pascom
communication without borders

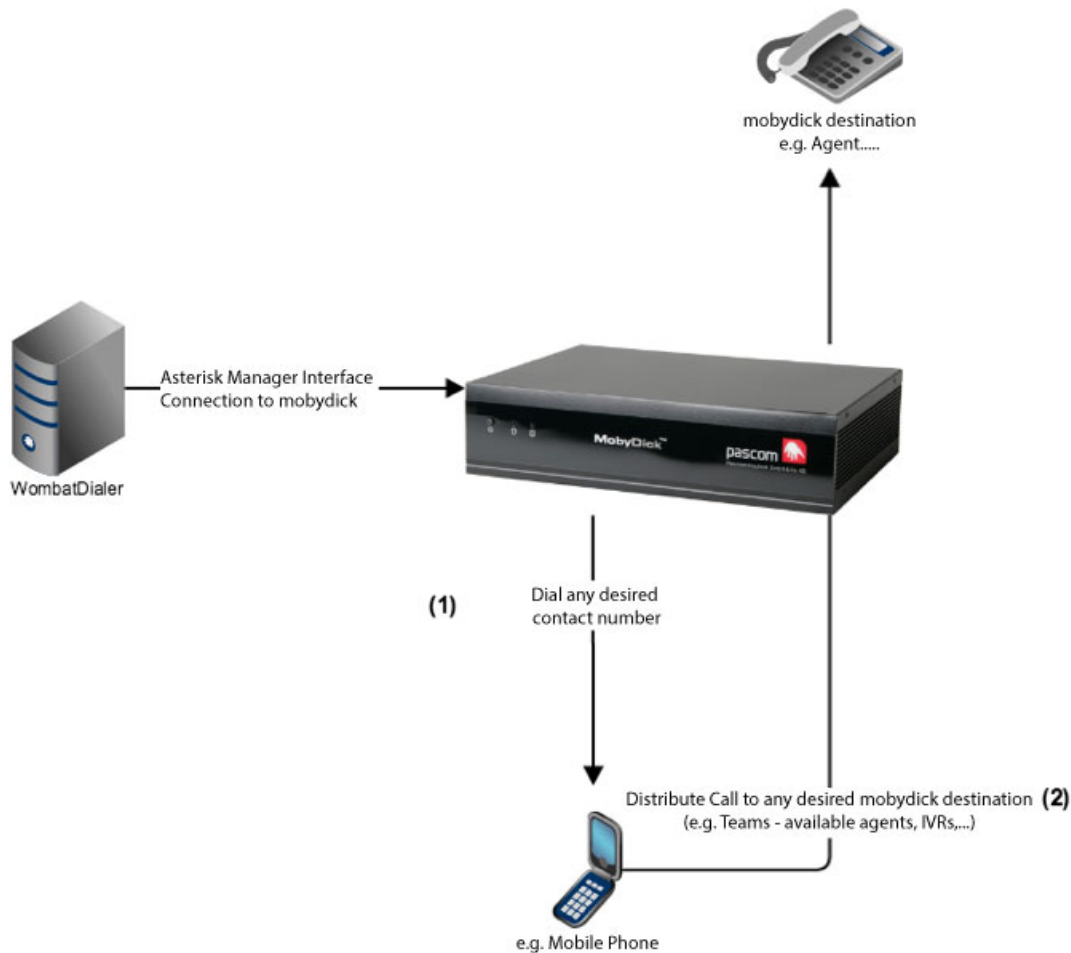# Overview

WombatDialer is a predictive dialer, which has been developed by Loway. The WombatDialer automates the telephone call origination process, thus eliminating the cumbersome, time consuming manual dialling. This enables you to achieve greater, more efficient CallCenter agent utilisation for example during Telemarketing Campaigns. Further applicable usages are outlined on the Wombat Dialer Homepage.

The WombatDialer is installed on a separate server (not directly on the mobydick server). In order to communicate with mobydick, the WombatDialer establishes a connection via the Asterisk Manager Interface. Next, a list of the desired phone numbers which are to be called must be stored within the WombatDialer which upon starting the campaign (1) will be called and as soon as the call is answered (2) be connected to any desired point of contact within mobydick, such as Teams, Users, IVR menus or scripts.

Configuration

Preparing mobydick

Asterisk Manager Interface Configuration

In order to communicate with the mobydick server, the WombatDialer requires access to the Asterisk Manager Interface (AMI).

To do this, you will need to (if you have not already done so) enable the AMI to allow public access which can be done via *Appliance > Services* and then selecting *public* under the *Allowed AMI Connections* within the **Basic Data** tab as shown below:

The next step is to create a manager account for the WombatDialer using the menu options *Appliance > Asterisk Manager accounts*:

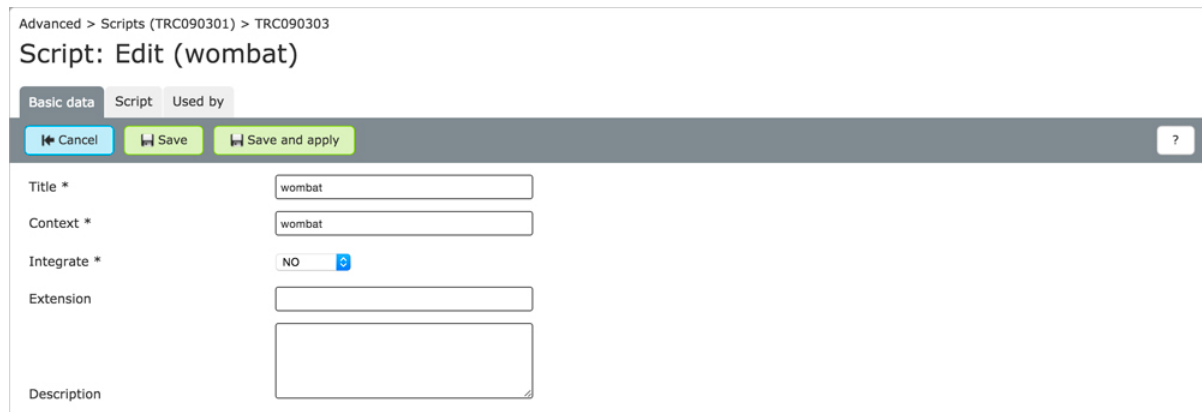Under **Enabled net** you can enter the IP address of the WombatDialer server and set the **Enabled Mask** to 255.255.255.255. This will ensure that only the WombatDialer's IP

address will be able to gain access to mobydick server, this increases security significantly and is therefore recommended.

Creating a WombatDialer Integration Script

In order to integrate the WombatDialer, a script within the mobydick is required. With future mobydick versions this script will already preinstalled and will be named **wombat.** Should the script not be preinstalled within your mobydick version, you can add it using the *Advanced > Scripts* menu:

Advanced > Scripts (TRC090301) > TRC090303

Script: Edit (wombat)

| Basic data | Script | Used by |

| ◀ Cancel | 💾 Save | 💾 Save and apply | | ? |

Title *   wombat

Context *   wombat

Integrate *   NO

Extension

Description

Using the *Script* tab, enter the following script:

```
exten => _X.,1,NoOp(WombatDialer) same => n,Wait(0.25) same =>
n,Set(MDC_CALLER_NUM_TRUNK=${CALLERID(num)}) same =>
n,Set(MDC_CALLER_NUM_INTERNAT=${CALLERID(num)}) same =>
n,Set(MDC_CALLEE_NUM_TRUNK=${EXTEN}) same =>
n,UserEvent(ResolveCallerName,Strategy: default,Outbound: 0,Channel: ${CHANNEL})
same => n,Wait(0.25) same => n,Set(MDC_NUMPREFIX_TRUNK=0) same =>
n,Set(CHANNEL(language)=de) same => n,Goto(mdc_external,${EXTEN},1)
```

Additionally, you can modify the following script variables (optional):

Variable	Meaning
CHANNEL(language)	Sets the channel language for all announcements that you may wish to use later on. Possible values can be found under the *Advanced > Prompts* menu options and then via the selection button *Action > Language: edit* . You will be able to use any of the values found within the table column **Language Abbreviations** with the CHANNEL(language) variable.
MDC_NUMPREFIX_TRUNK	Should you have configured your mobydick so that you have to dial a prefix (e.g. 0) in order to select your

	standard trunk, then enter this prefix within the MDC_NUMPREFIX_TRUNK variable.

Please don't forget that you will need to Apply all of the above modifications (jobs) in order to activate the configurations set.

Finding the Internal Trunk Name

The WombatDialer requires access to a Trunk in order to able to call external parties, which means that you will need to provider the WombatDialer with the internal mobydick name for your desired trunk. Unfortunately, it is not possible to determine the internal trunk name via the mobydick interface. In order to find out what the internal trunk name is, please login into the Asterisk CLI (see Accessing MobyDick).

Enter the command **sip show peers** which will provide you with the following output info:

```
mobydick*CLI> sip show peers Name/username        Host                  Dyn
Forcerport ACL Port    Status     Description KgZR1z3w4GBgWHa/KgZR1z3w4 172.16.214.1
D   a      64358   OK (1 ms) mdc_trunk_conf-1/dev-test 172.16.214.10              N
5060    OK (3 ms) 2 sip peers [Monitored: 2 online, 0 offline Unmonitored: 0 online, 0
offline] mobydick*CLI>
```

All internal trunk names begin with **mdc_trunk_conf-** (in our example, **mdc_trunk_conf-1**). Using the username (in this example **dev-test**) and the host address (in this scenario **172.16.214.10**) you should be able to identify which internal name belongs to which trunk, in the majority of cases.
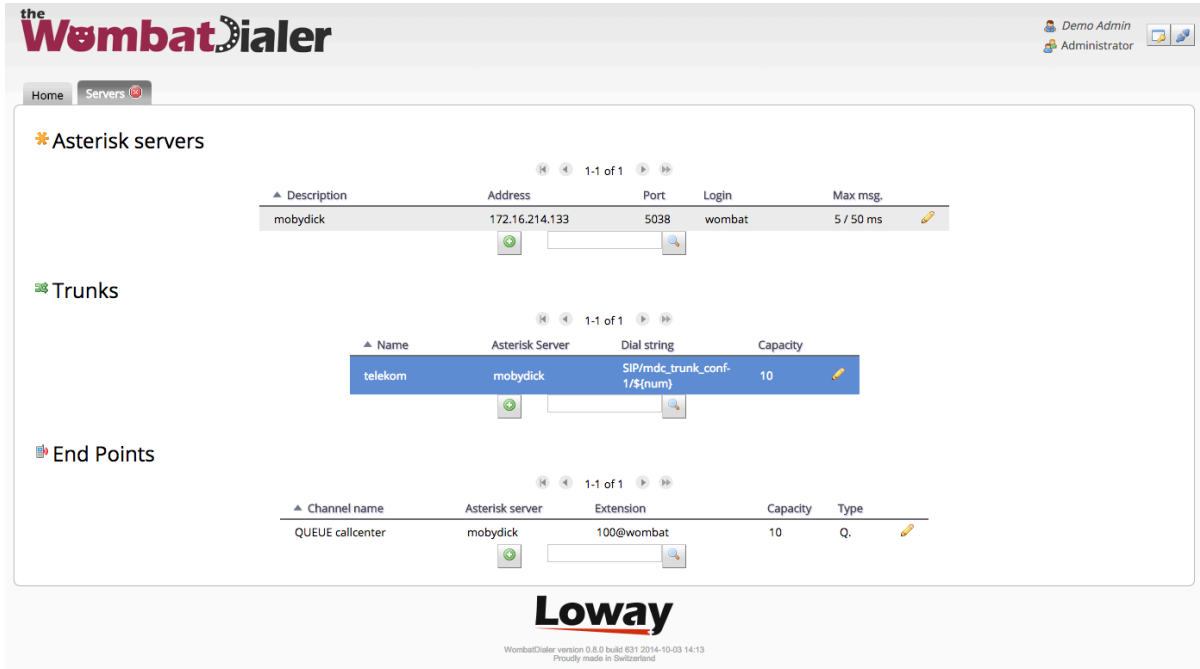
WombatDialer Installation

In accordance with Loway's Instructions you should setup the WombatDialer on a separate server.

Please Note: Under no circumstances should you install the WombatDialer directly on the mobydick server, rather you should always use another Linux Installation such as Ubuntu, Debian, CentOS, etc. on a separate server or separate virtual machine!

WombatDialer Configuration

Once the WombatDialer installation has been completed, login into the application. Under *Edit Basic Settings* you will need to configure the connection to mobydick:

Asterisk servers

Enter your mobydick IP address as the **Server address**. As **Login** and **Password** use the user data you entered when creating your Asterisk Manager Account.

Trunks

Here you will need to define which mobydick trunk the WombatDialer should use to be able to do its job. This is where you will need to enter the internal trunk name we found earlier and enter it in the *Dial String* field. Which means if your trunk is called e.g. `mdc_trunk_conf-1` the Dial String will look like:

```
SIP/mdc_trunk_conf-1/${num}
```

${num} will always be replaced by the called telephone number later on.

End Points

An End Point is the desired destination within the mobydick with those external parties who have been called over the trunk will be connected with. These end points could be for example mobydick teams, users, IVRs or scripts. There are two forms of end points available within the WombatDialer:

QUEUE	Should you like the destination phone numbers to be connected with a mobydick team, then you should use this end point form as this will mean that the WombatDialer monitors the availability of your agents etc.
PHONE	All other End Points, such as Users, IVR menus or scripts require you to use this end point type

Further parameters which you will need to setup within an End Point include:

Queue name / Phone	End point name. If you use the QUEUE end point type, then the name must match exactly the name of the that team within mobydick, Otherwise this is completely up to you
Located at: Extension	The extension number of the team, user, IVR or script etc, …
Located at: Context	Is always wombat. This ensures that the script we added above, will always be used as the initiate point into the mobydick dialplan.

Starting the Dialer

To start the dialer, navigate back to the Home Tab within the WombatDialer and start the dialer by click on the Play Button under *Dialer status*.

That's it, the integration between mobydick and the WombatDialer is done. Further details can be found either in Loway's instructions manual or in our example video documentary below.