on the "Area code analysis" button.







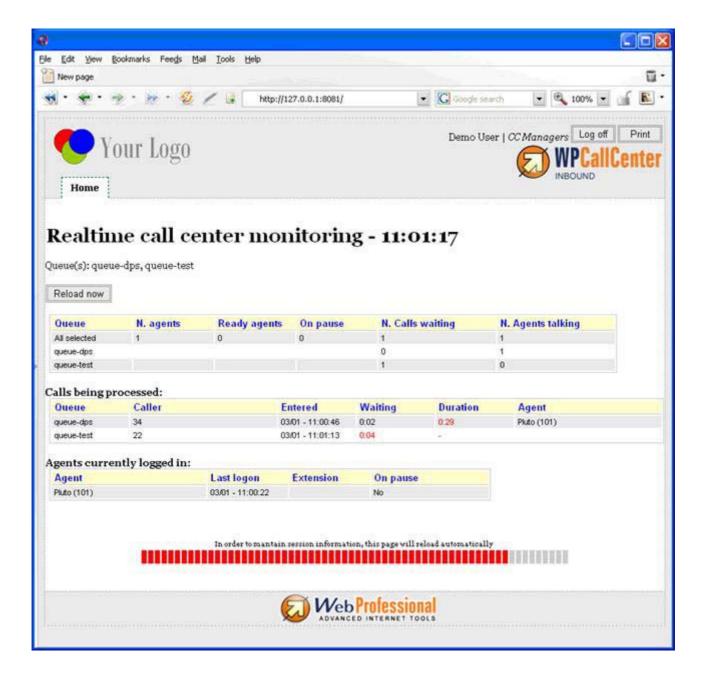
By selecting a number of caller id digits to search upon, you get a number of statistics grouped by area codes.

This report gives an immediate check of the geographical origin of calls handled by your call center.



The real-time status panel

The real time status panel can be accessed by clicking the "Start real-time monitoring" label from the home page. It will show a page similar to the one below:



On the top of the page there is a table showing which calls are currently handled by the queue system, and the agents logged in at the moment. This page is invaluable as it refreshes automatically and can tell you in a glimpse what's happening in the call center; it is meant to stay open in a window on the CC manager's workstation to have the exact feeling of what is going on at the moment.



On the sample page above, you can see two calls (one being handled by Agent 101 and the other one waiting) and one connected agent. Just like in the main analysis, you can choose which queues you want to monitor to avoid being overwhelmed by data.

On the bottom of the screen, a moving bar shows the time left before the next reload. The page reloads automatically in order to show what's happening almost in real time. You can anyway force a faster reload by clicking the "Reload" button.

Top status panel

The top status panel show a quick status report for the current situation.

The first line shows information for all selected queues as a sum, while if there is relevant information for a specified queue it is displayed in a separate line.

The displayed fields have the following meanings:

- Queue: The name of the queue
- *N. agents*: how many agents are logged on to the system, in total
- *Ready agents*: how many agents are ready to take calls, i.e. are logged on and are not in conversation or on pause
- On pause: how many agents are currently on pause
- N. Calls Waiting: how many calls are currently waiting in the selected queue
- N. Agents talking: how many agent are talking on the selected queue

Please note that – as agents are not linked to a specific queue save for the moment they are actually talking to a caller on the queue – the agent information is computed for all agents on the WP ePhone server and not for specific queues.

Calls being processed

A list of calls flowing through the selected queues is presented on the middle table. If no call is present the table is displayed empty.

When a call is processed, the following fields are shown:

- Queue: the queue that is handling the call;
- Caller: The Caller*ID, if available;
- Entered: The date and time the call entered the queue system.

If the call is not answered yet, the "Waiting" field is displayed in red and is calculated according to the current date and time of the server. If your WP CallCenter Inbound is on a different server, make sure the clocks are exactly aligned or you may see strange values in this field. The NTP protocol offers excellent clock synchronization precision and is available on most operating systems.

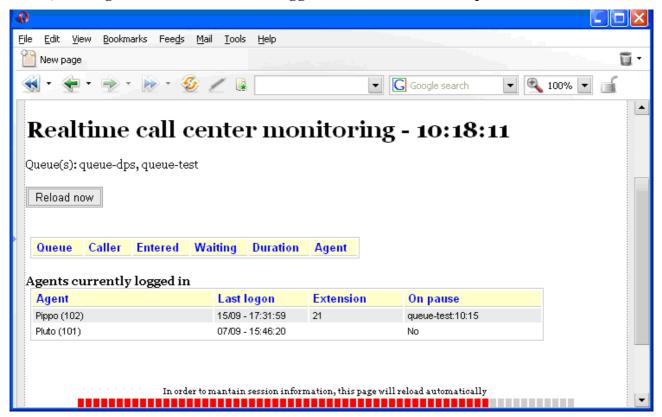


When a call is answered, the "Waiting" field tells the time that the caller had to answer; the "Agent" field shows the agent (or terminal) the caller is talking to and the "Duration", in red, is the current call duration.

As soon as a call is completed or hung up, it exits the Calls panel.

Agents currently logged in

A list of available agents for all queues is displayed in this field. For each agent, the name, last log on and extension – if logged in via call-back – is provided.



If an agent goes on pause, like Agent 102 in the screenshot above, the queue(s) he has paused from are shown and so is the time he left for pause.



The real-time agent page

WP CallCenter Inbound lets each agent have his/her own page, where they can see the current flow of calls they have just answered and launch external CRM web apps.

This is quite useful, because:

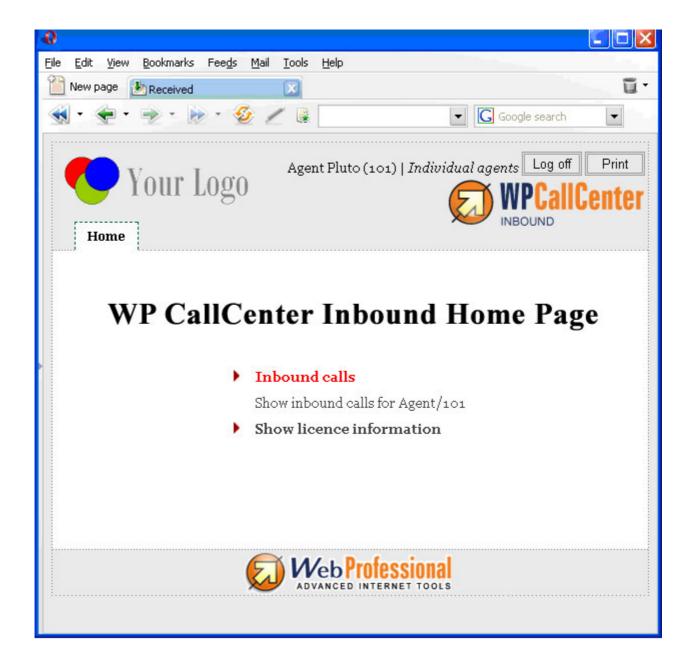
- Each agent can see their own status, i.e. whether they're logged on or they've been disconnected;
- Each agent can see their last calls, including information like Caller ID, duration and waiting times;
- Each agent can see from which queue the call is coming, even if they lost the announcement message;
- Each agent can launch external web apps like CRM software that might be automatically linked to the Caller*ID or other information input by the caller.

To avoid excessive consumption of system resources on big centres, only the most recent part of the log file is processed and so only a few calls are shown.

In order to use this feature, a user must be configured as having the same login as the WP ePhone agent string (e.g., "Agent/101") and it must hold the key AGENT. Our suggestion is to use the same set of credentials the agent uses to login to the WP ePhone system.

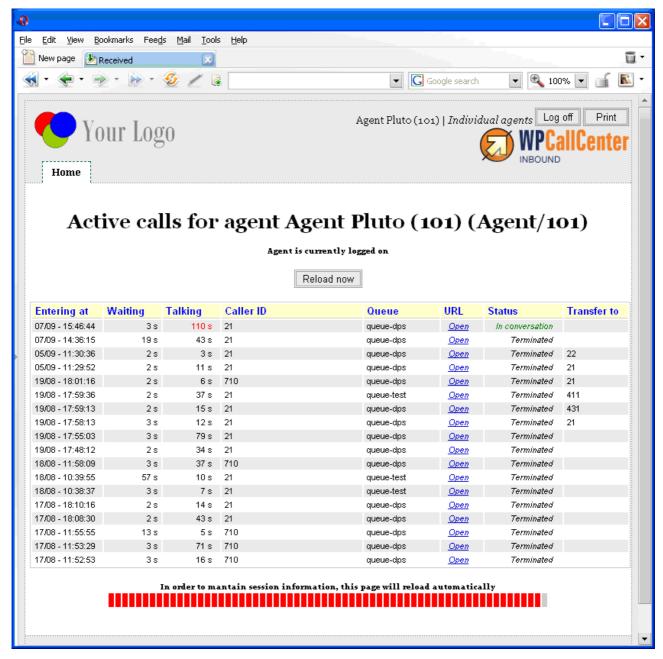
When the agent logs on - you can try this by using the demo account **Agent/101**, password **999** - s/he sees a reduced home page like the following one:





By clicking on "Show inbound calls", the agent is led to the inbound calls page:





This page tells the agent that he's logged on and shows the last calls he has handled.. In this case we see that there is an ongoing conversation and many previous calls.

The fields are as follows:

- *Entering at* is the date and time the call entered the queue system;
- *Waiting* is the waiting time before being connected to the agent;
- *Talking* is the talking time for that call. If the call is ongoing, the time will be estimated and written in red.;
- *Caller ID* is the call's identification, if available;



- *Queue* is the queue handling the call;
- *URL* (if present) is a clickable link that opens a third party CRM app. It is defined in the Queue command on WP ePhone. The agent opens the third party CRM app by clicking on it.
- *Status* is whether the call is ongoing or terminated;
- *Transfer* is the extension the call was transferred to (if any).

On top of the page a field tells the agent whether s/he is:

- *Logged on*: ready and able to take calls;
- *Logged off*: the agent has voluntarily left or has forcibly been disconnected by the queue system;
- *On pause*: the agent has asked for a pause from the queue system;
- *Undetermined*: whether there is no relevant information to tell the agent status in the last part of the log file.

To avoid hammering the WP CallCenter Inbound server with excessive work, only the last 60k or so of the log are analyzed. This parameter can be fine tuned by the system administrator in order to maximise usefulness without creating an excessive server load (see page 49).

Whenever an agent receives a call, s/he should press the "Reload" button on the page in order to see the incoming call.

The page reloads automatically every two minutes in order to maintain the user logged on in WP CallCenter Inbound.

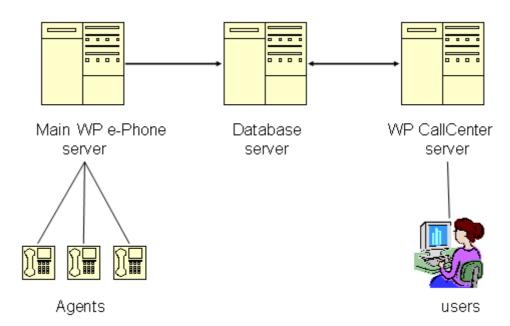


Storing queue data on MySQL

WP CallCenter Inbound lets you store the queue_log data on a MySQL table and is able to produce the very same analyses – including real-time analyses – from data stored on a database.

This scenario is mostly useful for large call centres, where the queue_log data starts to be quite large and the main WP ePhone server is quite busy handling its own traffic. In this case, it would be a better solution to have WP CallCenter Inbound run on a separate server, so that even if it has to run a huge analysis the main WP ePhone server will not be slowed down.

WP CallCenter Inbound lets you have a deployment scenario like the following one:



In this case, we see that we are using two separate servers; one for the database and one for the WP CallCenter Inbound server itself. It is possible to use the same server for both the database and WP CallCenter Inbound, or to consolidate the database on an existing database server and WP CallCenter Inbound on an existing servlet container.

It is very important that all the servers share the same system time; this way real-time events will be shown in an exact way¹.

WP CallCenter Inbound Version: 19/01/06

_

¹ The *ntpdate* command can be used on Linux to synchronize the system clock to an external timing source with a high degree of precision. Usage in a daily cron script is highly recommended.



Who should use MySQL storage?

We believe that MySQL storage will be useful in the following scenarios:

- Large call centres with a very busy or mission-critical WP ePhone server
- Large WP CallCenter Inbound reports run very often
- A large number of agents reloading WP CallCenter Inbound agents pages

In smaller environments, it is probably overkill to use MySQL storage, because the extra complexity will not be matched by an extra performance advantage.

Understanding MySQL storage

The WP CallCenter Inbound database storage engine was built with a need to adapt to existing MySQL schemas; therefore the database storage option is very flexible.

It lets you:

- Define the names of each SQL field
- Define the name of the SQL table used (it must reside in the same database as the other WP CallCenter Inbound tables)
- Define one or more table partitions

The storage system makes no assumptions on the underlying field layout of the table used, therefore you are free to define each field as you best see fit for your scenario.

To obtain these results, the SQL settings are divided into presets and partitions.

A **preset** is a schema definition to be used, i.e. the names of each field involved in database storage. You can have a number of different presets, e.g. to connect to different tables in the same database. Presets are defined in the *WEB-INF/configuration.properties* file.

A sample preset can be seen here:

```
# Preset 1: standard DB access. Edit to suit your DB needs.
sqlPreset.1.table=queue_log
sqlPreset.1.f_time_id=time_id
sqlPreset.1.f_call_id=call_id
sqlPreset.1.f_queue=queue
sqlPreset.1.f_agent=agent
sqlPreset.1.f_verb=verb
sqlPreset.1.f_partition=partition
sqlPreset.1.f_data1=data1
sqlPreset.1.f_data2=data2
sqlPreset.1.f_data3=data3
```





sqlPreset.1.f data4=data4

You can have more than one preset, by entering the same data multiple times under sqlPreset.1...., sqlPreset.2....., sqlPreset.3..... and so on.

The values for each field are:

- *Table* is the table name
- *Time_id* is the first field in the queue_log. This is used for most extractions and should be an access key.
- *Call_id* is the second field of the queue_log
- Queue is the third field of the queue_log
- Agent is the fourth field of the queue_log
- *Verb* is the fifth field of the queue_log
- *Data1...Data4* are the remaining fields of the queue_log. Currently Data4 is not defined in the queue_log; in case just leave it blank.
- *Partition* is a logical partition of the table.

A **partition** is a key under which separate entries are present in the same queue_log table. You could have separate servers – like test and production - uploading each one to a different partition, and each of them would be completely independent.

If you use a partition, your *partition/time_id* combo should be an access key for the table, as WP CallCenter Inbound will access the table every time under this plan.

If you do not use a partition, just leave this field blank and make sure that time_id is an access key for the table.

Uploading data to MySQL

There are a number of ways for data to be uploaded into MySQL. If we plan to use the real-time monitoring features, we must upload data to MySQL as events happen, in order to have them seen immediately by WP CallCenter Inbound.

We provide a sample data upload script that can be used to upload data to MySQL, either in batch (one file at a time) or in real-time mode.

The script can be found under *WEB-INF/mysql-utils* and is called *queueLoader.pl* (a working installation of Perl5 with MySQL DBI is required to run this script).

To use the script, edit it with a text editor in order to set the MySQL server, database, user and password, like in the following example:

```
my $mysql_host = "10.10.3.5";
```



```
my $mysql db
             = "log code";
my $mysql user = "WP CallCenter Inbound";
my $mysql pass = "javadude";
```

After the database is set, you can upload a whole file (in this case, /my/queue_log) to a partition named "P01" by running:

```
perl queueLoader.pl /my/queue log P01
```

You can also upload your queue log file to partition "P02" as it is being written by issuing:

```
tail -n 0 -f queue log | queueLoader.pl - P02
```

(do not forget the minus sign, separated by spaces, before the partition name!).

Loading data in WP CallCenter Inbound

After you configured the table in *configuration.properties*, using the table is only a matter of inputting "sql:[partition] | [preset]" as the queue_log file name to analyze. The partition defaults to "" (blank) if absent, while the default preset is 1.

You can do it directly from the "Run custom report" form, or preset the file name in configuration.properties as you best see fit by setting the default.queue_log_file property.

Examples:

sql:P03

Means accessing the partition named "P03" for preset #1.

```
sq1:X23|3
```

Means accessing the partition named "X23" for preset #3.

sq1:|2

Means accessing the present #2 with no partition, and

sql:

Accesses preset #1 with no partition.





If you use agents pages, keep in mind that the value in *realtime.max_bytes_agent* will not be the portion of the queue_log to be read, but the time interval (in seconds) that will be read for the current agent (i.e. if set to 10000, it will search agent data for the last three hours or so).

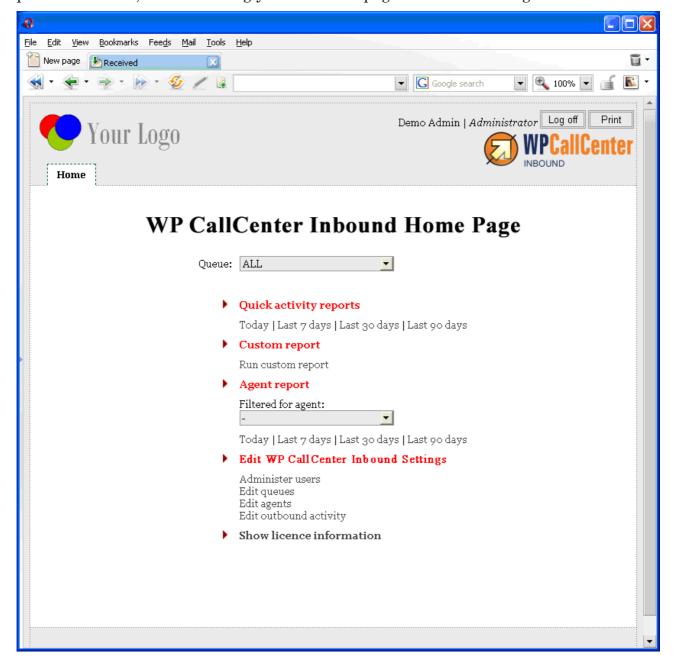
When you enter a "sql:" file name, the error "The file sql: does not exist" means that there is a misconfiguration of the table access fields in *configuration.properties*.



Editing WP CallCenter Inbound settings

System configuration must be done by the system administrator. Most configuration may be done straight from WP CallCenter Inbound itself, while system wide preferences must be set editing a text file on the installation server.

To log on as an administrator, you can use the supplied account **demoadmin**, password **demo**, that will bring you to a home page like the following one:

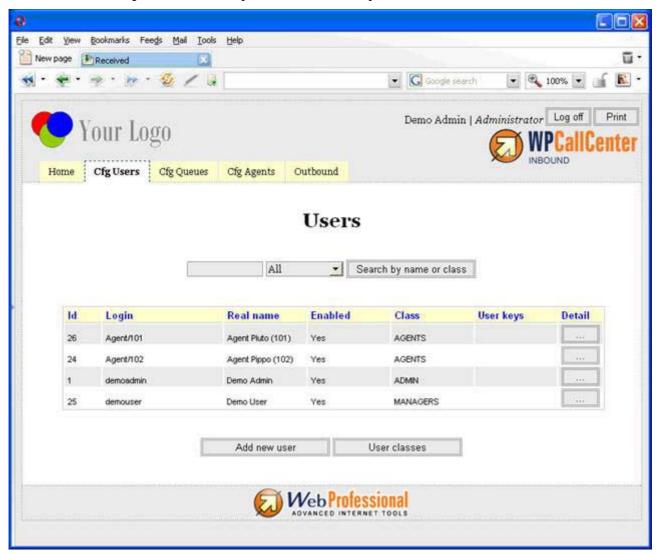




Configuring users

Users and classes can be added, modified and deleted right from WP CallCenter Inbound.

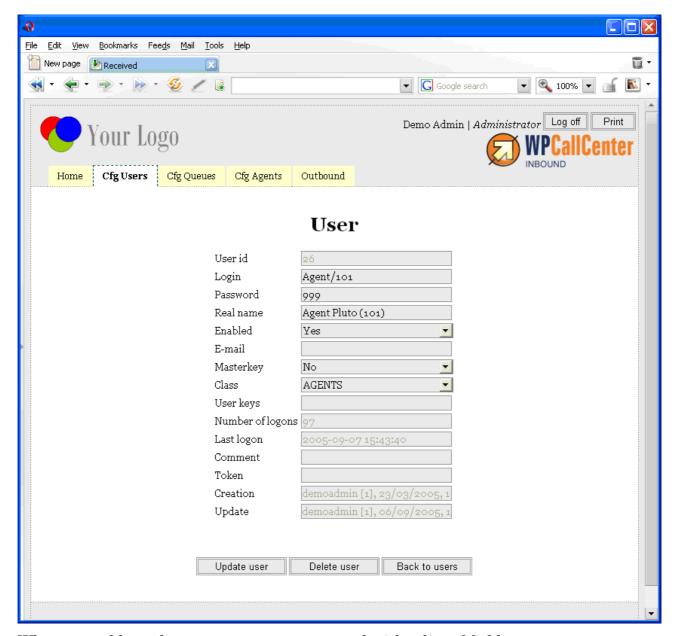
A list of users is presented and you can filter it by class or user name.



For each user, the login and full name are shown, together with the current class and any additional user keys. A user must be enabled in order to log on, so if you want to prevent somebody from logging on without deleting its user information, you can simply disable it.

The "Add new user" button lets you add new users while the "User classes" button leads to the class editor.





When you add or edit a user, you are presented with a list of fields to enter:

- *Userid* is a technical reference used internally. Read only.
- *Login* is the login string.
- *Password* is the password, shown in clear text.
- *Real name* is the name shown in the top part of the screen
- *Enabled* lets you temporarily disable somebody from using WP CallCenter Inbound.
- *E-mail* is the user's e-mail address (Optional).
- *Masterkey*: if set to Yes, all security key checks are bypassed. DO NOT SET UNLESS YOU KNOW WHAT YOU ARE DOING!



- *Class* is the current user class
- *User keys* are additional keys the user holds. Separate each key with a space. If a key is preceded by the minus sign, it means it's revoked even if the class grants it.
- *Number of logons* tells how many times the user logged on in WP CallCenter Inbound. Read only.
- *Comment* is an optional free comment.
- *Token* has no current use. Read only.
- *Creation* and *Update*: the user and date/time when the record was first created and then last updated. Read only.

IMPORTANT:

When you first log on to WP CallCenter Inbound, you <u>must</u> change the passwords to all default users. Failure to do so represents an important security breach!



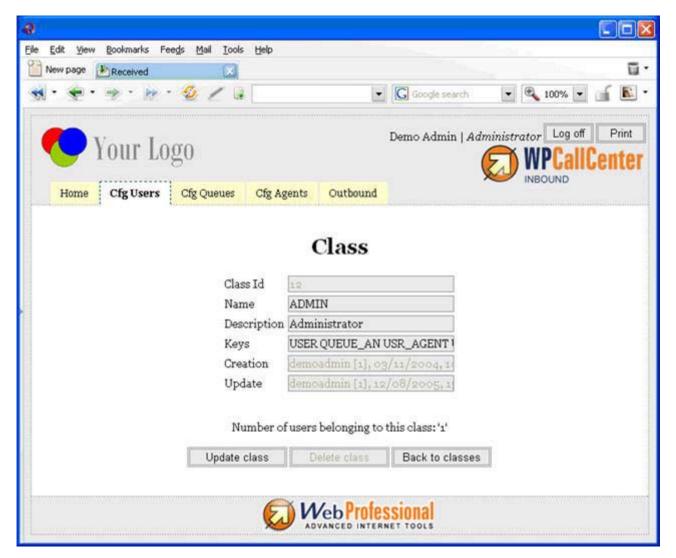
Editing user classes

User classes can be configured freely; you can create individual key rings with special privileges to best suit your needs.



Each class has a set of keys that can be freely edited in much the same way as users.





No class can be deleted as long as there is at least one user that is member of it.

The default classes should be enough to get most systems started:

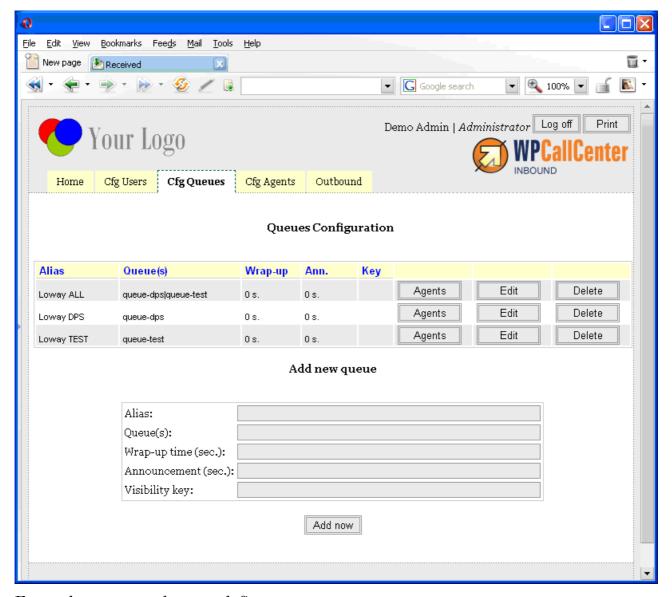
- ADMIN is for the system administrator only, and lets you do nearly everything, including system configuration;
- MANAGERS is for most WP CallCenter Inbound users, the ones that have to run the reports and monitor real-time activity;
- AGENTS is for individual agents logging on to their web page.

Configuring queues

A list of queues must be set before accessing WP CallCenter Inbound. Each queue can be made visible to only a specific set of users by adding a key – this can be useful if, for example, each queue has a manager viewing data for it, while only a CC manager sees data for all queues in the center.







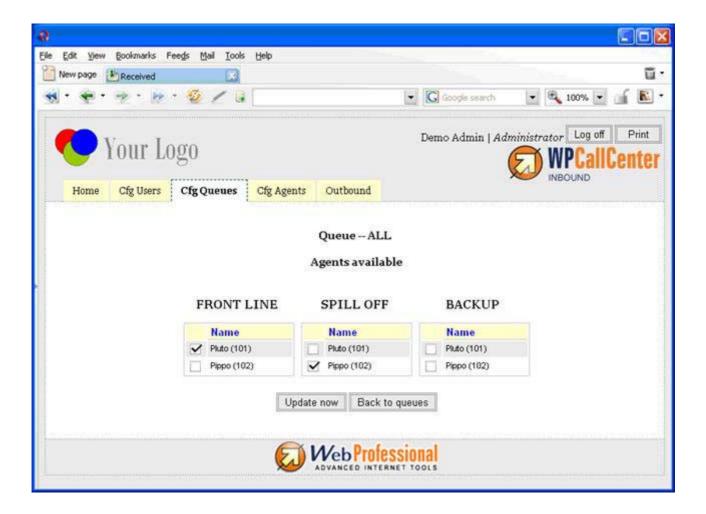
For each queue you have to define:

- An *Alias*, that is the name users will see in the queues combo box on the Home
- A set of Queues, that can be the name of an WP ePhone queue as seen from the Queue() command or a set of names separated by the pipe symbol, as in queue1 | queue2 | queue3. This lets you aggregate queues freely.
- An optional Wrap-up time, i.e. how many seconds an agent stays idle after hanging up;
- An optional *Announcement* duration, that lets you deduce the duration of the queue announcement that is played to the agent from the actual metrics;
- An optional *Visibility key*, that makes the queue visible only to users holding that key.



By clicking on the Agents button, you can define the position of each agent as a member of the service groups for that key. An agent cannot be a member of more than one group per each queue s/he is a member of.

It is of course perfectly legal for an agent defined not to be used in a specific queue.

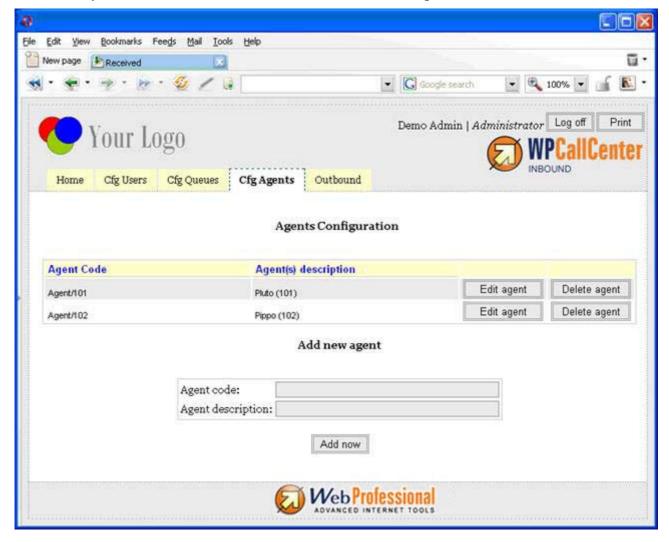




Configuring agents

Agents can be configured so that:

- They're decoded to their own name when they are found in reports
- They can be set as members of service levels for queues.



For each agent in use, enter:

- *Agent code* as the WP ePhone agent code, e.g. Agent/101;
- *Agent description* as the agent's own name.

If you want an agent to log on to their own page, you also have to create a user with the same name.



Configuring system preferences

System preferences can be edited by editing a text file called *configuration.properties* located in the WEB-INF directory of the WP CallCenter Inbound webapp.

This lets you set system-wide parameters and edit the logo and splash screen text for WP CallCenter Inbound.

Property name	Description	
default.queue	Internal ID (ex. 7, 49) of the default queue, leave blank for no default queue.	
default.queue_log_file	Default queue log file.	
realtime.max_bytes_agent	When the real-time page for an agent is computed, the queue_log is NOT read in its entirety but only the last 'n' bytes. In database storage mode, the number of seconds, starting from now and counting backwards, that will be queried for agent events.	
default.monitored_calls	The top level directory where monitored calls are held. All its subdirectories are explored recursively. Do NOT forget to add an ending slash.	
default.areacode_digits	How many digits to consider as a default area code	
default.start_hour	Preset start and end hours and number of days for the custom report.	
default.end_hour		
default.days		
liveclock.enable	If live clock is enabled, the system clock is synchronized with WP ePhone server system clock.	
default.max_realtime_age	How old a call can be included in real-time report	
sla.max_monitored_delay	The max delay and interval that will be shown in the	
sla.interval	TOS graph	
layout.logo	Your company logo (full or relative path) - shall be resized to be an image 200 x 72. The variable \$WEBAPP refers to the local webapp, as an alternative use the full http:// URL.	



layout.splash	HTML string displayed on the login page.
${\it sqlPreset.} i. {\it table}$	Sets the table name for preset 'i'
sqlPreset.i.f_time_id sqlPreset.i.f_call_id sqlPreset.i.f_queue sqlPreset.i.f_agent sqlPreset.i.f_verb sqlPreset.i.f_partition sqlPreset.i.f_data1 sqlPreset.i.f_data2 sqlPreset.i.f_data3 sqlPreset.i.f_data4	Defines the field used by the table in MySQL storage. See page 35 and following for complete information.

Configuring WP ePhone for WP CallCenter Inbound

Though WP CallCenter Inbound is designed to analyze queue_log data provided by any WP ePhone installation, the following guidelines help make the most out of it.

Configuring queues to report exit status

In the following example:

- all calls are monitored, i.e. saved to disk;
- if after 60 seconds on the queue the call is unanswered, the call is routed to voicemail and this event is reported correctly by WP CallCenter Inbound;
- there are two levels of agents: agents 302 and 303 will answer the queue (level 1); only if none of them is available the call is routed to agent 301 (level 2). If nobody is available, the queue keeps trying until timeout is reached.
- Agents can transfer the call to other extensions by pressing the "#" key;
- Agents terminate the current call by pressing the "*" key.

Extensions.conf

```
[q-my-sample]
; ...queue description....
exten => s,1,SetVar(MONITOR_FILENAME=/var/spool/calls/QSAMPLE-${UNIQUEID})
exten => s,2,Queue(q-sample|nt|||60)
exten => s,3,Playback(voicemail-invitation)
```





```
exten => s,4,VoiceMail,s2001
```

Queues.conf

```
[q-sample]
music = default
announce = q-sample-announce
strategy = roundrobin
timeout = 60
retry = 5
maxlen = 0
announce-frequency = 0
announce-holdtime = no
monitor-format = wav
monitor-join = yes
queue-youarenext = silence
queue-thankyou = q-sample-thankyou
member=>Agent/302,0
member=>Agent/303,0
member=>Agent/301,1
```

Make sure that you do not forget the explicit timeout when calling the Queue() command from *extensions.conf*, or queue timeouts will not be logged by WP ePhone and therefore not reported by WP CallCenter Inbound.

Configuring URLs to be launched by the agent real-time page

The URL should be embedded in the Queue() command as prescribed by WP ePhone

```
Exten => s,7,
Queue(myqueue|nt|http://mysite/app?uid={$UNIQUEID}&clid={$CALLERID}||60)
```

This command launches the queue "myqueue" and launches the webapp located at http://site/app passing the following parametrs:

- 1. *uid* is the WP ePhone internal unique call id
- 2. *clid* is the Caller*ID for the current call

Listening to calls using WP CallCenter Inbound

• *Make sure it is legal*This is not strictly a WP CallCenter Inbound issue, but before attempting to record all calls on a queue, you should consult a lawyer to make sure it is legal in your country. It would be probably fair enough to tell your operators their calls are being recorded and to add a voice message telling the customers their call will be recorded.



• *Tell WP ePhone to record all calls*To record all calls add something like this to *extensions.conf*:

```
exten => s,1,SetVar(MONITOR_FILENAME=/var/spool/calls/q/ QSAMPLE-
${UNIQUEID})
exten => s,2,Queue(q-sample|nt|||60)
```

This way all sound files are stored under /var/spool/calls/q/ with the name of the queue (QSAMPLE) followed by the call id.

• Tell WP CallCenter Inbound where to look for the calls
You should set up the WEB-INF/configuration.property file in WP CallCenter
Inbound like this:

```
default.monitored_calls=/var/spool/calls/q/
```

When looking for the recording of a call, WP CallCenter Inbound will explore all files contained in /var/spool/calls/q/ and any directories below for a file name containing the right call ID. It might find more than one file name and will display all of them. It is possible that sometimes WP ePhone fails at mixing together the two files (WP ePhone records separate files for the caller and the agent, and then tries to mix them together at the end of the call) so you will find two files named -in and -out instead.

- *Tell WP CallCenter Inbound you have the right to listen to the calls*Any user willing to listen to calls must hold the key CALLMONITOR. This is to make sure that only authorized personnel can listen to recorded calls. If you do not have this key, no sound files will be shown.
- Make sure WP CallCenter Inbound has the right to read saved calls
 You should make sure that the process running WP CallCenter Inbound (i.e. the
 servlet container, might be Tomcat, Jetty, or something else depending on your
 setup) has the rights to access the files where recorded calls are stored.
 If using a separate web server, it should not be able to access those files
 directly, as WP CallCenter Inbound will pipe out files only after enforcing
 security checks.
- Debug tip: see which files WP CallCenter Inbound sees
 There is a hidden transaction in WP CallCenter Inbound made to debug call
 listening. To launch it, logon as an administrator and type the transaction
 "qm_show_files.do" in the URL bar instead of the page name.
 You will be lead to a page showing the filenames WP CallCenter Inbound can
 read from the hard disk, whether the current user has the CALLMONITOR key
 and the search path as defined in default.monitored_calls.

Using AddQueueMember for dynamic agents

AddQueueMember is a command that lets you add dynamic agents to a queue. Its main advantage is that you can add channels, i.e. terminals, so you'll have most of the

Page 53 of 56



advantages of agents without the performance and stability problems that the agents module may cost in very large systems.

Its disadvantage is that it does not log the agent login/logoff to the queue_log, and so programs that analyze the queue log data like WP CallCenter Inbound will not see agents logging on and off. This is a major organizational problem in a real-world call center, where tracking agent logons and logoffs is vital to the smooth running of the operations.

The answer is to add a fake queue_log data for each logon and logoff. For WP CallCenter Inbound, it is important to avoid multiple logoff lines and to compute online permanence with logoffs.

To do the adding, you dial 422XX, where XX is your local extension; the same happens with 423XX to be logged off.

```
; Add Member - 422
exten => 422XX,1,Answer
exten => 422XX, 2, AddQueueMember (my-queue, SIP/${EXTEN:3})
exten => 422XX,3,System( echo
"${EPOCH}|${UNIQUEID}|NONE|SIP/${EXTEN:3}|AGENTLOGIN|-" >> queue log )
exten => 422XX,4,DBput(dynlogin/log Agent-${EXTEN:3}=${EPOCH})
exten => 422XX,5,Hangup
; Remove Member - 423
exten => 423XX,1,Answer
exten => 423XX,2,RemoveQueueMember(my-queue,SIP/${EXTEN:3})
exten => 423XX,3,DBget(ORGEPOCH=dynlogin/log Agent-${EXTEN:3})
exten \Rightarrow 423XX,4,Set(RV=$[${EPOCH} - ${ORGEPOCH}])
exten \Rightarrow 423XX,5,GotoIf(\{["\{RV\}" = "0"]?8:6\})
exten => 423XX, 6, System ( echo
"${EPOCH} | ${UNIQUEID} | NONE | SIP / ${EXTEN: 3} | AGENTLOGOFF | - | ${RV}" >> queue log )
exten => 423XX,7,DBdel(dynlogin/log Agent-${EXTEN:3})
exten => 423XX,8,Hangup
```

With this setup, we verified that the queue_log can be analyzed by WP CallCenter Inbound and the dynamic agent shows up fine (albeit with the name of a terminal, like SIP/23, instead of the usual Agent/23 string, but you can modify it in WP CallCenter Inbound itself).

This setup might even be used in a call center where agents are not actually used but queues connect straight to terminals to "fake" agent logon/logoff, in order to have such data available for reporting.



For more information...

To know more about WP CallCenter Inbound in your specific setting or inquire about commercial licences, please feel free to contact CP Software.

Appendix I: Default users

The following users are come preconfigured in the default database.

Login	Password	Explanation
demoadmin	demo	The sample admin user
demouser	demo	The sample CC manager
Agent/101	999	A sample agent
Agent/102	998	Another sample agent

Make sure you change their default passwords before letting users access WP CallCenter Inbound!



Appendix II: Security keys

The following security keys are defined:

KEY	MEANING
USER	Must be held by any valid user
USRADMIN	User can edit other users and classes
USR_AGENT	User can edit agents
USR_QUEUE	User can edit queues
REALTIME	User can see real-time stats
QUEUE_AN	User can run reports
AGREP	User can filter reports by agent
AGENT	User is an agent and sees agent page
CALLMONITOR	The user can listen to a recorded call



Appendix III: Glossary

Agent: a person working at the monitored call center and answering to calls. WP ePhone offers a way for agents not to be bound by physical telephone terminals but to log on to tell the system they are available.

Caller: a person calling the WP ePhone system

Composite queue: A virtual queue made of more than one physical queue. Useful for reporting all center activity at once.

Monitoring: in WP ePhone terminology, the act of recording to disk.

Queue: the call distribution object that let WP ePhone keep callers waiting and distributes them in the correct order to available agents. Each caller is processed on a first-come-first-server basis.