User Manual

Release 3.2

October 14, 2010

CONTENTS

CHAPTER

ONE

Getting started

Welcome to InstantVoice! This is a quick guide to configuring a two-brick distributed installation for inbound and outbound calls. Although we will stick to the basics, the guide gives you enough information to get started and grow. By the time you get through it, you will be able to:

- Configure additional servers for use in your InstantVoice cluster.
- Add and configure SIP trunks.
- Create new extensions, queues, and agents.
- Structure basic IVRs according to your needs.
- Route outbound calls appropriately.

If you need more information than is provided here, please visit http://www.instantvoice.net and check out our support options.

1.1 Installing the ACD Server

We will begin by installing the ACD server, from here on referred to simply as the *ACD*.

1.1.1 Prerequisites

Before you begin, you should have two InstantVoice compatible servers (from here on out referred to as *bricks*) ready for installation. This means:

1. They should have an i386 compatible processor.
2. Each brick needs one hard drive with at least 8 GB free for a new partition.
3. They must each have a functioning network interface connected to the same network.
4. Each one needs an optical drive to read installation media.

If configured properly, virtual machines will work just as well, but we recommend "bare metal" installations for production use.

To complete this guide you will also need at least one SIP IP phone (of the soft or hard variety) and a SIP DID provider to make and receive calls. This provider may take the form of an internet SIP service, a local SIP gateway, or even a SIP phone that can handle IP calls.

Finally, you should also have downloaded and burned the installation media to a CD or DVD. Once you have your boxes and installation media ready, continue with the guide.

1.1.2 Beginning the Installation

Note: Installation of the Media Server will take place in much the same way. Where applicable, this guide will highlight differences with a note such as this one.

Insert the installation media into the ACD's optical drive, and boot from it. If everything went according to plan, you should be presented with the welcome screen. When facing it, simply press `ENTER` to continue.

After a few moments, the installer will ask you what keyboard layout you use. Use the arrow keys to select the one that matches your current keyboard and press `ENTER` to continue.

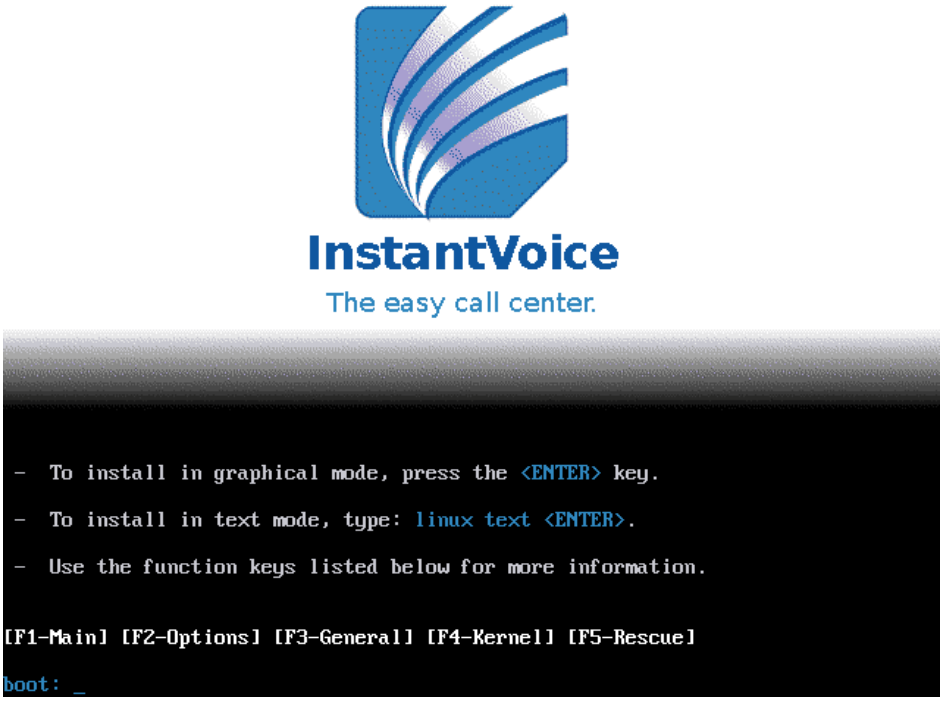

Figure 1.1: The welcome screen.

1.1.3 Disk Partitioning

Next, you will get to the disk partitioner. There is a great deal of freedom in choosing possible disk layouts, however most possible configurations go beyond the scope of this guide. The default setting for the first dropdown box is the following:

```
Use free space on selected drives and create default layout.
```

This will not erase any previous partitions on your drive. However, we recommend installing the system afresh by selecting:

```
Remove all partitions on selected drives and create defaut layout.
```

This will erase everything on the disk(s), but the added space and reduced complications will be welcome if this installation is to be used in production. The rest of this guide will assume you chose this last option. If you wish to only use a part of your disk for installing InstantVoice, separate at least 8 Gb for the task.

After partitioning your disk(s), click `Next` to continue.

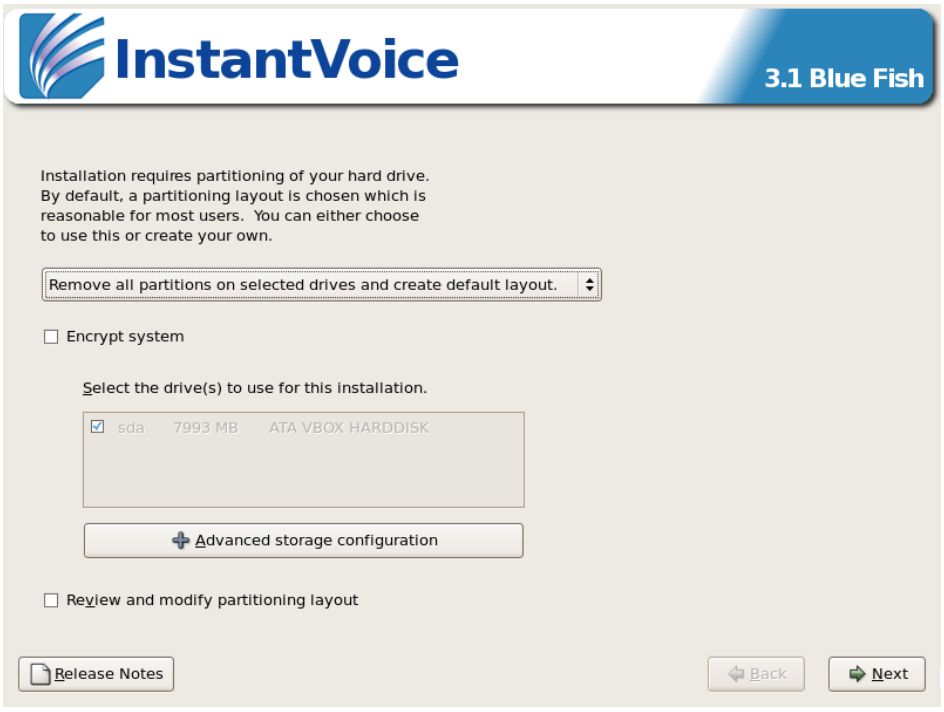

Figure 1.2: The disk partitioner.

1.1.4 Network Configuration

You should now be facing the network configuration screen. Here, you have the ability to configure the network as you see fit. However, for best results we recommend using this ACD's built-in name resolution service, so each brick will be configured with a static IP address. Make sure the first available network interface is selected, then click on `Edit`. You will see this interface's configuration dialog box.

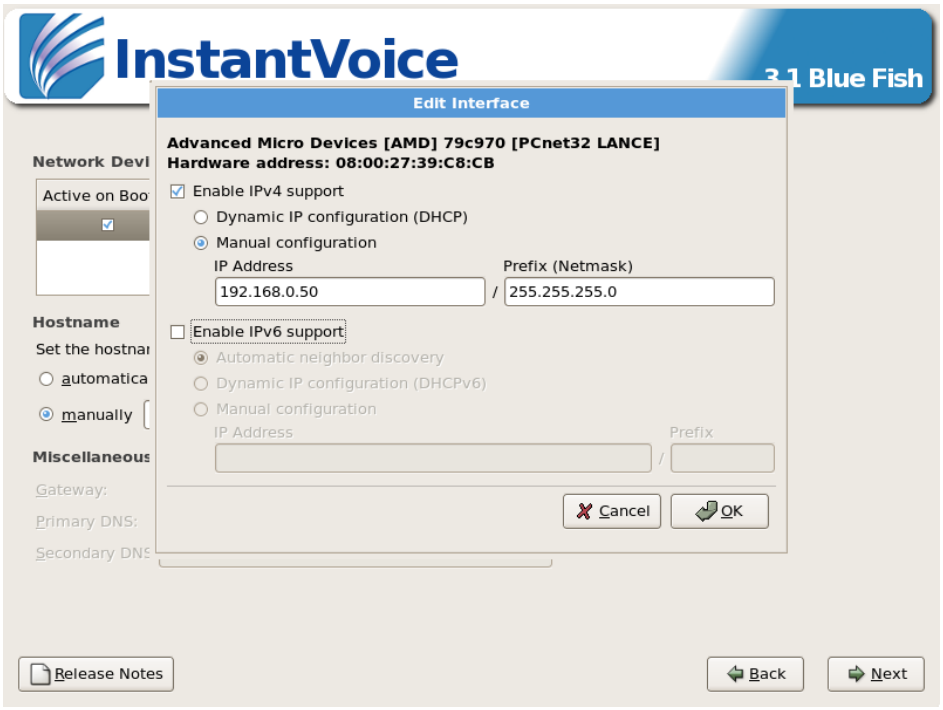

Figure 1.3: Configuring eth0.

For illustration purposes this guide will use the 192.168.0.0 IPv4 range, so from here on down the ACD's IP address will be referred to as *192.168.0.50*, and its netmask *255.255.255.0*. Nevertheless, you can enter a valid IP address and Netmask of your choice, taking note of the values for use later on during configuration. And unless your network uses IPv6, disable it here.

When done, click on `Ok` to close this window and continue with network configuration.

> **Warning:** If you don't configure each primary interface with a static IP address, you **must** configure corresponding hosts on your internal network's DHCP **and** DNS servers. However, this scenario falls outside the scope of this guide.

Next, configure a hostname for this machine. From the two radio-button options under `Hostname`, choose `manually`, and in the text box define an arbitrary hostname. Enter whatever name you prefer, just don't use spaces (when in doubt, stick to letters and numbers). Also make a note of this value, you will need it later on. For this guide, we will refer to this machine's hostname as *instantvoice-acd*.

Under the `Miscellaneous` tab, configure the default gateway for this machine. For illustration purposes we will use a value of *192.168.0.1*, but you will need to use a correct value for your network.

Finally, for `Primary DNS` enter *127.0.0.1* regardless of you local network configuration. This will ensure that name resolution works properly for the cluster. If you don't do this, the distributed environment won't work unless you configure your own network's DNS server accordingly.

Leave `Secondary DNS` empty, and click on `Next` to continue, bringing you to the regional configuration screen.

Note: When installing the Media Server, make sure to use an IP address that is different from the one in use by the ACD. Also, take care to enter the ACD's IP address in the `Primary DNS` field (**not** 127.0.0.1), which in our example would be *192.168.0.50*.

1.1.5 Last Steps

At this point you should be seeing the regional configuration screen. Chose the closest city to your location using the dropdown or the map. Regional options will be set accordingly. We also recommend checking the `System clock uses UTC` checkbox. After choosing your location, click on `Next` to continue. You should now see the root password configuration screen. Enter an arbitrary password in the first box, and again in the second. For security, it should not be based on names or any other dictionary word; we recommend mixing upper and lower case letters, numbers, and other symbols. When done configuring the root password, click `Next` to continue.

> **Warning:** Do not lose the root password! It is essential for future maintenance of the machine, and if lost little can be done in the event of a serious technical problem.

From here until the final reboot, installation will proceed automatically. Sit back for a few minutes while package installation proceeds. When it is done the server will reboot.

Take care to remove or disable installation media during this stage, otherwise you might be presented with the welcome screen again. If this happens, simply remove the CD or DVD and reset the machine manually to continue.

If all went well during first boot, you should see the server login screen on the terminal. This concludes installation, and we're now ready to configure the ACD.

1.2 Configuring the ACD Server

To start configuring the ACD server, point you web browser to the IP address of the ACD which you specified during installation, in this case, http://192.168.0.50. You will be presented with the InstantVoice homepage.

Figure 1.4: Package installation.

1.2.1 Logging in for the first time

To log in, first locate the login box on the upper left-hand corner of the home page. You will log in with the user *localadmin* and password *localadmin*, so fill in the two fields appropriately and then click on `Login`.

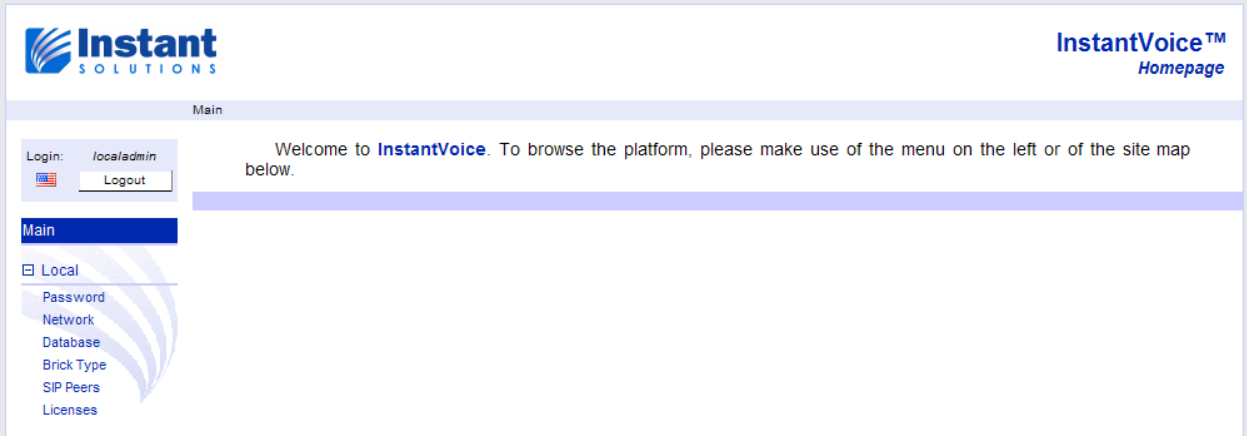

Figure 1.5: Login page.

Note: Each InstantVoice installation comes with two pre-configured web users, *localadmin* and *admin*. The first one, *localadmin*, is used to configure any and all features specific to each brick in the cluster, such as network addresses or the brick's type. It is accessible on all bricks. The second

one, *admin*, is only accessible on the ACD itself (or on Standalone bricks which are also ACDs), and is used to access and configure cluster-wide features, such as SIP extensions. More on this later on, under Configuring the Cluster.

If your login was successful, you should see the `Local` expandable menu tab on the left column of the home page, just below the `Main` menu. If not, please check you typed the login and password correctly, and try again.

1.2.2 Changing the localadmin password

After logging in, click on the `Local` parent menu to expand it. The first thing to do is change the default *localadmin* password to something more secure. To do so, click on the `Password` sub-menu, and type in the default password on the first field ("localadmin"). On the next one, type in a new password and confirm it on the last field. Click on `Save` to make the change, and you should see a confirmation banner if it was successful.

In the future, when logging in as *localadmin* you will need to use the password you just created.

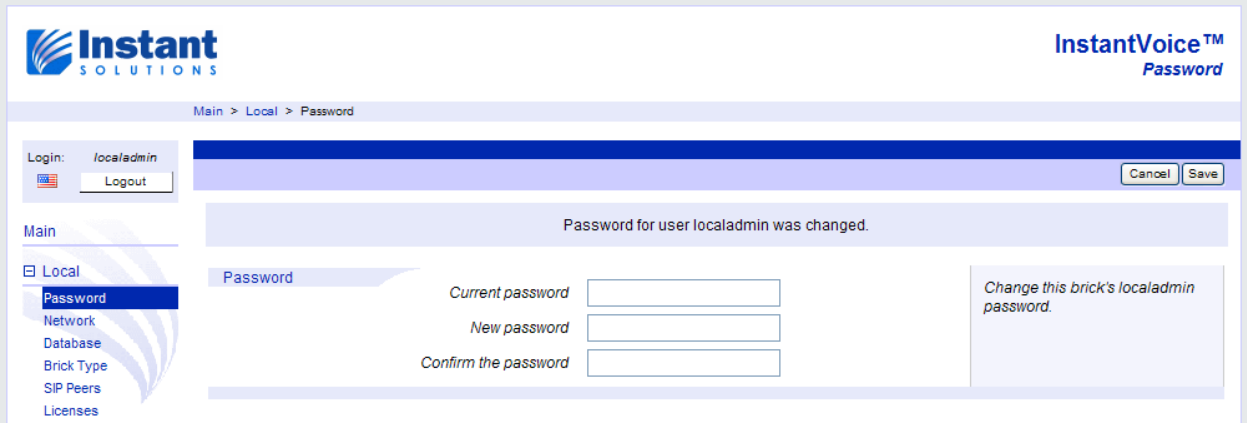

Figure 1.6: Password successfully changed.

1.2.3 ACD Network configuration

Next, you will check the ACD's network configuration. Click on the `Network` sub-menu on the left-hand side of the page to continue.

Everything should be set to the values you specified during installation. However, if you need to change anything, from this point on you will do it here. When and if a change is made, click on `Save`, after which the web server will be restarted. Note that if you change the IP address of the server, you will have to type in a new URL pointing to the server's new network location in the address bar of your browser.

1.2.4 ACD Brick Type

The location of the database need not be changed for the ACD, so we will skip database configuration here. However, we still need to tell the system that this will be the cluster's ACD: click on the `Brick Type` menu on the left.

All installations produce *Standalone* bricks by default. To change this, select *ACD* from the `Type` dropdown. Since for the purposes of the guide there will only be one ACD, make sure to check the `Active` box. We highly recommend you enter an IP address for the `Recursive DNS` field, otherwise external hostnames will not be resolveable by the cluster. The address should point to a working DNS in a sub-network that is accessible by the ACD. You can use either your own network's DNS or a publicly available one.

Warning: A `Recursive DNS` that can resolve internet domains is required for license activation. More information on license activation can be found at the end of this topic, under the section Installing More Agent Licenses.

This concludes the initial configuration of the ACD. Next, you will install and configure the cluster's Media Server.

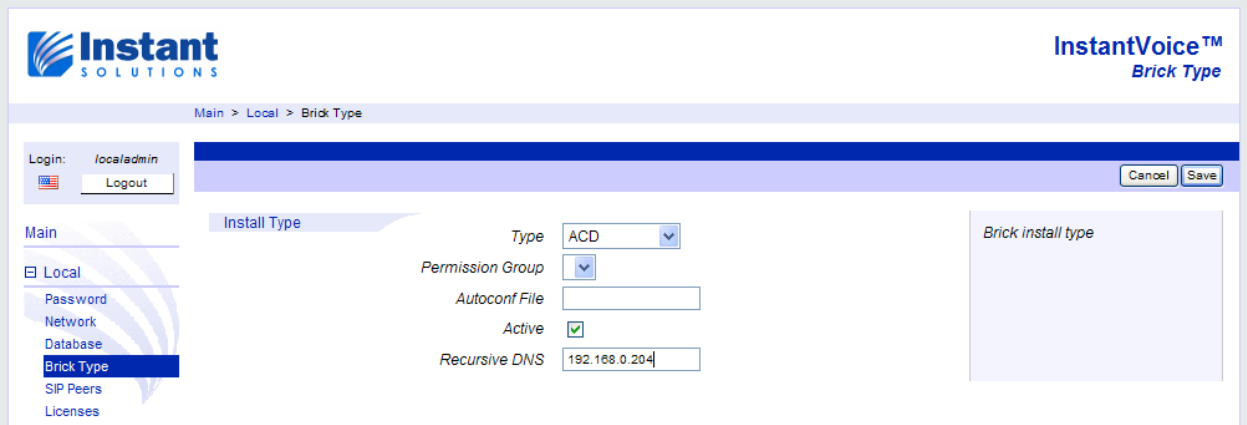

Figure 1.7: ACD brick type.

1.3 Installing and Configuring the Media Server

To install the Media Server, please follow the section Installing the ACD Server, but take note of the following exceptions to the network configuration.

- The IP address of the Media Server should be different to the address used for the ACD. In this guide, we will use the value of *192.168.0.51*.
- This brick's hostname should also be different. In this case, we will use *instantvoice-ms*.

- Finally, make sure you specify the ACD's IP address (*192.168.0.50* in this example) as the Media Server's Primary DNS.

All other installation steps can be followed exactly as detailed in Installing the ACD Server.

1.3.1 Beginning Media Server Configuration

To configure the Media Server, point you web browser to the IP address of the you specified during installation, in this case, http://192.168.0.51. As when configuring the ACD Server, you will be presented with the InstantVoice home page. Locate the login box on the upper left-hand corner, and login with the user *localadmin*, using password *localadmin*.

Warning: Make sure you are logging in to the Media Server, not the ACD! To minimize confusion, we recommended opening a different browser tab (or window) for each brick.

As with the ACD, start by changing the localadmin password and checking network configuration. Follow the same procedure for these as specified in the sub-sections Changing the localadmin password and ACD Network Configuration, but nothing further.

1.3.2 Database Configuration

Next, you need to tell the Media Server to use the ACD's database. To do so, click on the `Database` sub-menu item in the left column of the page. In the `Hostname` field, enter the hostname of the ACD as you specified in during ACD installation (in this example, *instantvoice-acd*). Leave the `User name` and `Password` field values as they are.

Note: You could enter the IP address of the ACD instead of the hostname, but if in the future you change the IP address of the ACD, you would have to remember to change it on this page too.

Before saving changes, click on the `Test connection` button. If everything is configured correctly, you should get a success message. If the connection test was not successful, please review network configuration in the ACD and the Media Server and make sure:

- The bricks have different IP addresses and hostnames.
- The Media Server's DNS is configured to point to the ACD's IP address.
- Both the ACD and Media Server are either on the same subnetwork or on mutually accessible subnetworks.

Only click on `Save` when the connection test is successful.

1.3.3 Creating Media Server Trunks

In order to actually make and receive calls, you will need to configure trunks on the Media Server. The first one will be of type *Internal*, for local calls between extensions, the second of type *SIP*,

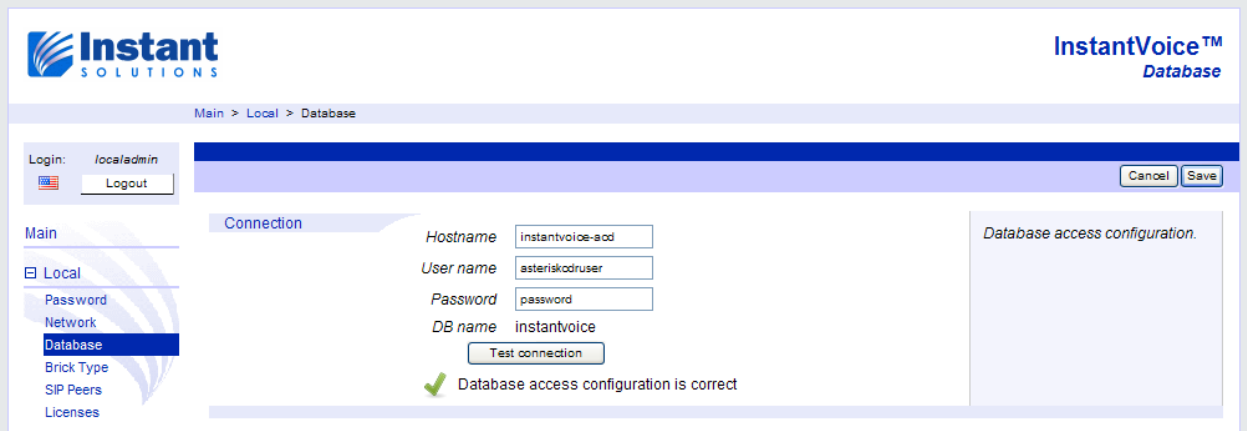

Figure 1.8: Successful database connection test.

necessary for external calls.

Note: Each Media Server needs at least one configured trunk in order to be of any use. On distributed installations, even calls from one extension to another will fail without an *Internal* trunk! But first, you need change the Media Server's brick type from Standalone to Media Server. Click on the `Brick Type` sub-menu on the left of the page. From the `Type` dropdown, chose *Media Server*. After you do so, the `ACD Hostname` field should automatically display the hostname of the currently active ACD server, in this case *instantvoice-acd*. If not, before you go on go back to the ACD Brick Type sub-section and make sure the ACD is marked as active.

Next, you will create two trunks for use with inbound and outbound calls. To add the internal trunk to the Media Server, click on the `Add trunk` button in the `Brick Type` page. You will be presented with the trunk configuration dialog. Choose a name and enter it in the `Name` field. You can name it anything you like, but for this guide we will call it *Internal*. From the `Type` dropdown, choose *Internal*, and then click on `Add`. The *Internal* trunk should now be listed in the `Trunks` section.

At this point please click `Save` on the upper right-hand corner of the page before going on, otherwise, you might lose any unsaved data.

After saving, click on `Add trunk` again, but this time select *SIP* from the `Type` dropdown. In the `SIP Peer` dropdown you should see two peers, one for the ACD and one for the Media Server. However, you can use neither! You need a new peer, one that can handle inbound and outbound calls. To create it, you will first need configuration information from your SIP provider. Typically, the following data is required:

- The internet address where the provider hosts its SIP service.
- The service's port.
- A user and password to register your endpoint with.

Once you have this information, click on the `Add SIP Peer` hyperlink next to the `SIP Peer`

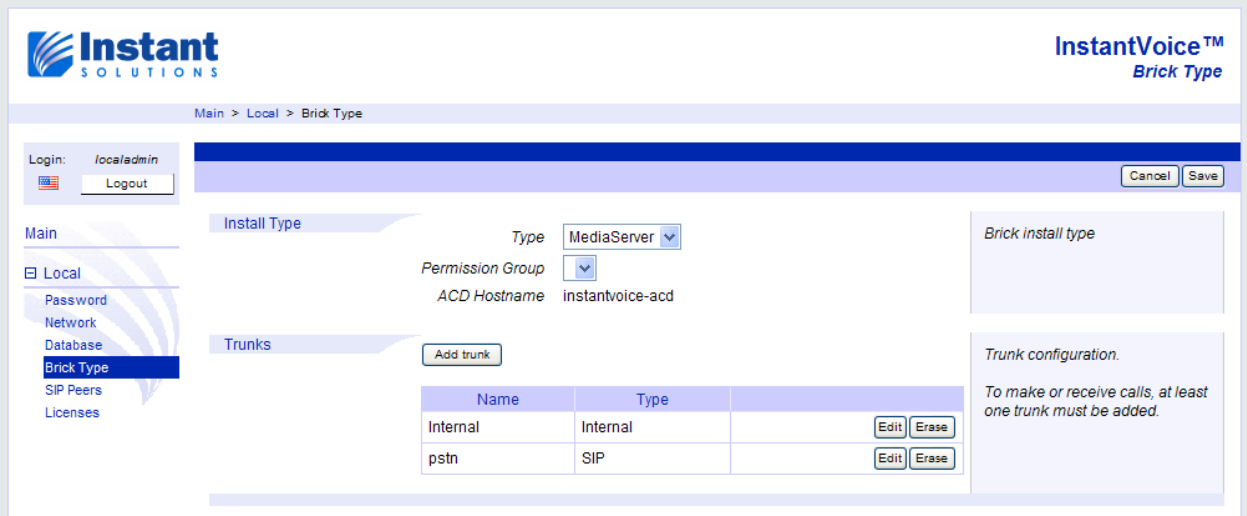

Figure 1.9: Media Server brick type.

dropdown. You will be taken away from the `Brick Type` page, so make sure changes up to this point are saved!

When facing the peer configuration screen, first enter an obligatory name for it on the `Name` field. It can be anything you like, but for this example we will use *pstn*. The `Description` field is optional, but the `User` field must contain the user name your provider gave you. If a password is required, enter it once on `Password` field, and again on the `Confirm` field.

Next, open the `Network` tab. First, uncheck the `Dynamic IP` box so that you can specify the provider's internet host name or IP address on the `Host` field, and its port on the `Port` field. If unsure about the port, use the default value of *5060*.

Other provider-specific configuration may be required. When done, click on `Submit`. If any errors were encountered (such as an invalid peer name), they will be displayed on screen, in which case you should correct them accordingly and try again.

Now that you have a SIP peer configured for you provider, you are ready to create its trunk. To do so, go back to the `Brick Type` page by clicking on the appropriate menu item on the left-hand column of the page. Once there, click on `Add Trunk`, and select *SIP* from the `Type` dropdown. Now click on the `SIP Peer` dropdown, and select the peer you just created on the previous step; in this example, we will select the one named *pstn*. Name the peer arbitrarily using the `Name` field. You can use the same name you used for the peer; we will use *pstn* again.

Finally, make sure to check the `Register` checkbox if your provider requires you to register, click `Add` to create the trunk, and finally `Save` to record all changes.

Note: You may be asking yourself why peer registration is configured for a trunk and not for a peer. The reason is that a SIP peer (such as the one created previously) may be reused by any number of trunks, on any number of Media Servers. However, the SIP protocol specifies only one registration at a time per IP address and port. Thus, SIP endpoint registration per trunk provides a mechanism to control exactly which Media Server will receive inbound calls from a given provider.

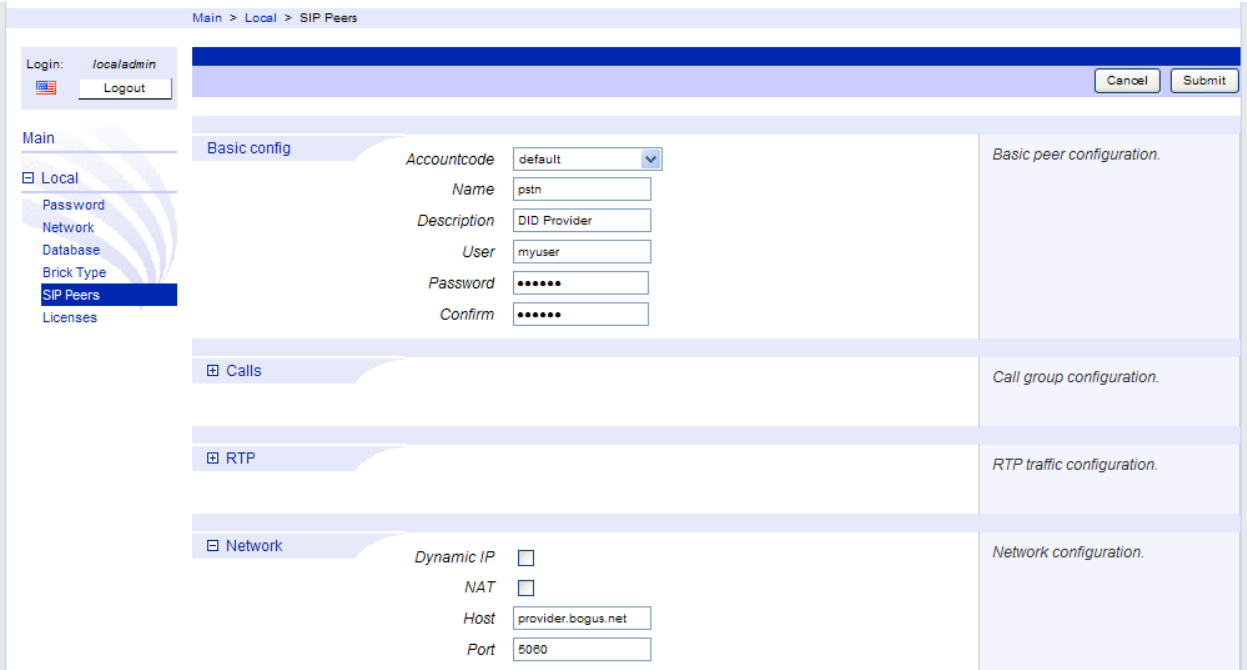

Figure 1.10: SIP peer configuration.

This concludes local configuration for the Media Server. Next, you will configure cluster-wide features such as extensions, agents, and queues.

1.4 Configuring the Cluster

In this section we will configure everything necessary to get basic telephony and call center features up and running. Specifically, our objectives in this section are:

- To be able to receive an external call through an IVR, send it to a queue, and have it sent to an idle agent.
- To be able to make calls from one extension to another and to an outside line.

To get started, connect to the ACD server using your web browser. If you followed this guide's examples exactly, type http://192.168.0.50 into the address bar, otherwise use the ACD IP address you configured previously. This time, you will log in using the *admin* user. Type "admin" into the `Login` field, and "admin" again as the password. Then, click on the `Login` button. You should then see a full complement of menu items on the left, starting with *My Account* and ending with *Reports*. If you don't see them, or if login otherwise failed, make sure that:

- You are trying to log in with user *admin*, not *localadmin*.
- You are accessing the ACD server, not the Media server.

If you logged in successfully, go on to the next section.

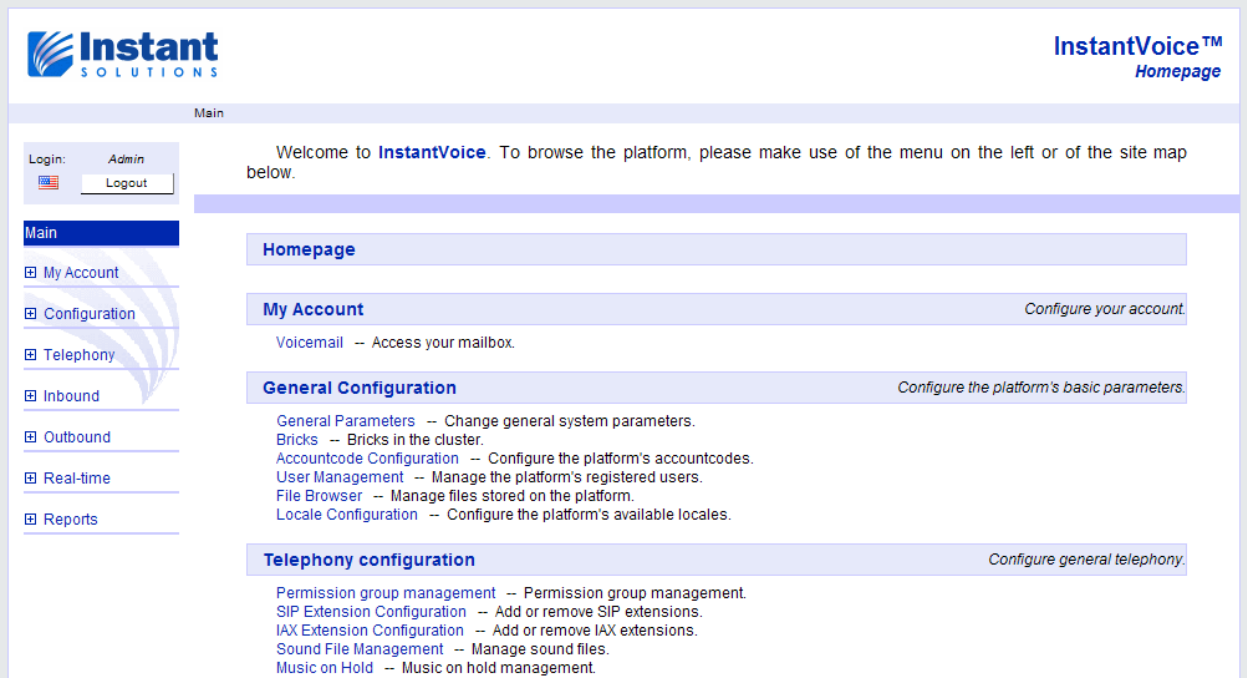

Figure 1.11: Admin logged in.

1.4.1 Creating SIP Extensions

We will start by creating a few SIP extensions. Locate the `Telephony` menu section on the left of the page, and click to expand it. Find the `SIP Extensions` menu item under it and click it. You should then be facing an empty SIP extension page.

Find and click on the `Add` button on the upper left hand corner, and you will see a page similar to SIP peer configuration. The principle is the same, with some added options and conventions. For starters, it is possible to add a range of SIP extensions at one go. That's precisely what we're going to do, so check the *Enable range* box at the top of the `Basic Config` tab.

We'll be using the *default* accountcode, so leave that dropdown alone. On the `Extension` field, type in the number of the first extension. Only numbers, such as *1001*, are accepted. On the next field, `Final Extension`, enter the number of the last extension, for example, *1006*. Finally, check the box labeled `Password is extension number`. This will create each extension with a password equal to its number. For instance, a SIP phone registering at extension *1003* will need to use password *1003*. Alternatively, do not check that box and simply choose one password that will be used on all created extensions.

When done, click on `Submit`. You should then see a list of extensions which you just created.

Note: All SIP extension users are prefixed with that extension's accountcode. This is so that if desired, a single InstantVoice installation can have the same extension number in use by two different accounts. It is implicit in this setup that one accountcode's extensions cannot dial to the ones on another accountcode.

To test your work, at this point you must register a SIP phone using one of the extensions that were

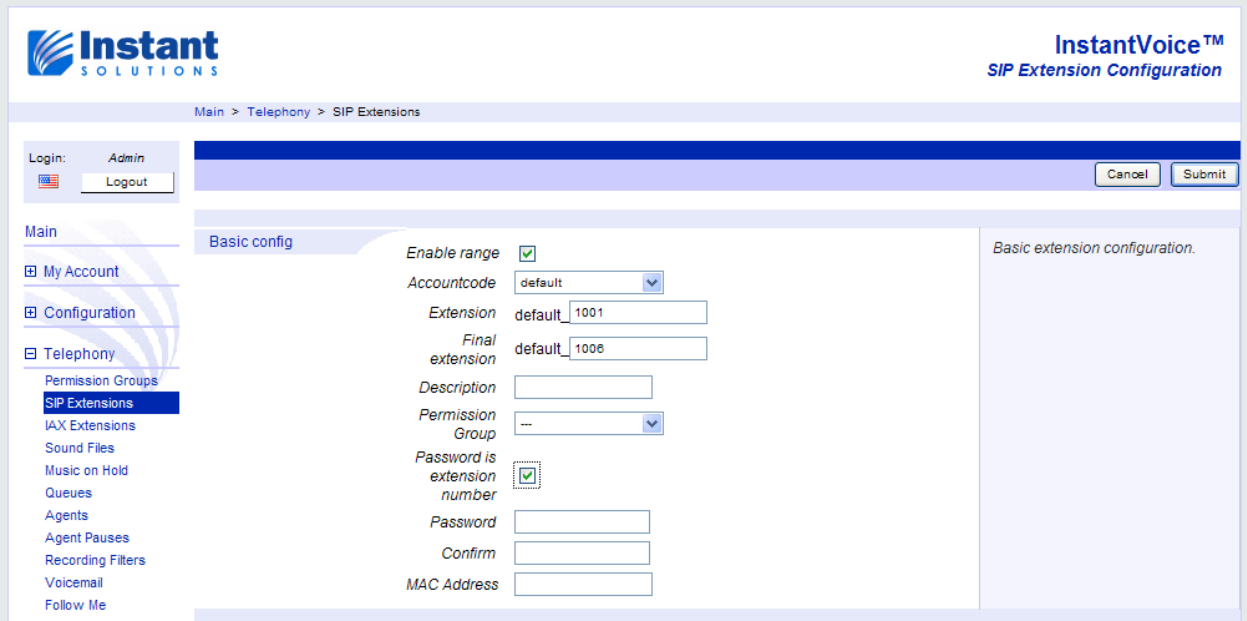

Figure 1.12: Creating a range of SIP extensions.

just created. Configure your phone to register at the ACD's address, in our case *192.168.0.50*. The authentication user string is the chosen extension number prefixed with its accountcode: in our case, *default_1001*, and the password, *1001*. If everything was properly configured, your phone should be able to register successfully with the ACD (even though making calls won't be possible yet).

Warning: In distributed installations, SIP phones should always register with the ACD, not the Media Server!

1.4.2 Creating a Queue

Once you have created a few extensions and successfully registered at least one SIP phone, it is time to create a call queue and an agent, in order to be able to receive inbound calls.

You will first create a basic call queue. On the main menu on the left, click on `Queues`, under `Telephony`. There are none as of yet, so click on the `Add Queue` button.

On the edit page, start by entering a queue number on the `Number` field. Much as an extension, this number will uniquely identify a queue, and as such it is mandatory. We will use the number *10* for futher reference in this guide. The `Description` field is optional, but useful for identification later on: we will use simply *Queue 10*.There are many other queue configuration options, but they fall beyond the scope of this section. However, default values will probably meet most of your initial needs.

When done, click on `Save`.

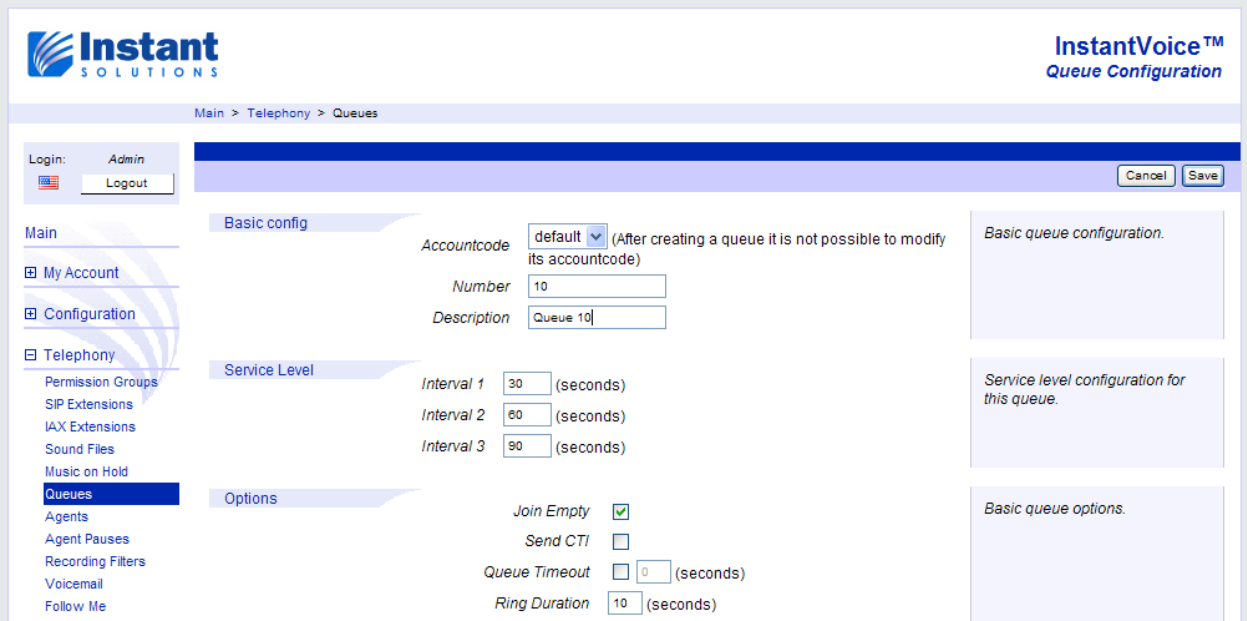

Figure 1.13: Adding a queue.

1.4.3 Creating an Agent

After creating *Queue 10*, you will be able to create an agent. Under the `Telephony` menu, click on `Agents`.

Creating an agent is a three-step process: first you create the agent, then its available lines, and finally you assign it to one or more queues. To start, click on `Add agent`.

On the first step, in the appropriate fields give the agent a `Number` (which must not be the same as any extension number), a `Name`, and optionally assign it a numeric `Password`. This password will need to be dialed in at the extension for critical agent operations such as logging in or logging off, so we recommend keeping it short. For illustration purposes we will use *2001* as the agent number, *Agent 2001* as its name, and *123* as this agent's password. When done, click on `Next`.

On this screen you can control how many simultaneous calls an agent can make or take. For instance, if you want an agent to be able to receive two calls simultaneously, add two inbound lines. If you want it to be able to receive a call on one line and make a call on another one, add one inbound and another outbound line. If you don't care if the agent makes or receives calls on a given line, create *bidirectional* lines.

For this guide this agent will only need one *bidirectional* line, so simply click on `Add line` to add it, and then on `Next` to continue.

Finally, in the last step we will assign this agent to the queue you created previously. In our case, this is *Queue 10*. Check the box next to the queue, and click `Save` to finish creating this agent. It should then show up on the agent list.

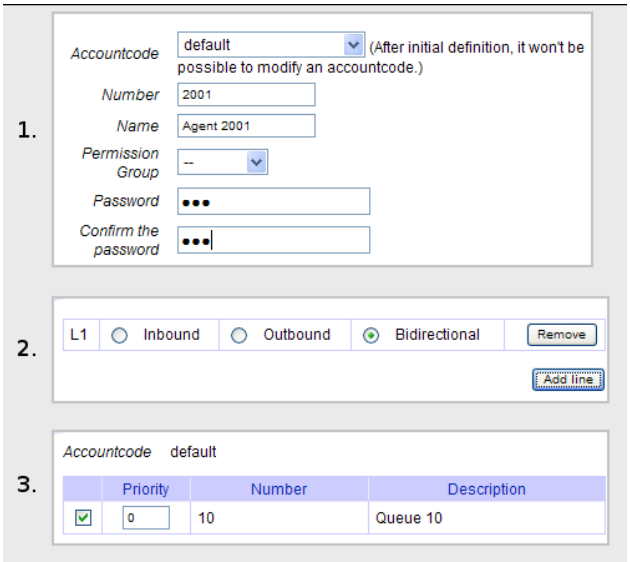

Figure 1.14: Creating an agent.

1.4.4 Logging in with an Agent

The first thing to do after creating the agent is to actually log it in at one of the SIP extensions you created previously. Assuming you're going to log in agent *2001* at extension *1001*, simply pick up the phone at that extension and dial:

```
*02001
```

Where `*0` is the global login command, and `2001` is the agent number. If you configured a password, the IVR will prompt you for it. Assuming the password for this agent is "123", dial the following when prompted:

```
123#
```

After this, if everything went according to play you should hear an "Agent logged in" message. This means that InstantVoice now recognizes *Agent 2001* as being logged in at extension *1001* and expecting calls from *Queue 10*.

Note: Other important agent commands are:

- `*1` to pause.
- `*2` to unpause.
- `*3` to log off.

As long as you issue them at the extension the agent is logged in, you do not need to specify the agent number after these commands. Nevertheless, you will be prompted for the agent password if one was configured.

You can also pause, unpause, and log off agents at the realtime web interface. To use it, enter the `Agents` page under the `Real-time` menu.

1.4.5 Building a Simple IVR

The basic flow for inbound calls in InstantVoice is the following:

```
Trunk --> SLN --> IVR --> Queue --> Agent
```

In more detail:

1. **Trunk**: One of the configured trunks receives a call from the PSTN.
2. **SLN**: Every PSTN call has a *Subscriber Line Number*, or SLN, that identifies what number was dialed by the calling party. InstantVoice uses the call's SLN to forward it to a specific IVR.
3. **IVR**: Based on a series of factors the IVR in turn decides what Queue (if any) to forward the call to.
4. **Queue**: The Queue holds the call until an Agent becomes available.
5. **Agent**: The call is finally answered by an Agent.

You have already configured a trunk, a queue, and an agent, but before your queue can actually receive calls, you will need to create a simple IVR that forwards calls into it and an associated SLN that tells InstantVoice to route a given call through. Let's begin with the IVR. Click and expand the `Inbound` menu section on the left of the page, then click on `VIVRE` to access the *Visual IVR Editor* main page. Click on `Add`, and fill in the `IVR Name` field with a name for your IVR (in this guide, we will refer to it as "MyIVR"). Ignore the other fields and click on `Submit` to save, after which you should now see an entry for your IVR on the page.

At this point you will create a very simple IVR that prompts the calling party to dial "1". If that number is dialed the call will be forwarded to the queue you created previously. Otherwise, the caller will hear a message and the call will be hung up. The basic mechanics are simple: for each interactive menu you want the calling party to hear, you must create a *module*. To jump from one module to another, you will test user input and forward the call accordingly.

Click on the previously created entry's `Edit IVR` button to start creating the actual structure of your menus. A new window will pop up, in which you will see two buttons, `Save IVR` and `New Module`. Click on `New Module` now to generate this IVR's first module. It will appear immediately to the right. You can click on its title bar and drag the module anywhere you like on the work space. Try it!

Newly created modules are always named "NewModule" by default. Click on its `Edit` button and select the "NewModule" string to change it. For illustration purposes we suggest using "Main"

because this is where new calls will be directed, but you can use any name you like, as long as you don't use spaces, punctuation, or other special characters.

Modules are composed of lines. Click on `New Line` to add a new one. The first thing to do is welcome the calling party, so from the dropdown on the left choose the `tts` command and enter a phrase such as "Welcome to InstantVoice, please dial 1 to talk to an agent." Next, create a new line and select the `waitdigits` command. This command tells the IVR engine to expect input from the caller in the form of digits. Leave its options on its default values, but since the caller will only enter one digit, on the `Max digits` column enter "1".

Note: `tts` stands for *Text To Speech*: text you enter on this command will be synthesized and played back to the caller. Licenses for the better quality AdvancedTTS engine, which supports a range of different languages, must be acquired separately. Although the `tts` command only supports English, it can be used freely.

To test what the calling party dialed in response to the prompt, create one more line and select the `ifvar` command . Under the `Variable Name` column, type in the variable name that was used on the line above ("input", by default). Under `Value`, type in "1" to check if the caller dialed that number. The last two dropdown boxes will be used to select the destination of the call depending on if a "1" was entered or not, but leave them blank for now.

On a new line, select the `tts` command once more. If the call gets here, the caller did not dial a 1, so enter a message such as "Invalid option, goodbye!" Finally, on the last line of this module select the `hangup` command to end the call.

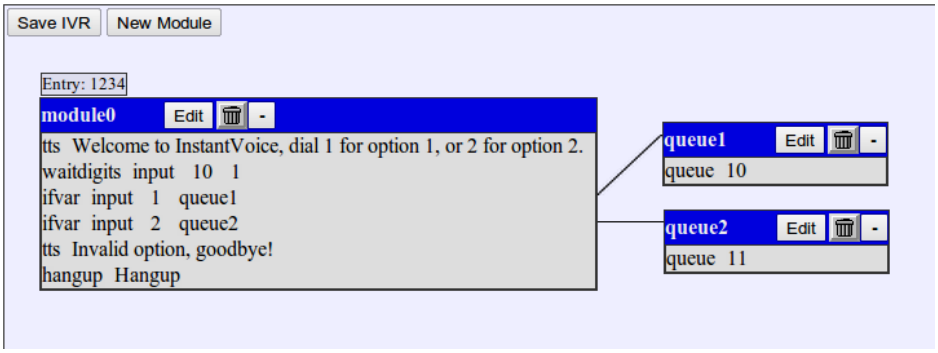

Figure 1.15: A simple IVR.

At this point, click on the `Ok` button to close this module temporarily, and then click on `New Module` at the top of the workspace to create a new one. This module will be where the call is forwarded if the caller dials a "1", so name it "Queue". It will only contain one line; create it and select the `queue` command from the first dropdown. Under the `Queue` column you should see a dropdown listing the queue you created previously. Select it, and click on `Ok` to close this module.

You can now tell the "Main" module to forward the call to the "Queue" module if the caller dials "1". To do so, on the "Main" module, click on `Edit`. On the dropdown under the `Destination True` column of the `ifvar` line, select the "Queue" module you just created, and leave the `Destination False` dropdown empty: this tells the IVR to jump to the "Queue" module if a "1" was dialed, or to continue to the next line if not.

Click on Ok to close the "Main" module once again. Notice that now the "Main" and "Queue" modules are joined together by a line, indicating their relationship. When you have more than 2 modules, we recommend dragging-and-dropping them so that the IVR's structure is made visually clear. Each module's position will be saved and restored the next time you open the IVR!

At this point, make sure to click on Save IVR. After you do so, close this window and return to the main web interface.

> **Warning:** Make sure to always click on Save IVR before closing the window, otherwise your work may be lost!

You still need to create an SLN that points to this IVR, otherwise InstantVoice cannot forward inbound calls to it! Under the Inbound menu on the left, select the SLNs page. Click on Add, and in the SLN field on the edit page, enter the actual DID number you provider sends with each call, for example, "5552345". From the IVR dropdown, select the IVR created previously ("MyIVR", in our example), and from IVR Module select that IVR's "Main" module. Finally, check the Active box and click on Submit.

This concludes your initial inbound call setup. Try it out by calling your DID. Just make sure to have at least one agent logged in (as described in the Logging in with an Agent section), otherwise no extensions will be rung!

1.4.6 Routing Outbound Calls

If you want to make outbound calls through the trunk you configured previously, or even calls from one extension to another, you will need to create appropriate routes. Simply put, routes are how you tell InstantVoice where each call goes, how much they cost, and who can use them. For maximum flexibility, routes are defined on a priority system based on the number that is dialed, and who is dialing.

Let's say you would like to have it so that any 4-digit number dialed at an extension makes a call to the extension at that number, and all 8-digit numbers are calls to be completed by a SIP trunk. Start by opening the Routing page under the Outbound menu and then clicking on Add route.

First, enter a suggestive name on the Name field, such as "Internal calls". On the Expressions tab, you will enter a pattern that will be used to match numbers as they are dialed on an extension. Roughly speaking, route expressions are read by InstantVoice in this way: "Does the dialed number start with the numbers in this expression?" This implies that you can simply enter a straightforward number, such as:

```
1001
```

If you dial "1001", this route will be matched. However, it will also match "10010", "10012345", or any other number that **starts with** "1001". To tell InstantVoice to match only 1001, you need to add a "$" to the end of the expression, like so:

```
1001$
```

However, this is still not ideal. You want to match any 4-digit number, not just 1001, and entering 9,999 expressions would be too time-consuming. To do so, you would use a *character class*. For instance, to match any single digit from 0 up to 9, you would write:

```
[0-9]
```

So to match any 4-digit number, create an expression exactly as the following, and click on `Add`:

```
[0-9][0-9][0-9][0-9]$
```

You only need this single expression for internal calls, so click on `Next` to continue.

On the next step, to specify the trunk through will calls that match this route will be routed to, click on `Add Trunk Group` followed by `Add Trunk`. Since this route will be used for extension-to-extension calls, select the "Internal" trunk that was created previously (see the section entitled Creating Media Server Trunks). Click on `Next` when done.

On the third and final step you will configure the extensions that can use this route. From the top dropdown chose `SIP Extensions`. You should then see a list of all configured SIP extensions on the left-hand box. To give them all access to this route, click on `Select all`, followed by `Allow`. This will transport to the right-hand box, which lists all entities permitted to access this route. Click on `Save` when done.

At this point it is already possible to make calls between extensions, for instance by dialing "1002" at extension 1001. To make external calls, however, you will need to add another route. Follow the same procedure specified above for the "Internal calls" route, except:

- On the first step, name the route "External calls" and use the following expression for 8-digit numbers:

```
[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]$
```

- On the second step, select the "pstn" trunk instead of the "internal" one.

After saving, you should be able to make outbound calls. If not, please check your outbound trunk's peer configuration and make sure everything is set properly according to your provider's specifications.

Note: If you're having routing problems, another important step in troubleshooting is to check if the number being dialed is being matched by an existing route. The main routing page provides a simple mechanism to do so. On the `Filters` dropdown on the upper right-hand corner, select the type of entity that is trying to make the call, such as "SIP Extension". On the next dropdown, choose the actual entity, such as extension "1001". Finally, on the input box to the right, enter the

exact number that will be dialed and click on `Filter`. If no routes are shown, the selected entity and number to dial do not match any routes.

This concludes basic cluster configuration, which should give you a solid base to build on.

1.5 Installing More Agent Licenses

Installing agent licenses is very simple. Once you have received an email with your license codes after ordering from http://www.instantvoice.net, access your ACD ou Standalone server's web interface with the *localadmin* account (see the section entitled Logging in for the first time), and enter the `Licenses` page under the `Local` menu. Once you have done so click on the `Add License` button on the right. On the text field that shows up, paste the license code (the one that starts with "IV-") and click on `Register`.

Warning: On distributed installations, licenses only need to be activated on the ACD server!

As noted on the dialog box, the server needs to be able to access the internet for license registration to succeed. Permanent access is not required, however: you may disable internet access once your licenses are registered.

CHAPTER

TWO

Integration

The integration module is a set of tools which will allow your custom software to interact with the InstantVoice IP telephony platform. Currently we offer basic agent functionality and call events.

2.1 Pieces Description

2.1.1 IVWS

IVWS (InstantVoice Web Service) is a SOAP, WSDL defined, web service which processes all requests made by client applications to our plaftorm. Through it, you will be able to perform all basic agent operations: login, pause, unpause and logoff agents. Generate, transfer and hangup calls.

2.1.2 IVACD

IVACD (InstantVoice Automatic Call Distributor) is the central piece of our platform. For integration purposes, what you need to know is that this piece will be the one to send you the call events NEWCALL and HANGUP.

2.1.3 IVClient

IVClient is a COM object which wraps our SOAP methods and handles the reception of the call events plus a tray icon.

2.2 Server Configuration

There is only one parameter to configure on the server side and it is optional:

```
Family: instantvoice.conf
Group: ivws
Name: port
Default value: 7789
```

It describes the port in which your IVWS will listen on your active ACD brick.

Note: If you change this it is necessary to manually restart the IVWS service on your server.

2.3 IVWS Access Keys

The first parameter of every method of our web service is an access key, and these keys must be given permission to call the desired methods.

Note: A key is an alphanumeric string with a maximum length of 20 characters.

You can manage you keys on our web administration interface in the `Configuration` menu section, item `Web Service`.

2.4 Installing IVClient

Note: If you plan to use the web service directly, you can skip directly to Web Service Methods Description.

2.4.1 Installing

Assuming you already have an InstantVoice PBX up and running, all you need to do is install IVClient on each seat of your call center. In order to do this, simply double-click the setup file and go through it's steps.

Note: The COM object is registered for you during the installation process, you can unregister/register it in `Start Menu/IVClient/ Unregister COM` and `Register COM`.

2.4.2 Instantiating

IVClient's progid is "Instantvoice.Integration". The clsid is "{F9DD083D-1A0A-4B67-80FA-5565ECBF51C0}" but you probably don't need this for anything. After instantiating, make sure to call the method StartTray, more about this later.

Some code examples:

```
# Python
import win32com.client
```

```
ivclient = win32com.client.Dispatch("Instantvoice.Integration")
ivclient.StartTray()
```

```
' Visual Basic 6
Set ivclient = CreateObject("Instantvoice.Integration")
ivclient.StartTray()
```

```
// Delphi 7
uses ComObj;

var ivclient : Variant;
begin
    ivclient := CreateOLEObject('Instantvoice.Integration');
    ivclient.StartTray;
end;
```

```
# Perl 5
use Win32::OLE;

my $ivclient = Win32::OLE->new('Instantvoice.Integration');
$ivclient->StartTray();
```

```
# Ruby
require 'win32ole'

ivclient = WIN32OLE.new('Python.Teste')
ivclient.StartTray()
```

2.5 Client Configuration

If you right click the icon and select `config`, a window will open with the following fields:

Field	Description
IP Address	This must be the ip address of your active ACD.
Port	If you modified the web service's port as described in the Server Configuration, inform the same port here.
Key	Here you must insert the desired key created in IVWS Access Keys .
Local Port	In this field you inform which local port you want our client listening for call events.
Debug	This will make IVClient write all error messages on the text file %ProgramFiles%\InstantVoice\IVClient\ivlog.txt .

Note: IVClient automaticaly reloads all configuration changes.

2.6 COM Methods Description

These are the methods you will be able to use through our COM interface. All methods return 0 on success. For more information, see Return Codes.

2.6.1 StartTray

This method takes no parameters and starts IVClients's tray icon. This must be the first thing done after you instantiate the COM object on your application.

2.6.2 AgentLogin

```
AgentLogin(accountcode,username,password,interface,cti_addr)
```

This method will login the given agent on you InstantVoice platform, allowing him to make and receive calls.

Method Argument	Description
Account-code	The agent's accountcode
Username	The agent's useraname, which is always <accountcode>_<agentNumber>. Ex.: default_1001, instant_304512, etc.
Password	The agent's password.
Interface	The telephony interface where this agent will answer his calls in the format <TECH>/<name>. Ex.: SIP/default_0001, IAX2/default_230, etc.
cti_addr	This is the <ip_address>:<port> to where you want the call events for this agent sent. If you're using IVClient, you want it sent to that same pc.

```
Example:
AgentLogin("default","default_1001","1337","SIP/default_0001","192.168.0.101:8888")
```

2.6.3 AgentLogoff

```
AgentLogoff(accountcode,username,password)
```

This method will logoff the given agent from you InstantVoice platform.

Method Argument	Description
Accountcode	The agent's accountcode
Username	The agent's useraname, which is always <accountcode>_<agentNumber>. Ex.: default_1001, instant_304512, etc.
Password	The agent's password.

```
Example:
AgentLogoff("default","default_1001","1337")
```

2.6.4 AgentPause

```
AgentPause(accountcode,username,password,reason)
```

This method pauses the given agent, preventing him from receiving calls.

Method Argument	Description
Accountcode	The agent's accountcode
Username	The agent's useraname, which is always <accountcode>_<agentNumber>. Ex.: default_1001, instant_304512, etc.
Password	The agent's password.
Reason	The pause number, as defined in Telephony/Agent Pauses.

```
Example:
AgentPause("default","default_1001","1337","1")
```

2.6.5 AgentUnpause

```
AgentUnpause(accountcode,username,password)
```

This method unpauses the given agent, allowing him to receive calls again.

Method Argument	Description
Accountcode	The agent's accountcode
Username	The agent's useraname, which is always <accountcode>_<agentNumber>. Ex.: default_1001, instant_304512, etc.
Password	The agent's password.

```
Example:
AgentUnpause("default","default_1001","1337")
```

2.6.6 BlindTransfer

```
BlindTransfer(accountcode,username,dest_ext)
```

This method will redirect all calls currently taken by the agent to the desired dest_ext. This destination can be anything, a queue, another agent or even an external number.

Method Argument	Description
Accountcode	The agent's accountcode
Username	The agent's useraname, which is always <accountcode>_<agentNumber>. Ex.: default_1001, instant_304512, etc.
Password	The agent's password.
Dest_ext	The destination extension you want the calls to be redirected.

```
Example:
BlindTransfer("default","default_1001","6100")
```

2.6.7 NewCall

```
NewCall(accountcode,username,dest_ext)
```

This method starts a new call to the desired dest_ext and returns it to the agent.

Method Argument	Description
Accountcode	The agent's accountcode
Username	The agent's useraname, which is always <accountcode>_<agentNumber>. Ex.: default_1001, instant_304512, etc.
Password	The agent's password.
Dest_ext	The destination extension you want to call.

```
Example:
NewCall("default","default_1001","6100")
```

2.6.8 Hangup

```
Hangup(accountcode,uniqueid)
```

This method terminates a given call.

Method Argument	Description
Accountcode Uniqueid	The agent's accountcode The desired call uniqueid. You have received this on a NEWCALL event through PollEvents.

```
Example:
Hangup("default","1286208779.1.30")
```

2.6.9 PollEvents

This method takes no parameters and will return you a string in JSON format with all events that arrived after the last method call. You should call this at least on a 10 second interval.

```
[{
    "direction": "OUTBOUND",
    "callerid": "1234",
    "type": "NEWCALL",
    "uniqueid": "1282779516.1.6",
    "agent": "default_1001"
},{
    "direction": "INBOUND",
    "callerid": "2198765456",
    "type": "NEWCALL",
    "uniqueid": "1282781304.2.5",
    "agent": "default_1001"
},{
    "direction": "",
    "callerid": "2198765456",
    "type": "HANGUP",
    "uniqueid": "1282781304.2.5",
    "agent": "default_1001"
},{
    ...
}]
```

Field	Description
Direction	This can be either INBOUND or OUTBOUND, it indicates the call's direction. Only for NEWCALL events.
Callerid	This is whatever comes as callerid for inbound calls and what was dialed by the agent on outbound calls.
Type	Indicates the type of the event. Currently we have NEWCALL, HANGUP, LOGIN, LOGOUT, PAUSE and UNPAUSE.
Uniqueid	The system's uniqueid for that call. It is useful to match the NEWCALL events with the corresponding HANGUP events.
Agent	Each event is related to a single agent informed here.

Note: You can find JSON libraries to most modern programming languages on http://www.json.org/.

2.6.10 StopTray

This method takes no parameters. It closes IVClients's tray icon and all connections to InstantVoice server from this instance.

2.7 Web Service Methods Description

Note: If you are using IVClient, jump to Return Codes.

You can bypass IVClient entirely and access directly our web service methods. They use the same Return Codes and run on you ACD or STANDALONE brick.

Note: If you want to see the complete WSDL simply access http://acd_hostname:7789/?wsdl or whatever port you configured.

2.7.1 AgentLogin

This method will login the given agent on you InstantVoice platform, allowing him to make and receive calls.

Method Argument	Description
Key	Your web service access key.
Accountcode	The agent's accountcode
Username	The agent's username, which is always <accountcode>_<agentNumber>. Ex.: default_1001, instant_304512, etc.
Passwd	The agent's password.
Interface	The telephony interface where this agent will answer his calls in the format <TECH>/<name>. Ex.: SIP/default_0001, IAX2/default_230, etc.
cti	This is the <ip_address>:<port> to where you want the call events for this agent sent. If you're using IVClient, you want it sent to that same pc.

Example:

```
<soapenv:Envelope
 xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:ivws="Ivws.Ivws">
    <soapenv:Header/>
    <soapenv:Body>
        <ivws:AgentLogin>
            <ivws:key>f054bbd2f5ebab9</ivws:key>
            <ivws:accountcode>default</ivws:accountcode>
            <ivws:username>default_1001</ivws:username>
            <ivws:passwd>1337</ivws:passwd>
            <ivws:interface>SIP/default_0001</ivws:interface>
            <ivws:cti>192.168.0.101:8888</ivws:cti>
        </ivws:AgentLogin>
    </soapenv:Body>
</soapenv:Envelope>
```

2.7.2 AgentLogoff

This method will logoff the given agent from you InstantVoice platform.

Method Argument	Description
Key	Your web service access key.
Accountcode	The agent's accountcode
Username	The agent's useraname, which is always <accountcode>_<agentNumber>. Ex.: default_1001, instant_304512, etc.
Password	The agent's password.

Example:

```
<soapenv:Envelope
 xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:ivws="Ivws.Ivws">
  <soapenv:Header/>
  <soapenv:Body>
     <ivws:AgentLogoff>
        <ivws:key>f054bbd2f5ebab9</ivws:key>
        <ivws:accountcode>default</ivws:accountcode>
        <ivws:username>default_1001</ivws:username>
        <ivws:password>1337</ivws:password>
     </ivws:AgentLogoff>
  </soapenv:Body>
</soapenv:Envelope>
```

2.7.3 AgentPause

This method pauses the given agent, preventing him from receiving calls.

Method Argument	Description
Key	Your web service access key.
Accountcode	The agent's accountcode
Username	The agent's useraname, which is always <accountcode>_<agentNumber>. Ex.: default_1001, instant_304512, etc.
Password	The agent's password.
Reason	The pause number, as defined in Telephony/Agent Pauses.

Example:

```
<soapenv:Envelope
 xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:ivws="Ivws.Ivws">
  <soapenv:Header/>
  <soapenv:Body>
     <ivws:AgentPause>
        <ivws:key>f054bbd2f5ebab9</ivws:key>
        <ivws:accountcode>default</ivws:accountcode>
        <ivws:username>default_1001</ivws:username>
        <ivws:password>1337</ivws:password>
        <ivws:reason>1</ivws:reason>
     </ivws:AgentPause>
  </soapenv:Body>
</soapenv:Envelope>
```

2.7.4 AgentUnpause

This method unpauses the given agent, allowing him to receive calls again.

Method Argument	Description
Key	Your web service access key.
Accountcode	The agent's accountcode
Username	The agent's useraname, which is always <accountcode>_<agentNumber>. Ex.: default_1001, instant_304512, etc.
Password	The agent's password.

Example:

```
<soapenv:Envelope
 xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
```

```
 xmlns:ivws="Ivws.Ivws">
  <soapenv:Header/>
  <soapenv:Body>
     <ivws:AgentUnpause>
        <ivws:key>f054bbd2f5ebab9</ivws:key>
        <ivws:accountcode>default</ivws:accountcode>
        <ivws:username>default_1001</ivws:username>
        <ivws:password>1337</ivws:password>
     </ivws:AgentUnpause>
  </soapenv:Body>
</soapenv:Envelope>
```

2.7.5 BlindTransfer

This method will redirect all calls currently taken by the agent to the desired dest_ext. This destination can be anything, a queue, another agent or even an external number.

Method Argument	Description
Key	Your web service access key.
Accountcode	The agent's accountcode
Username	The agent's useraname, which is always <accountcode>_<agentNumber>. Ex.: default_1001, instant_304512, etc.
Password	The agent's password.
Dest_ext	The destination extension you want the calls to be redirected.

Example:

```
<soapenv:Envelope
 xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:ivws="Ivws.Ivws">
  <soapenv:Header/>
  <soapenv:Body>
     <ivws:BlindTransfer>
        <ivws:key>f054bbd2f5ebab9</ivws:key>
        <ivws:accountcode>default</ivws:accountcode>
        <ivws:username>default_1001</ivws:username>
        <ivws:dest_ext>6100</ivws:dest_ext>
     </ivws:BlindTransfer>
  </soapenv:Body>
</soapenv:Envelope>
```

2.7.6 NewCall

This method starts a new call to the desired dest_ext and returns it to the agent.

Method Argument	Description
Key	Your web service access key.
Accountcode	The agent's accountcode
Username	The agent's useraname, which is always <accountcode>_<agentNumber>. Ex.: default_1001, instant_304512, etc.
Password	The agent's password.
Dest_ext	The destination extension you want to call.

Example:

```
<soapenv:Envelope
 xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:ivws="Ivws.Ivws">
  <soapenv:Header/>
  <soapenv:Body>
     <ivws:NewCall>
        <ivws:key>f054bbd2f5ebab9</ivws:key>
        <ivws:accountcode>default</ivws:accountcode>
        <ivws:username>default_1001</ivws:username>
        <ivws:dest_ext>6100</ivws:dest_ext>
     </ivws:NewCall>
  </soapenv:Body>
</soapenv:Envelope>
```

2.7.7 Hangup

This method terminates a given call.

Method Argument	Description
Key	Your web service access key.
Accountcode	The agent's accountcode
Uniqueid	The desired call uniqueid. You have received this on a NEWCALL call events, see Receiving Call Events.

Example:

```
<soapenv:Envelope
 xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:ivws="Ivws.Ivws">
  <soapenv:Header/>
```

```
  <soapenv:Body>
     <ivws:Hangup>
        <ivws:key>f054bbd2f5ebab9</ivws:key>
        <ivws:accountcode>default</ivws:accountcode>
        <ivws:uniqueid>1286208779.1.30</ivws:uniqueid>
     </ivws:Hangup>
  </soapenv:Body>
</soapenv:Envelope>
```

2.7.8 Receiving Call Events

Our platform will send call events to the IP and port informed in the *cti* field of the agent's login action. Our server makes an HTTP POST to that address sending events in JSON format. We always send them one by one.

```
Example:
event =
{
    "direction": "OUTBOUND",
    "callerid": "1234",
    "type": "NEWCALL",
    "uniqueid": "1282779516.1.6",
    "agent": "default_1001"
}
```

Field	Description
Direction	This can be either INBOUND or OUTBOUND, it indicates the call's direction. Only for NEWCALL events.
Callerid	This is whatever comes as callerid for inbound calls and what was dialed by the agent on outbound calls.
Type	Indicates the type of the event. Currently we have NEWCALL, HANGUP, LOGIN, LOGOUT, PAUSE and UNPAUSE.
Uniqueid	The system's uniqueid for that call. It is useful to match the NEWCALL events with the corresponding HANGUP events.
Agent	Each event is related to a single agent informed here.

Field	Description
Direction	This can be either INBOUND or OUTBOUND, it indicates the call's direction. On Hangups it comes empty.
Callerid	This is whatever comes as callerid for inbound calls and what was dialed by the agent on outbound calls.
Type	Indicates the type of the event. Currently we have NEWCALL and HANGUP only.
Uniqueid	The system's uniqueid for that call. It is useful to match the NEWCALL events with the corresponding HANGUP events.
Agent	Each event is related to a single agent informed here.

2.8 Return Codes

We always return 0 on success. These are the possible return codes when something goes wrong with a request.

Code	String	Description
1	RESPONSE_MISSING_ARGS	There are mandatory arguments which were not filled.
2	RESPONSE_NO_MEMORY	The server has run out of memory.
3	RESPONSE_NOT_FOUND	The target of the request was not found. It could mean that the accountcode or the agent doesn't exist, for example.
4	RESPONSE_DUPLICATE	The server interprets this request as a duplicate from a previous request and will not act on it.
5	RESPONSE_INVALID	The request doesn't make sense to the server.
6	RESPONSE_NOTEMPTY	This request requires the target to be empty, but currently it is not.
7	RESPONSE_ITEM_FULL	The target cannot accept any more requests of this kind at this moment.
8	RESPONSE_ITEM_EMPTY	The target is currently empty, this request cannot be performed on an empty target.
9	RESPONSE_UNAUTHORIZED	The request doesn't contain the correct password for the desired target.
10	RESPONSE_NOT_ALLOWED	The server is not allowed to perform the requested operation.
11	RESPONSE_DISABLED	The target is disabled, the server cannot perform this request.
1000	RESPONSE_UNDEFINED	The server cannot perform any request because the IVACD is currently disabled.
9999	RESPONSE_INTERNAL_ERROR	An unexpected behaviour caused the request not to be completed.
-1	DAC_OFFLINE	IVWS could not communicate with IVACD.
-2	PERMISSION_DENIED	The provided key doesn't have permission to perform this request.

CHAPTER

THREE

Automated Dialing System

The ADS module will allow you to create dialing campaigns and automate the calling process.

3.1 Recipients List Format

Our software accepts two file formats for those lists. One is a proprietary billing software format called Siscob which we won't cover in this documentation. The other is a simple CSV (comma separated values) file.

3.1.1 CSV files

The first line of the file must be the field names, it will be ignored by the ads. The first column must be the number to be dialed, this number will go through the same routing any regular outbound call does. In the following columns you can place whatever information you need. You can acess these inside any VIVRE IVR using the name given to the column.

```
Example:
TelNumber,FullName,CustomerCode
2132458767,Henry Case,0023
1140637865,Peter Riviera,0156
5198776500,Sally Shears,3278
2144786045,Willis Corto,0009
...
```

3.2 Adding a Recipients List

Point your web browser to the address of you ACD or STANDALONE brick and login with your admin account. Click on the ADS menu and then choose Recipients

Click on the "Add list" button and you will be taken to the recipients list addition form with the following fields:

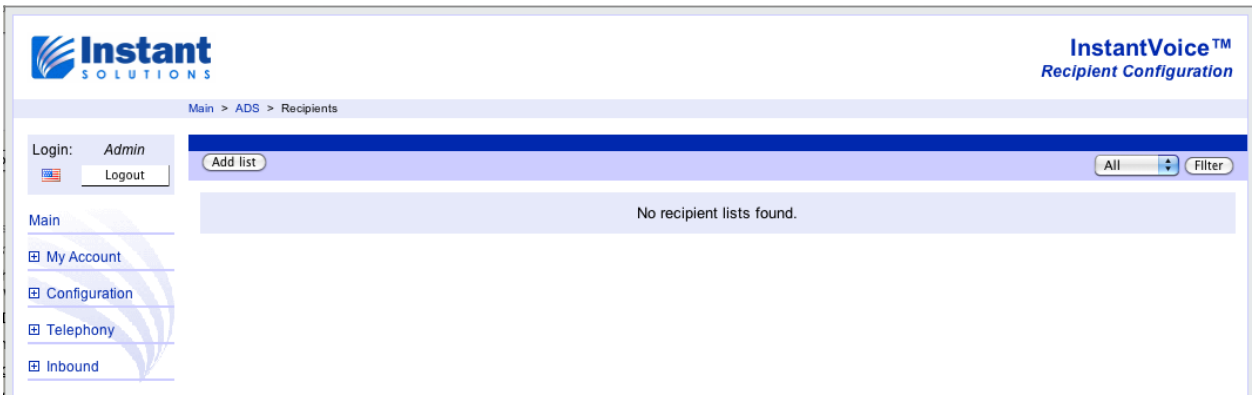

Figure 3.1: Empty recipients list window.

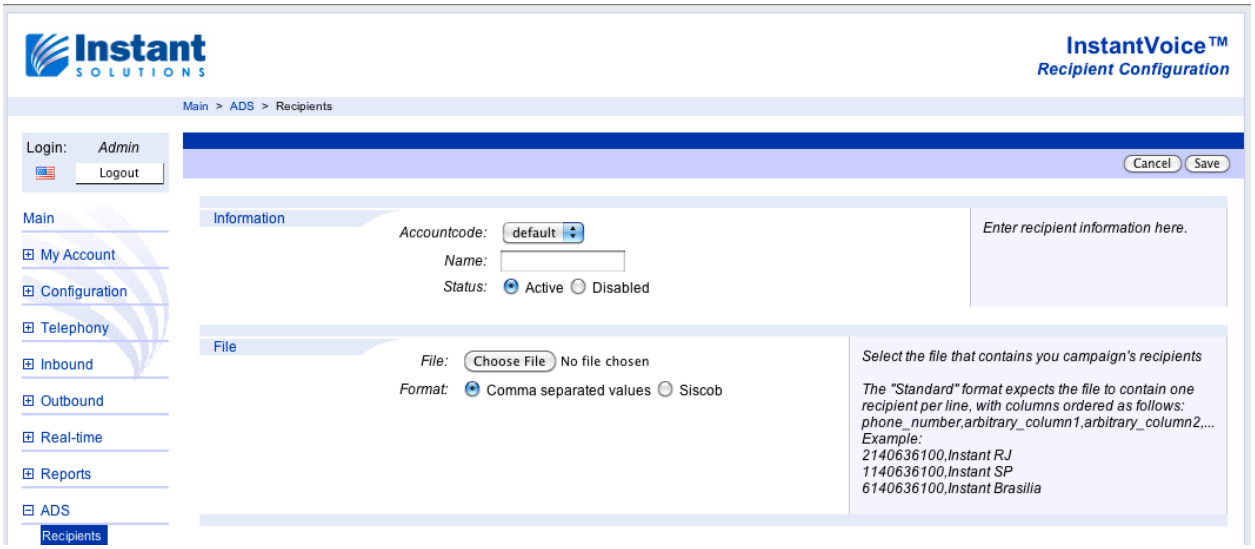

Figure 3.2: Recipients list form.

Field	Description
Accountcode	The list's accountcode.
Name	A name for the list. Just for future refference.
Status	Wheter this list is active or not. Disabled lists can't be used in campaigns.
File	The actual file with our recipients list.
Format	The format in which the list is in (covered above).

3.3 Creating a Campaign

Now in the same ADS menu, click on Campaigns.

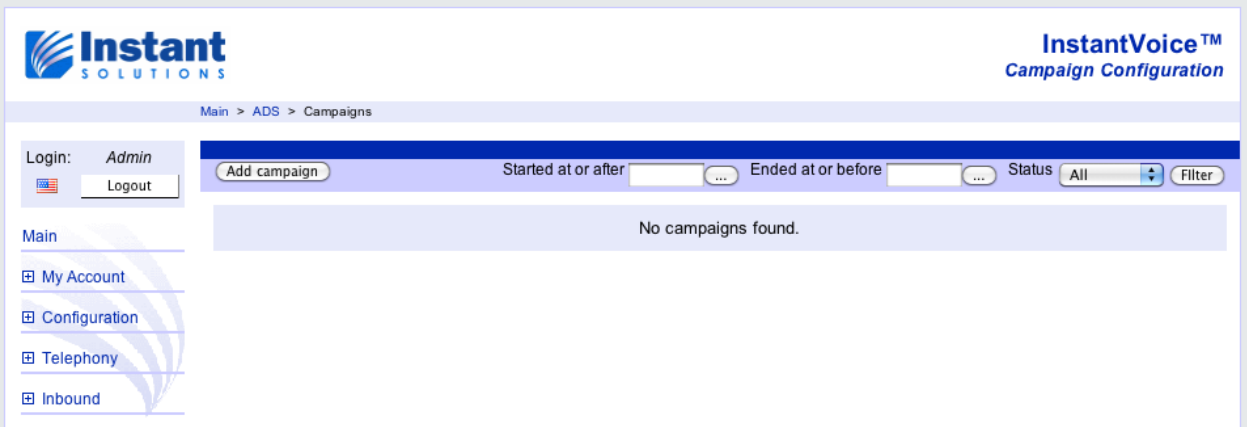

Figure 3.3: Empty campaigns list window.

Click on the "Add campaign" button and fill the following fields:

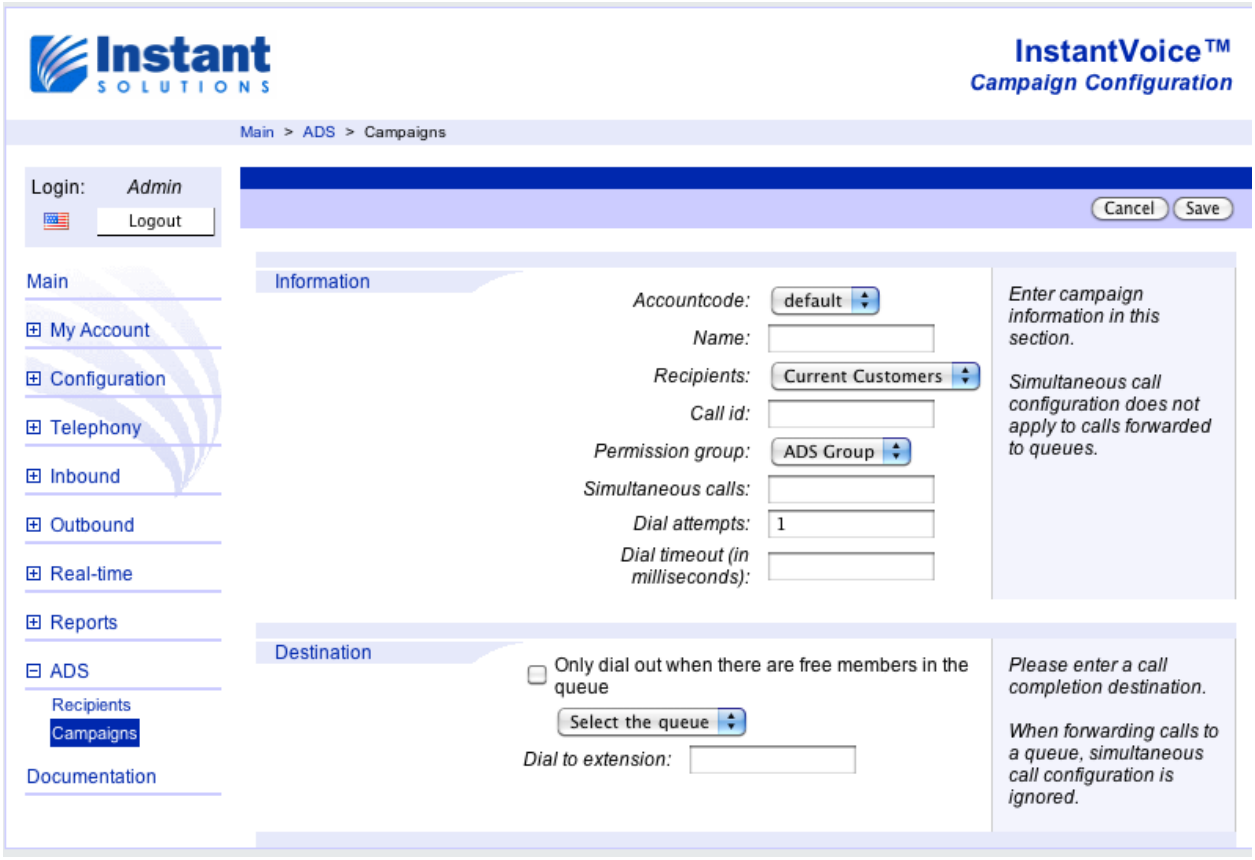

Figure 3.4: Add campaign form.

Field	Description
Accountcode	The campaign's accountcode.
Name	A name for the campaign. Just for future refference.
Recipients	A dropbox showing all your active recipients list.
Call id	The caller id you want to appear to the phones being called. This field can be left blank.
Permission group	This is the permission group which will be used to define the routing for this campaign's calls.
Simultaneous calls	This field defines the amount of simultaneous calls that can be started for this campaign. The ADS daemon will watch the number of current calls and start a new one as soon as a call is hung up, always keeping this ammount of calls running until the campaign is finished. Note that if you check the "Only dial out when there are free members in the queue" option, this will be ignored. This field can be left blank.
Dial attempts	The number of attempts to call every individual registry from your recipients list. It works this way: We will call all your registries once. If this parameter is set to, say, 3 the ads will see the calls which failed and dial again. Then for the third and last time, it will check the calls which failed again and try them.
Dial timeout	This field let's you define a timeout for this campaign's dial attempts. This value is in milliseconds and can be left blank.
Only dial out when there are free members in the queue	If you check this, the ADS will check for free agents on the informed queue and start calls until all the agents are occupied.
Dial to extension	The extension which the callee will be sent after he answers the call. This field can be left blank.

3.4 Managing your campaigns

On this same page you can manage yor campaigns.

3.4.1 Search Filters

You can use 3 filters to browse through your campaigns: start date, end date or status.

Filter	Description
Started at or after	Use this to see only the campaigns which were startded from this day on.
Ended at of before	Use this to see only the campaigns which ended before or on the defined date.
Status	Use this to filter by status.

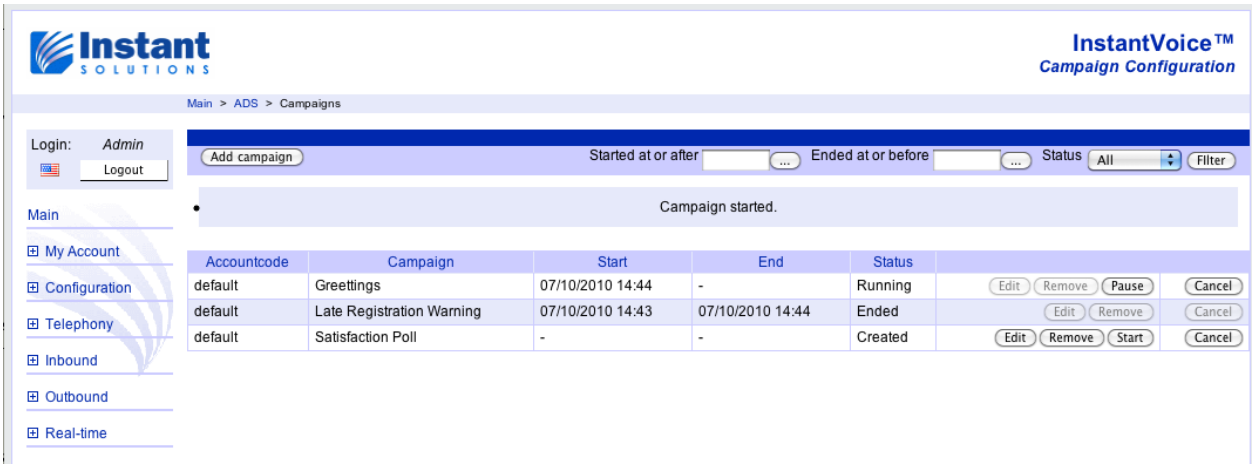

Figure 3.5: Campaigns list window.

3.4.2 Campaign Statuses

When a campaign is created it's assigned the "Created" status. From that point you can start or cancel it. A "Canceled" campaign can't be used and can't have it's state changed anymore. When you start a campaign it will be assigned "Running" status and when all calls are finished, "Ended" status. While running, a campaign can be paused and unpaused at will, it will gain "Paused" status. You can also cancel a running campaign.

3.4.3 Other Actions

While a campaign hasn't been started, you are able to edit and remove it. After the start, those are not possible anymore.

3.5 Reports

On our reports module, there are three reports related to the ADS: Overview, Statistics by Campaign and Detailed Report.

3.5.1 Overview

This report lists the campaigns and show the basic overview data.

It receives the following parameters:

Parameter	Description
Accountcode	The desired campaigns accountcode.
Start Date	The beginning of the date range of the wanted campaigns. Including this date.
End Date	The final date of the date range of the wanted campaigns. Including this date.

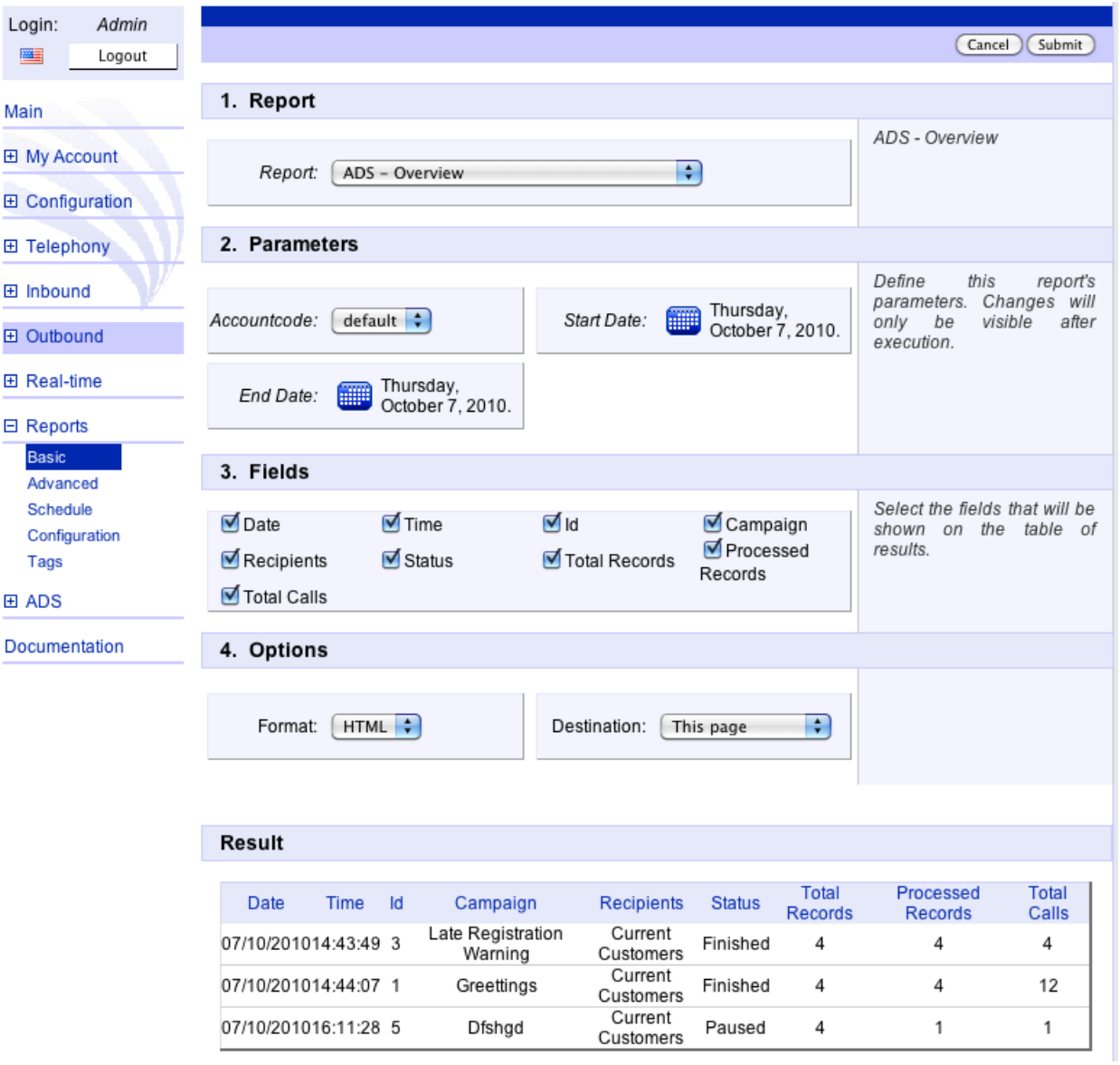

Figure 3.6: ADS - Overview.

And the returned fields are:

Field	Description
Date	The date in which the campaign was started.
Time	The time in which the campaign was started.
Id	The unique identifier of the campaign, you will need this to get the other ADS reports.
Campaign	The campaign name.
Recipients	The campaign recipients list name.
Status	The current status of the campaign.
Total Records	The total number of records in the recipients list of this campaign.
Processed Records	The number of records which were processed.
Total Calls	The total number of calls made.

3.5.2 Statistics by Campaign

This reports groups the records called by 3 metrics: final status, number of dial attempts made and tags. For each one of those 3, the report shows the described retur fields below.

It receives the following parameters:

Parameter	Description
Accountcode	The desired campaign accountcode.
Campaign id	The unique identifier of the desired campaign.

And the returned fields are:

Field	Description
Type	The group which this call falls into. For the final status it will be one of the 9 call status possibilities. For the dial attempts will be the number of dial attempts needed by that call. For the tags will be the whichever was tagged to it.
Total	The absolute total of calls from this campaign which fit this type.
Percentage	The percentage of calls from this campaign which fit this type.

3.5.3 Detailed Report

This reports lists line by line all the records from the recipients list and its current status.

It receives the following parameters:

Parameter	Description
Accountcode	The desired campaign accountcode.
Campaign id	The unique identifier of the desired campaign.

And the returned fields are:

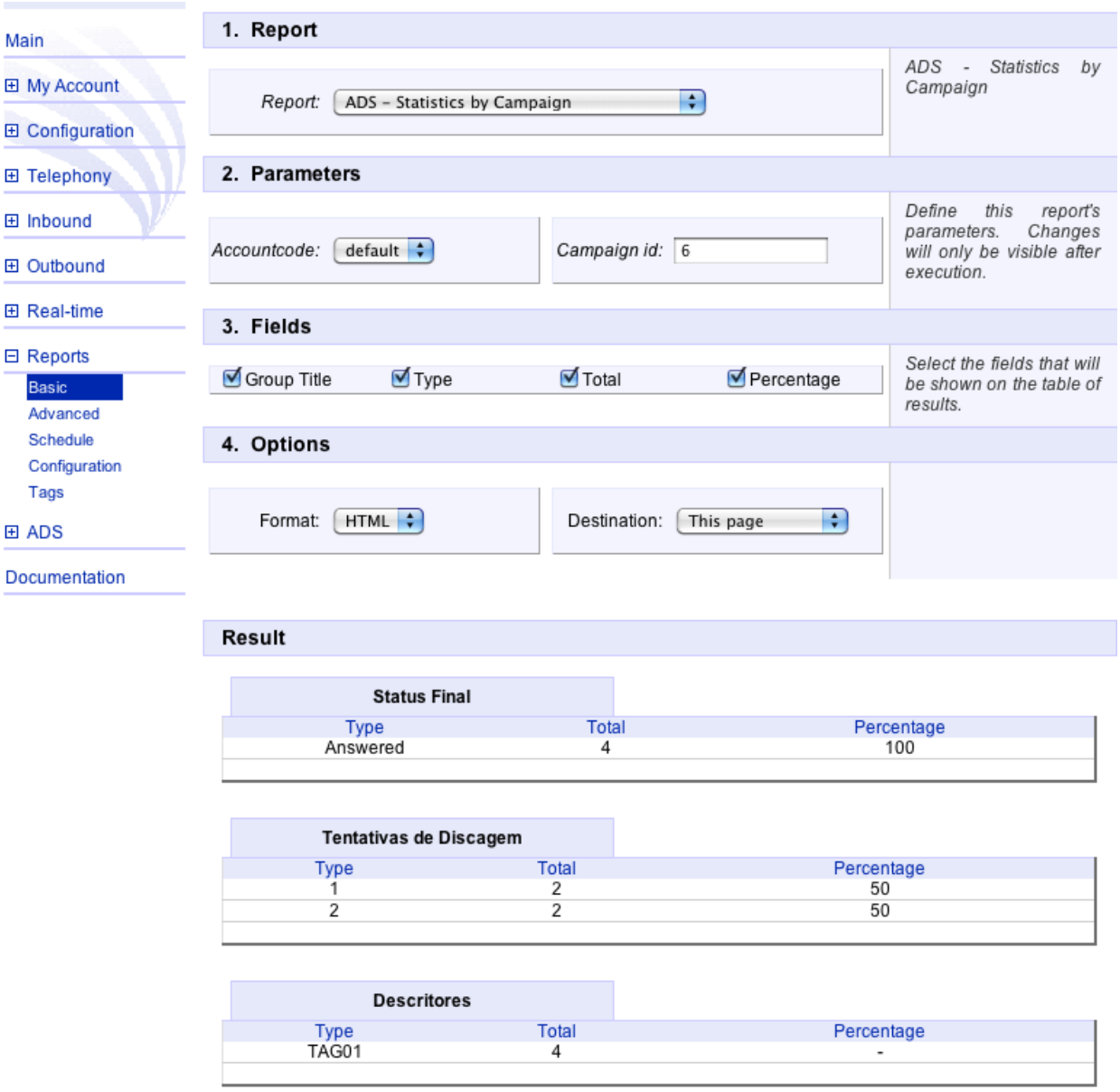

Figure 3.7: ADS - Statistics by Campaign.

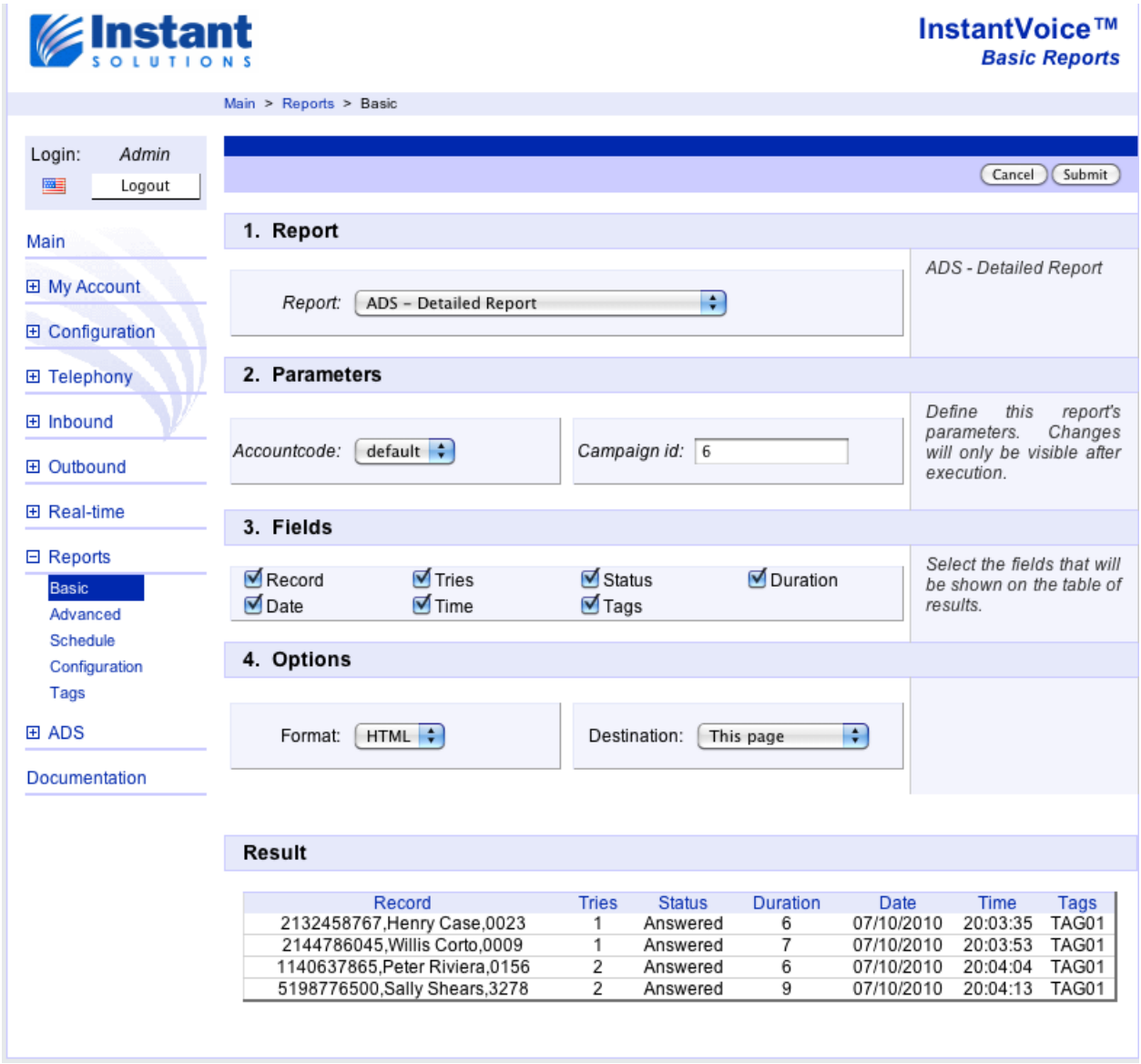

Figure 3.8: ADS - Detailed Report.

Field	Description
Record	The full recipient line from your recipients file.
Tries	The number of attempts made to call this number.
Status	The current status of the registry. The possibilities are: ready, running, talking, answered, busy, noanswer, failed, answmachine, fax.
Duration	The duration of the last call attempt.
Date	The date of the last call attempt.
Time	The time of the last call attempt.
Tags	The tags of the call.

CHAPTER

FOUR

Changelog

Here you will find a list of changes in software from one InstantVoice release to another. Changes are organized by release version and package.

4.1 Release 3.2.0

Following is a list of changes from InstantVoice release 3.1.0 to 3.2.0.

4.1.1 Asterisk

Package: instantvoice-asterisk-1.2.19_3.2.0-1

- (Bug fix) Prevents chan_local deadlock from occurring when transferring a call under heavy load.
- (Enhancement) The Hangup manager action now also takes Uniqueid as a parameter.

4.1.2 Asterisk Addons (Exclusive)

Package: instantvoice-asterisk-addons-instant-3.2.0-1

- (Enhancement) Implements web service hooks in app_ivacd.

4.1.3 Automated Dialing System

Package: instantvoice-ads-3.2.0-1

- (Enhancement) Implements dial-out pacing by linking a campaign to a queue.

4.1.4 Automatic Call Distributor

Package: instantvoice-ivacd-1.2.0-1

- (Bug fix) Fixes bug where an inbound call would be dropped by the system after having been transfered by an agent to another queue, if it was ignored by the first agent to receive it in the second queue.
- (Bug fix) Fixes a bug where if an agent makes an attended transfer using SIP signalling (such as a phone's "Transfer" button) instead of Asterisk's features, this agent's lines would be hung on an "up" state indefinitely.
- (Enhancement) Creates new JSON agent events: login, logout, pause and unpause.

4.1.5 Dialplan

Package: instantvoice-config-3.2.0-1

- (Bug fix) Fixes a bug where sound files loaded through the web interface would not be played when called through ViVRE's *waitdigits* action.
- (Enhancement) Implements web service hooks in the dialplan.

4.1.6 Licensing

Package: instantvoice-ivlicense-1.2.0-1

- (Enhancement) In order to make licenses backward- and forward-compatible for releases with the same major number, only check a given license's major version number.

4.1.7 Query Processor

Package: instantvoice-ivquery-3.2.0-1

- (Bug Fix) Fix HTML syntax error on table ouput when *sumcols* or *groupcols* were defined for a query.

4.1.8 Reports

Package: instantvoice-sql-3.2.0-1

- (Bug Fix) Fixes quality survey reports to conform to new data model.
- (Enhancement) Implements new ADS-specific reports.

4.1.9 Schemas

Package: instantvoice-schemas-3.2.0-1

- (Enhancement): Creates migration for web services data model.
- (Enhancement): Creates migration for the Automated Dialing System data model.

4.1.10 Web Interface

Package: instantvoice-interface-3.2.0-1

- (Bug fix) Fixes trunk dropdown layout in route configuration.
- (Bug fix) Fixes bug where it was not possible to save a SIP extension with a MAC address.
- (Bug fix) Fixes bug where one could not change a report's *groupcols* and *sumcols* parameters.
- (Bug fix) The accountcode can now be properly selected in follow-me configuration.
- (Bug fix) The *Previous* button now works properly on Opera when configuring agents.
- (Bug fix) Fixed a bug where if a SIP extension was removed, it would not be deleted from an associated route's permissions, causing that route to become inalterable.
- (Bug fix) Fixed bug where logoff button would not work on acdagents.
- (Enhancement) Removes dependency on soy templates.
- (Enhancement) Starts and stops the new web service daemon, ivws, according to brick type.
- (Enhancement) Implements the new ivws configuration page.
- (Enhancement) Implements the new ADS management pages.
- (Enhancement) General en_US and es_ES localization updates.
- (Enhancement) Standardization of most GUI components.
- (Enhancement) Documents Integration, the Automated Dialing System, and introduces a Changelog.

4.1.11 Web Services

Package: instantvoice-webservices-1.0.0-1

- (Enhancement) Implements the web service daemon, ivws.

INDEX