Tango 2000 HOWTO

Table of Contents

Tango 2000 HOWTO......1
Shayne Lebrun......1
1.Introduction......1
2.The Basics......1
3.System Configuration / Requirements......1
4.Tango Installation and Configuration......1
5.TAS and Web Servers......2
6.TAS and Oracle......2
7.TAS and ODBC......2
8.Handy TAF Files......2
1.Introduction......2
1.1 Contributors and Contacts......2
1.2 Acknowledgements......3
1.3 Revision History......3
1.4 New Versions......3
1.5 Copyrights and Trademarks......3
1.6 Feedback......3
2.The Basics......4
2.1 Tango Application Sever......4
2.2 TAS, Web Server and Web Client......4
2.3 TAS Life Cycle......4
3.System Configuration / Requirements......5
3.1 Solaris......5
System Requirements......5
System Configuration......5
3.2 Linux......6
4.Tango Installation and Configuration......6
4.1 Installation – Tango 3.x......6
Solaris......6
Installing the Files......6
Setting up the Tango user account......7
Linux......7
4.2 Tango 2000......8
Solaris......8
Linux......8
4.3 Tango 2000 Service Pack 1......8
Solaris......8
Linux......9
4.4 tXserver.ini – TAS Settings......9
Cache......9
CacheIncludeFiles......9
CacheSize......9
ConfigPasswd......10
DataSourceLife......10
DebugMode......10
DSConfigFile......10
ItemBufferSize......10

Table of Contents

License.....10
ListenerPort.....11
LoggingLevel.....11
MaxActions.....11
MaxHeapSize.....11
QueryTimeOut.....11
RequestQueueLimit.....11
ThreadPoolSize.....12
ValidHosts.....12
5.TAS and Web Servers.....12
5.1 Netscape Server Configuration.....12
Tango 3.x.....12
Tango 2000.....13
Tango 2000 Service Pack 1.....13
5.2 Apache Server Configuration.....14
Tango 2000.....14
Tango 2000 Service Pack 1.....14
5.3 tXclient.ini – CGI/Plugin configuration.....14
5.4 Load Splitting/Load Balancing.....16
Tango Load Splitting.....16
Load Balancing – Hardware Load Balancers.....17
6.TAS and Oracle.....17
6.1 Oracle Support.....17
7.TAS and ODBC.....18
7.1 Solaris.....18
7.2 Linux.....18
7.3 The odbc.ini Configuration File.....18
7.4 Solaris, ODBC and Oracle.....19
8.Handy TAF Files.....20
8.1 CGI_OR_PLUGIN.taf.....20

Tango 2000 HOWTO

Shayne Lebrun

July 2000

This document describes the installation, configuration, and basic troubleshooting of Pervasive Software's Tango Application Server on Sun Solaris and various flavours of Linux.

1.Introduction

- 1.1 Contributors and Contacts
- 1.2 Acknowledgements
- 1.3 Revision History
- 1.4 New Versions
- 1.5 Copyrights and Trademarks
- 1.6 Feedback

2.The Basics

- 2.1 Tango Application Sever
- 2.2 TAS, Web Server and Web Client
- 2.3 TAS Life Cycle

3.System Configuration / Requirements

- 3.1 Solaris
- 3.2 Linux

4.Tango Installation and Configuration

- 4.1 Installation – Tango 3.x
- 4.2 Tango 2000
- 4.3 Tango 2000 Service Pack 1
- 4.4 tXserver.ini – TAS Settings

5.TAS and Web Servers

- 5.1 Netscape Server Configuration
- 5.2 Apache Server Configuration
- 5.3 tXclient.ini – CGI/Plugin configuration
- 5.4 Load Splitting/Load Balancing

6.TAS and Oracle

- 6.1 Oracle Support

7.TAS and ODBC

- 7.1 Solaris
- 7.2 Linux
- 7.3 The odbc.ini Configuration File
- 7.4 Solaris, ODBC and Oracle

8.Handy TAF Files

- 8.1 CGI_OR_PLUGIN.taf

1.Introduction

This is the Tango for Linux and Solaris HOWTO, designed to be a quick reference to the installation and configuration of various versions of Tango on supported UNIX and UNIX–like platforms. This guide is valid for Solaris 2.6 and 7 on the SPARC architecture, and for Red Hat 5.2 and 6, S.U.S.E 6.2+ and Caldera OpenLinux 2.2+, all running on Intel.

1.1 Contributors and Contacts

This HOWTO was created by Shayne Lebrun and is currently maintained by the same. Pervasive Software technical support can be reached at 1–800–287–4383 or techsupport@pervasive.com

1.2 Acknowledgements

Many thanks to the fine HOWTO HOWTO by Mark F. Komarinski. You should be able to find it at the LDP homepage at http://www.linuxdoc.org

The Tango Documentation Team makes a damn fine set of manuals; don't let your MTCV book out of sight!

1.3 Revision History

This is the first attempt at laying this all down.

1.4 New Versions

The latest version is always available at http://www.members.home.net/slebrun/howto/TangoHOWTO.html

1.5 Copyrights and Trademarks

1.6 Feedback

Suggestions, flames, corrections, requests for a dinner date, etc etc, should be sent to slebrun@home.com with a subject of 'HOWTO' and they'll be looked at. Remember, it's you who knows what needs to be in that document that currently isn't. And it won't get there if you don't tell me about it.

2.The Basics

2.1 Tango Application Sever

Tango Application Server, aka TAS, runs as a daemon process. Tango 3.x is 'tangod' and Tango 2000 is 'tango4d'. Tango 2000 also runs a seperate Server Watcher process. Tango is a 'green threaded' application; it spawns a series of threads, but manages the threads itself. This means that a single TAS will not take advantage of multiple processors. However, Tango 3.6 and Tango 2000 include load splitting capabilities which allow you to run multiple TAS daemons on one box.

2.2 TAS, Web Server and Web Client

The Tango Application Server is never contacted directly by the web browser; nor does it ever send information directly to the web browser. The browser only ever makes requests of the web server, which then forwards the request to the TAS through either a CGI or a web−server specific plugin. There are several advantages to this scheme. First, nothing on the browser side needs to be updated; no plugins, Active−X objects, client side Java, or anything are required. Second, the CGI/Plugin allows the Tango Server to be on a machine other than the web server, or indeed on several machines other than the webserver.

2.3 TAS Life Cycle

Born

Tango Server is instanced, and loads it's configuration files and what not. No user requests are accepted.

Startup

Tango Server first looks for a StartupURL. If one is specified, a request is generated. A successful return will allow Tango to move to the next stage. If the request cannot be sent, or if there is no URL specified, Tango Server moves on to the 'Running' stage.

StartupURL−Response

Tango Server stays in this stage until a response is received from the StartupURL, or until the request times out.

Running

This is the normal operating state for Tango Server; user requests are accepted, Tango Cron jobs are run, and so on.

Shutdown

When a shutdown request is received, or generated by Tango itself on a fatal exception, a ShutdownURL is looked for, and run if it exists. Otherwise, it moves on to the Waiting for Running Threads stage.

ShutdownURL−Response

Tango stays in this stage until either a response is received from the URL requested, or it times out.

Waiting

Tango Server waits for threads already running to finish, but no longer than the specified waiting period.

Dead

Tango Server kills all still running threads, cleans up, and exits.

3.System Configuration / Requirements

3.1 Solaris

System Requirements

Tango for Solaris requires Solaris 2.6 or Solaris 7, SPARC architecture, approximately 30 megabytes of disk space, and approximately 2 megabytes of memory per user, given what Pervasive considers to be 'average' memory use.

Tango 2000 requires Solaris system patches. 105591 for Solaris 2.6. 106300 and 106327 on Solaris 7. Note that 105591 revision 08 causes Tango to core dump, as does 106327 revision 07. Not all the crashes are Tango's fault. :−) Solaris patches are available at http://access1.sun.com

System Configuration

There are a series of changes which can be made to your /etc/system file to increase various resources. These changes will often increase Tango performance, as well as preventing resource related crashes. The changes will require a reboot. Be very careful when making such changes.

These are generic values which are nevertheless better than defaults.

```
*** New Settings
set shmsys:shminfo_shmmax=4294967295
set shmsys:shminfo_shmmin=1
```

```
set
 shmsys:shminfo_shmmni=100
set shmsys:shminfo_shmseg=10
set semsys:seminfo_semmni=100
set
 semsys:seminfo_semmsl=100
set semsys:seminfo_semmns=200
set semsys:seminfo_semopm=100
set
 semsys:seminfo_semvmx=32767
*** New Settings End
```

Take a look at your Solaris documentation, or http://docs.sun.com for further information.

Also, add the following line to the login script of the Tango user after installation:

```
ulimit -n 1024
```

3.2 Linux

Tango for Linux requires the appropriate flavour for the version you downloaded or purchased. Pervasive will NOT support running a version of Tango for Linux on a different Linux flavour; Tango for Red Hat Linux might not run on Caldera 2.2, let alone on Debian.

Tango for Linux currently only supports the x86 architecture.

If you're using Red Hat 5.2, you'll need to upgrade your kernel. The Tango for Linux documentation has the details.

4.Tango Installation and Configuration

4.1 Installation – Tango 3.x

Solaris

Installing the Files

The current version of Tango 3.x for Solaris is Tango 3.62. There is not expected to be any further updates to the 3.x codebase. The installation path is designed for incremental upgrades, so is a bit tricky.

First, you'll need the Tango 3.6 version. This is a compressed tarball, and will untar itself into /var/opt/EDI/.

You'll need to be superuser to install this.

```
$su -
Password: ********
# uncompress tango36sol.tar.Z
# tar -xvf tango36sol.tar
```

Next, you'll need to grab the Tango 3.62 patch file. This too is a compressed tarball. This one will uncompress to whatever directory it's in, so make a temp directory somewhere and use that.

```
$su -
Password: ********
# mkdir Tango362
# cp ./tangoSol362.tar.Z ./Tango362/
# cd Tango362
# uncompress tangoSol362.tar.Z
# tar -xvf tangoSol362.tar
```

What you'll wind up with are three binary files, a Readme.txt, and an 'odbc' directory. Move the TangoNS_ep3.so to /var/opt/EDI/lib and the t3.cgi and tangod files to /var/opt/EDI/bin. Finally, remove your /var/opt/EDI/odbc directory and replace it with this one.

```
# mv./TangoNS_ep3.so /var/opt/EDI/lib/TangoNS_ep3.so
#mv ./t3.cgi /var/opt/EDI/bin/t3.cgi
#mv ./tangod /var/opt/EDI/bin/tangod
# rm -rf /var/opt/EDI/odbc
#mv ./odbc /var/opt/EDI/
```

Setting up the Tango user account

Next, you'll need to use your favourite method to create a user account to run Tango. You can use the graphical 'admintool' program, or command line programs such as 'adduser' or 'useradd' depending on your installation.

Generally, you'll want to call the account 'tango' and also perhaps make a group called 'pvsw'. The user should be given ownership of everything in the /var/opt/EDI directory tree. If deploying in a production environment, especially on something outside of a firewall or the like, set the account to have no password, no login capabilities.

Linux

There is no version of Tango 3.x for Linux. Had you going, though.

4.2 Tango 2000

Solaris

Tango 2000 is shipped in the form of a Solaris Package Archive. You'll need to login as root, and using either the 'admintool' graphical utility, or run the pkgadd command. If you have a CD−ROM copy,mount the CD. The automounter will generally mount it in /cdrom. Copy the /cdrom/tango_as−2000/tango2000/T2000Install.tar file to a temporary directory. If you downloaded Tango 2000, it will be a compressed tar file, and you'll need to uncompress it first. If it's already uncompressed, skip the first step below.

```
# uncompress T2000Install.tar.Z
# tar −xvf T2000Install.tar
# cd T2000Install
# pkgadd −d. PVSWtango
```

and follow the prompts.

> Warning: the shipping Tango 2000 installer sometimes rejects valid CD−keys as being invalid. If this occurs, leave the CD−Key blank, and manually add your key later (see section x.x).

Linux

Tango 2000 for Linux is distributed as an RPM. On a distribution CD, you'll find the file in the tango2000 directory. Copy the appropriate file to a temp directory; Tango2000−server−4−Linux_i386.rpm if you're using Red Hat 6, Caldera 2.2 or S.U.S.E 6.2 or later. Tango2000−server−4−RedHat52_i386.rpm if you're using Red Hat 5.2. Either way, you'll need to be root.

> Note that to use Red Hat 5.2, you'll need to manually upgrade your kernel to version 2.2 or higher.

```
#rpm −Uvh Tango2000−server−4−Linux_i386.rpm
```

4.3 Tango 2000 Service Pack 1

Solaris

Tango 2000 SP1 for Solaris is a Solaris Package Archive, designed to replace/update the existing installation. It is, however, a full install, so if you don't have Tango installed already, you'll get a working install.

Otherwise, the procedure to install is the same as Tango 2000, only the archive is T2000InstallSP1.tar.Z

You'll probably want to back up your t4client.ini and t4server.ini files, as well as your web server configuration files.

Linux

Tango 2000 SP1 for Linux is an RPM designed to replace/update the existing installation. It is, however, a full install, so if you don't have Tango installed already, you'll get a working install. Otherwise, the procedure to install is the same as Tango 2000, only the RPM is Tango2000−server−4.05.i386.rpm, or Tango2000−server−4.05.RedHat52.i386.rpm if you're using Red Hat 5.2.

4.4 tXserver.ini − TAS Settings

Tango uses a file called 't3server.ini' or 't4server.ini' to control many of it's behaviors and functions. For Tango 3.x, this file is /var/opt/EDI/t3server.ini and for Tango 2000, it's $TANGO_HOME/configuration/t4server.ini. You can find a complete description of all entries in your Metatags and Configuration Variables book, but the most important/commonly used ones are listed here.

Cache

Possible entries: TRUE, FALSE

This controls weather or not Tango caches TAF files. Caching will reduce disk access, speeding up TAF execution.

CacheIncludeFiles

Possible entries: TRUE, FALSE

This controls weather or not Tango caches included files. Caching will prevent repeated disk access, speeding up TAF execution.

CacheSize

Possible entries: Numeric, measured in bytes

This measures the size of the cache for TAFs and included files. If cache grows to near this size, older documents will be flushed.

ConfigPasswd

Possible entries: text

This is the password of the config.taf online configuration application.

DataSourceLife

Possible entries: numeric, measured in minutes.

This controls how long an unused datasource connection will live. A setting of 0 will cause a datasource connection to be closed as soon as it's query is finished.

DebugMode

Possible entries: ForceOn, ForceOff, appFileSetting

This controls how the TAS handles placing debug information on the bottom of each page created; always, never, or per file settings.

DSConfigFile

Possible entries: path to file

This points to the Data Source configuration file, which gives you some finer control over how Tango uses data sources. See the Data Sources section for more details.

ItemBufferSize

Possible entires: numeric, measured in bytes

This is the maximum size of any given field that can be returned in a database action. It's main function is to prevent Tango from becoming bogged down while downloading an unusually large piece of data from a database.

License

Possible entries: alphanumeric CD−Key

This is the Tango license. The CD−Key contains the licensing information which tells Tango how to configure itself in terms of licenses and behavior.

ListenerPort

Possible entries: TCP/IP Port number

This tells Tango what port to monitor for incoming connections from the Tango CGI or Plugins. Ports cannot be shared between server software, so multiple servers running on one machine will need their own ports.

LoggingLevel

Possible entries: NoLogging, LogLevel1, LogLevel2, LogLevel3, LogLevel4

This controls how much logging Tango does. The log, "Tango.log", is written to the location specified in the LOGDIR config variable. LogLevel 3 is the best to use if you're trying to debug a Tango problem, but will slow Tango down, and will eat disk space.

MaxActions

Possible entires: numeric, 0 for no limit

This controls how many actions Tango will allow in a TAF file. This guards against things like infinite loops and overly large programs; most often used in development environments.

MaxHeapSize

Possible entries: numeric, measured in bytes

This controls how much memory the Tango Daemon process will allow itself to consume. Memory is consumed by variables, datasource connections, and cache. If Tango exceeds this number, it will shut itself down with a 'process size exceeded' message and attempt to restart itself normally. This number should be changed to provide 20% more than what you record Tango as generally using during peak use.

QueryTimeOut

Possible entries: numeric, measured in seconds

This controls how long Tango will wait for a response from a database call before timing out. Note that not all databases and drivers support this functionality.

RequestQueueLimit

Possible entries: numeric

This controls how many requests from a CGI Tango will allow to 'queue'. A very busy site can sometimes have so many CGIs stacked, waiting for Tango to service them, that some will get lost and orphaned. This helps prevent that situation.

ThreadPoolSize

Possible entries: numeric

This controls how many simultaneous threads Tango will run with. On Solaris, it is generally better to have several Tango servers running a few threads apiece than to have one Tango server running the same number of threads.

ValidHosts

Possible entries: TCP/IP addresses, colon delimited.

This is a list of what IP addresses Tango will allow incoming requests from. On a machine where the web server and Tango server are both running, this should be set to 127.0.0.1, the localhost.

5.TAS and Web Servers

The Tango Application Server runs as it's own process, where it services requests from a web server. This allows for some functionality such as load splitting and direction, as well as freedom of choice in web server and platform. The TAS need not be on the same machine as the web server, nor is it limited to being even on one machine.

The preferred way of having a web server talk to Tango is through a plugin. The plugin is written in the web server's API, or Application Programming Interface, to take advantage of specific features of that web server, as well as the advantage of having the code execute as part of the web server, instead of as a separate CGI process. On Solaris, Tango has a plugin for Netscape Server and for Apache in later versions of Tango 2000. Tango for Linux has a plugin for Apache.

5.1 Netscape Server Configuration

Tango 3.x

Locate your Netscape configuration files in /whatever/netscape/suitespot/https−MYSERVER/config generally. To the mime.types file, add this line:

```
type=magnus−internal/taf exts=taf
```

Add these two lines to the top of the obj.conf file (no line breaks; each line which starts with 'init' is a full line. Thus, there are two lines total)

```
Init fn=load-modules shlib=/var/opt/EDI/lib/TangoNS_ep3.so funcs="Tango_main,Tango_main_init"
```

```
Init fn=Tango_main_init stanza="TangoNS_ep3.so"
```

Add this line to the obj.conf file inside of the <Object name=default> area, with the other services.

```
Service fn="Tango_main" method="(GET|HEAD|POST)" type="magnus-internal/taf"
```

You'll then need to use your Netscape Administration Server to restart the server, telling it to load the configuration files when it informs you that they've been hand−edited. Then, try hitting a TAF file. If it works, great! If not, go through your configuration file again. Most of the time, a misconfiguration is the reason for it not working. One space or quotation mark out of place will destroy the entire setup.

Tango 2000

Locate your Netscape configuration files in /whatever/netscape/suitespot/https−MYSERVER/config, generally. To the mime.types file, add this line:

```
type=magnus-internal/taf exts=taf,tcf
```

Then, add these two lines to the beginning of the obj.conf file. Note that there are no line breaks in each line.

```
Init fn=load-modules shlib=/opt/PVSWtango/lib/libtango4ns.so funcs="Tango_main,Tango_main_init"
```

```
Init fn=Tango_main_init stanza="libtango4ap.so" tangoconfigpath=/opt/PVSWtango/configuration
```

Add this line to the obj.conf file inside of the <Object name=default> area, with the other services.

```
Service fn="Tango_main" method="(GET|HEAD|POST)" type="magnus-internal/taf"
```

You'll then need to use your Netscape Administration Server to restart the server, telling it to load the configuration files when it informs you that they've been hand−edited. Then, try hitting a TAF file. If it works, great! If not, go through your configuration file again. Most of the time, a misconfiguration is the reason for it not working. One space or quotation mark out of place will destroy the entire setup.

Tango 2000 Service Pack 1

Installation is the same as for Tango 2000, but with this extra line added to the services in the <Object name=default> area of the obj.conf file:

```
Service fn="Tango_main" method="(GET|HEAD|POST)" type="magnus-internal/tml"
```

Also, the mime.types file should have this line used instead of the one for Tango 2000:

```
type=magnus-internal/taf exts=taf,tcf,tml
```

5.2 Apache Server Configuration

Tango 2000 for Linux, and Tango 2000 Service Pack 1 for Solaris both support the Apache webserver through an Apache plugin. Apache will require mod_so support for all of these. The documentation states that you must custom build your Apache; this is misleading. You must build your Apache only if the pre−built version you're using doesn't have mod_so support. There are two places to check for this:

First, try running your Apache server with a '−l' argument. If mod_so is listed, you're fine.

```
# httpd -l
Compiled-in modules:
http-core.c
mod_so.c
```

Second, try looking in your httpd.conf file for an 'AddModule mod_so.o' line. If you don't have mod_so support (also known as DSO support) in some way, you'll need to compile a version that does.

Tango 2000

At the end of your httpd.conf file, add these lines:

```
LoadModule t4_module /usr/local/tango/lib/libtango4ap.so
TangoModule t4_module
 /usr/local/tango/configuration/t4client.ini
```

Modify the paths to the .so and .ini files as appropriate on your setup.

Your t4client.ini will then need a stanza for t4_module. See the t4client.ini section for examples.

Tango 2000 Service Pack 1

At the end of your httpd.conf file, add these lines:

```
LoadModule t4_apache /usr/local/tango/lib/libtango4ap.so
TangoModule t4_apache
 /usr/local/tango/configuration/t4client.ini
```

Modify the paths to the files as appropriate; Solaris defaults to /opt/PVSWtango/ for the Tango home.

Your t4client.ini will then need a stanza for t4_apache. See the t4client.ini section for examples.

5.3 tXclient.ini − CGI/Plugin configuration

Tango uses a client configuration file to tell the CGI or Plugins where to send Tango requests. The file consists of two sections; a declaration section and an information section. Here is an example for a file controlling a CGI:

```
[Tango Client Definitions]
t4.cgi=Put A Description Here

[t4.cgi]
TANGO_SERVER=127.0.0.1,18100
```

And an example of a file with both a CGI and an Apache plugin:

```
[Tango Client Definitions]
t4.cgi=My Tango CGI
t4_apache=My Apache
 Plugin

[t4.cgi]
TANGO_SERVER=127.0.0.1,18100

[t4_apache]
TANGO_SERVER=127.0.0.1,18100
```

Further entries can be made as appropriate. Use this list to find what 'keyword' to use to reference a CGI/Plugin:

CGI

Use the name of the CGI. For example, you could have two CGIs, t4.cgi and t4private.cgi, each pointing to a different Tango server.

Netscape Plugin (Tango 2000)

Use 'libtango4ns.so' as the name. You shouldn't try to load multiple plugins.

Netscape Plugin (Tango 3.x)

Use 'TangoNS_ep3.so'

Apache Plugin

Use 't4_module' for Tango 2000 and 't4_apache' for Tango 2000 Service Pack 1. You shouldn't try to load multiple modules.

Tango 3 versions earlier than 3.6 use a slightly different format. TANGO_SERVER is broken into two lines; TANGO_HOST and TANGO_PORT. Here's an example:

```
[Tango Client Definitions]
t3.cgi

[t3.cgi]
TANGO_HOST=127.0.0.1
TANGO_PORT=18000
```

5.4 Load Splitting/Load Balancing

All versions of Tango starting with 3.6 have allowed Load Splitting. You can define a series of Tango servers that a plugin can use, and it will distribute new requests between the servers. Users who make subsequent requests are directed back to the server they were at previously via the UserReference search argument or cookie. Users cannot move between servers without losing their variables and what not. Also, what Tango currently does is not load balancing; that is expected for a future release of Tango.

Tango does not need to be running on the same machine as the web server in any event. You need only run configure Tango's VALIDHOST configuration variable to include the IP address of the web server machine. Then, in the web server machine's t4client.ini file, put in the IP address and port number as usual.

Tango Load Splitting

To add multiple Tango servers, add an entry to the t4server.ini file with the new name of the server, and add a new stanza for it, generally by copying an existing stanza. Then, change all directory names to be unique, such as LOGDIR. Make sure that it's running on a unique port. Tango defaults to port 18100. Here is an abbreviated example of a t4server.ini file with two Tango Application Server instances:

```
[Tango Definitions]
TAS_1=MyFirstServer
TAS_2=MySecondServer

[TAS_1]
...
LISTENERPORT=18100
...
LOGDIR=/usr/local/tango/log.TAS_1

...

[TAS_2]
...
LISTENERPORT=18101
...
LOGDIR=/usr/local/tango/log.TAS_2
...
```

All other configuration variables would be filled in as appropriate.

To run Tango with a specific configuration definition, use the −c switch.

```
$ ./tango4d −c TAS_1
$ ./tango4d −c TAS_2
```

The −c switch can be used in conjunction with the −k switch to kill servers as well.

Note that you'll need the appropriate licenses, either one Corporate or Professional license, or as many Standard licenses as you want Tango servers.

In the t4client.ini file, simply add the IP address and port of each server to the TANGO_SERVER line of the appropriate CGI or Plugin, colon delimited. Here is an example t4client.ini file, using the CGI, pointing to three separate Tango servers running on the same machine, on ports 18100, 18101 and 18102.

```
[Tango Client Definitions]
t4.cgi=My CGI

[t4.cgi]
TANGO_SERVER=127.0.0.1,18100:127.0.0.1,18101:127.0.0,1,18102
```

Load Balancing – Hardware Load Balancers

You can use a hardware load balancing device with Tango so long as your webserver farm is behind the load splitting device, and each webserver is using an exact copy of the same t4client.ini file. The t4client.ini file should be configured to use every Tango machine you want available. The UserReference based Tango redirection should function normally.

6.TAS and Oracle

6.1 Oracle Support

Tango 3.x for Solaris does not have native Oracle support. Tango 2000 for Solaris supports Oracle 8 and 8i, and generally works with Oracle 7.3.4 and above. Tango 2000 Service Pack 1 was not tested with Oracle 7.x however, and there have been reports of problems. The Oracle 8 client can connect quite happily to the Oracle 7 server, however, so that is an option.

To use Oracle with Tango, first you require the Oracle Client to be correctly installed on the machine. Generally, a good way to check for this is to:

- check to see if there is an 'oracle' account on the machine
- try to run the Oracle program 'sqlplus'

If you can successfully run sqlplus, you should be fine. If not, well, go install Oracle.

Next, you need to make sure that the client has the appropriate entries for the datasource in question to your TNS service; generally a tnsnames.ora file. You accomplish this with your Oracle software; there are programs such as net8config and what not that do it for you. The thing to remember here is that the name of the datasource needs to be the same as the name of the datasource you're using in Tango.

Finally, to allow Tango to use the Oracle software, you need to add this path to the LD_LIBRARY_PATH environment variable of the tango user account.

```
$ORACLE_HOME/lib
```

You'll also need to define what $ORACLE_HOME is; this, of course, is the root directory of Oracle, and will be some forboding path such as, for example, /u01/software/products/8.1.5/oracle or something similar.

If you're using Oracle 7.x or 8.0.x you'll probably need to create a new client library. Instructions for this are in $TANGO_HOME/odbc/src/oracle in the readme.ora file. You can check to see if you require this patch by attempting to load the Oracle ODBC driver (see section 6). If it fails with an 'unresolved symbol' error, then you need the patch.

As the readme file says, this newly created library must appear in your LD_LIBRARY_PATH before the reference to $ORACLE_HOME/lib does. If there are no other programs running, it's generally easiest to replace the old library with the new.

Some shells seem to have trouble with the LD_LIBRARY_PATH ordering; I've seen ksh fail to correctly load the new library, and switching to sh or bash solve it.

7.TAS and ODBC

7.1 Solaris

Tango for Solaris includes the Merant ODBC driver pack, which Tango can use to connect to a wide array of databases, such as Informix, Sybase, Oracle, and now Microsoft SQL Server.

The .odbc.ini file (see below) should be located in the home directory of the Tango user account. It can be stored elsewhere, for centralization or what not, and located using an environment variable called ODBCINI.

7.2 Linux

Tango 2000 for Linux includes the iODBC driver manager, and a Pervasive.SQL driver is included with Pervasive.SQL for Linux. It will also work with Postgres SQL and MySQL drivers. Other drivers are available at http://www.unixodbc.com and http://www.iodbc.org

The .odbc.ini file (see below) should be located in /usr/local/tango/etc/odbc.ini which can be symlinked to /usr/local/psql/etc/odbc.ini if both are installed.

7.3 The odbc.ini Configuration File

The drivers are controlled by an ini file called .odbc.ini (note the leading period; it's a hidden file on Solaris.)

The file contains a list of Data Source Names followed by a definition section. It is a standard INI file format. So long as the DSN name used by a TAF is listed in this file, and the database schema is the same as the one on the development machine, the database access will work transparently.

Here is an example of a .odbc.ini file containing a single DSN called MyOracleDatabase, which is pointing to an Oracle database with a TNS name of 'LocalOracleDatabase'.

```
[ODBC Data Sources]
MyOracleDatabase=

[MyOracleDatabase]
Driver=/opt/PVSWtango/odbc/lib/VQor815.so
ServerName=LocalOracleDatabase
```

All entries will have, at minimum, a Driver line; this points to a .so shared object which is the actual ODBC driver. There will be other lines dependant on the driver itself; server configuration, special flags, that sort of thing. Refer to the readmes and to the odbchelp.pdf file (on Solaris) or to specific driver documentation for those settings, as well as any environment settings required.

Note that ODBC drivers still often require database specific Client software; for example, the ODBC Oracle drivers require the same Oracle OCI client software that Tango requires for a native Oracle connection. Sybase client software must be installed for the Sybase ODBC driver to function, and so on. Again, see the odbchelp.pdf file, or driver documentation for specifics.

Anybody who would like to write small crib notes on configuring specific databases to work with Tango for Solaris or Linux through ODBC, feel free and send them to me and they'll be included in this HOWTO, with full credit of course.

7.4 Solaris, ODBC and Oracle

Certain versions of the Oracle client are missing key functionality that the Oracle ODBC drivers require, as well as the Tango native Oracle drivers.

The easiest way to check for this condition is to use the 'ivtestlib' program to attempt to load the Oracle ODBC driver. Become the Tango user, and move to $TANGO_HOME/lib. Look for a driver with 'or7' for Oracle 7, or 'or8' for Oracle 8. Depending on your version of Tango, the full name might be 'IVor714.so' or 'VQor714.so'. The last number before the .so is the ODBC version.

Once you have this file name, move one directory up, and then go into the 'bin' directory. There, execute the 'ivtestlib' program with the file name as an argument.

```
$ cd $TANGO_HOME/odbc/lib
$ ls *or8*
VQor814.so
$ cd ../bin
$ ./ivtestlib ../lib/VQor814.so
Load of ../lib/VQor814.so successful, qehandle is 0xFF24120
```

If the Oracle drivers need to be patched, you'll see a long message ending in the fact that there is an 'undefined symbol slpmprodstab'. This will prevent Tango Oracle Native connections from working as well as ODBC. In the $TANGO_HOME/odbc/src/oracle directory is a readme.ora file and two scripts, one for Oracle 7 and the other for Oracle 8. The readme file details the creation of a new 'libclntsh.so' file, which is the main Oracle client library. In a nutshell, make sure you have ORACLE_HOME defined, and run the appropriate script for your version of Oracle. This will create the .so file and place it, by default, into $TANGO_HOME/odbc/lib. Then, make sure that that path appears in your LD_LIBRARY_PATH before any references to $ORACLE_HOME/lib does. You can also move the libclntsh.so file to your $ORACLE_HOME/lib directory, or create a link between $ORACLE_HOME/lib/libclntsh.so and the one you just created; the preexisting libclntsh.so is a link to another library anyway.

8.Handy TAF Files

8.1 CGI_OR_PLUGIN.taf