



Server Installation Guide

Witango 5.5 Server for OS X

August 2004

Witango Technologies Pty Ltd
Level 1,
44 Miller Street,
North Sydney, NSW, 2060
Australia

Telephone: +612 9460 0500
Fax: +612 9460 0502

Email: info@witango.com
Web: www.witango.com

Table of Contents

I	Table of Contents	i
2	Introduction	1
	About Witango	2
	Understanding Witango Installation	2
	Getting Ready to Install Witango	3
	Checking Your Witango Package	3
	Viewing the Installation Notes	3
	Java Runtime Environment (JRE)	3
	Viewing the Release Notes	3
	Database Connectivity	3
3	Witango 5.5 Application Server for OS X 10.3	5
	Minimum System Specifications	6
	Operating System	6
	Hardware	6
	Web Server	6
	Other Requirements	7
	Extracting Witango Application Server 5.5 Installer Software	8
	Create a Witango user	10
	Cancelling Witango Application Server 5.5 Installer Processes	12
	Installing the Witango Application Server 5.5 Software	13
	Starting the Witango Installer	13
	The Welcome screen	14
	The Important Information screen	14
	The License Agreement screen	15
	The Choose Destination Location screen	16
	The Ready To Install screen	17
	The Software Install Process	17
	The Install Completion screen	19
	Witango Server Startup process	20
	The startup script	20
	For more information	22
	Startup sequence control	22
	Configuring Witango Server to Connect to FileMaker Client	24
	Remove the startup script	24
	Create a startup command script	24

Make the startup command run automatically on login	26
Configuring Witango Server with Java Integration	29
Configuring the Java environment for Witango Server	29
JAVA VIRTUAL MACHINE parameter in witango.ini	30
Configuring Witango Server for JDBC	30
OS X Java Documentation	30
Setting up a JDBC driver	31
Configure the Classpath	31
The jdbc.ini file	32
Configuring Witango Server for ODBC	34
iODBC	34
Further information on iODBC	34
Configuring Witango Server for Oracle Call Interface (OCI)	35
Install the OCI driver	35
Set up the Server's OCI environment	36
Development Studio Setup for the OCI driver	36
Tightening the permissions on the OCI Driver	37
Sample ~/.MacOS/environment.plist	37
Sample tnsnames.ora file	38
Manually Configuring Apache Web Server	39
Configuring Apache 1.3	39
Configuring Apache 2.0.x	39
Troubleshooting	41
The Unlicensed Server Message	41
The Web Server plug-in does not connect to the Server	42
Unable to connect to the specified data source	45
The server is starting up message	46
The XML of the taf file is returned	46
The server is not started	47
The Unix Command Line	49
Changing unix file permissions	49
Witango command line options	49
Global User environment variables	49
Useful terminal commands and what they do	49
Adding commands shortcuts to your terminal session	50

Introduction

Basic know-how for the Installation of Witango 5.5 Server



About Witango

Witango consists of a number of components, which are designed to allow you to easily develop, deploy, and maintain your dynamic Web site.

The two main components of Witango are:

- **Witango Development Studio** is used to create Web applications. It is available for Windows and Macintosh platforms.
- **Witango Application Server** is used in conjunction with a web server to execute the web applications developed with Witango Studio. It is available for Windows, Macintosh, Solaris, and Linux platforms. A development version of Witango Server is available to customers who purchase the Witango Studio.

A typical installation of Witango consists of a Witango Development Studio for each developer and one or more Witango Application Servers for deployment. The number of Witango Servers you need depends on the traffic volume on your Web site and the complexity of your applications.

Understanding Witango Installation

A typical Witango installation would include the following tasks:

- extract the executables and configuration
- enter the License Key
- creation of directories on your hard disk
- copying of Witango files to directory
- installation of startup and shutdown scripts in system directories
- configuration of your Web server to work with Witango
- installation of ODBC / JDBC drivers
- creation of ODBC data sources, if required
- installation of the Java2 Runtime Environment 1.4.2 or higher
- configuration of environment variables
- access to the superuser account.

Getting Ready to Install Witango

Checking Your Witango Package

You should ensure that you have downloaded the latest archive which contains all the files necessary to run Witango Server. This information will be available at <http://www.witango.com/downloads>

Viewing the Installation Notes

Before installing Witango, make sure you view the Installation Notes (`Install.txt`). The Installation Notes contain information concerning Witango installation.

The `Install.txt` file is located in the `Install` directory of the archive file.

Java Runtime Environment (JRE)

You must have Java2 Runtime Environment (J2RE) 1.4.2 or greater installed on your machine if you want the Witango Server to execute JavaBeans or initialise the JDBC interface. You can get the latest release of J2RE from the following Web sites:

<http://www.apple.com/java>

Viewing the Release Notes

After installing Witango, make sure you view the Release Notes (`Readme`). The Release Notes contain additional information concerning Witango. Also visit the Witango Web site (www.witango.com) to get any late-breaking news on Witango.

The `Readme` file is installed in the Witango installation directory.

Database Connectivity

Witango Server can access the following databases via database driver managers:

Connection	ODBC driver manager	Download
Oracle version 8*	OCI 8.1.7.6 or higher	otn.oracle.com
ODBC 3.0 or higher	MDAC 2.8 or above UNIXodbc iodbc	www.microsoft.com/mdac www.unixodbc.org www.iodbc.org
JDBC	JVM 1.4.2 or higher	java.sun.com www.apple.com/java
FileMaker	via Apple Events or JDBC	www.filemaker.com

- * If you use Oracle, the required libraries are included with Oracle. You need to add the `$ORACLE_HOME/lib` to the `LD_LIBRARY_PATH` environment variable for the Witango account so that Witango can use them.



Note The machine that runs Witango Studio also requires its own DBMS client software and database driver.

Witango 5.5 Application Server for OS X 10.3

2

Installation instructions for Witango 5.5 Server for OS X

Minimum System Specifications

Witango Technologies strongly recommend the following as minimum setup standards for Witango Application Server 5.5:

Operating System Mac OS X 10.3.4 or above
 Mac OS X Server 10.3.3 or above

Hardware PowerMac G4 400 MHz
 256 Mb RAM
 30 Mb hard disk space

Web Server Any cgi compatible Web Server. One of the following is recommended however:

- WebStar 5.3.3 or above
- Apache – version 1.3.27 or above
- Apache2– version 2.0.44 or above

Any frames-capable Web Browser.



Note Witango Application Server 5 may install and run on systems of lesser specifications than set out above however support from Witango Technologies staff will **not** be available for these systems.

Other Requirements

The Witango software is supplied to you via download from Witango's web site found at <http://www.witango.com>. The software installer is in a compressed Stuffit archive format (.sit) and will appear in the location that you chose during the download process.



Note The number which appears in the archive file name will vary in line with the current software version being supplied.

Version 8 or above of the Stuffit Expander utility must be installed on your machine in order to be able to extract and use the Witango installer. A copy of the Stuffit Expander utility can be downloaded free of charge from their website at <http://www.aladdinsys.com>.

It is strongly recommended that you close all programs or applications on the machine on which you are installing Witango prior to starting the Witango install application. This reduces the risk of software conflicts and frees system memory for the installation process.

You must have access to the administrative account on your system that has Administrator level privileges to successfully install Witango software.

Extracting Witango Application Server 5.5 Installer Software

The Witango Application Server software installer is supplied to you in Stuffit archive format. In order to use the installer it must first be decompressed from the Stuffit archive.

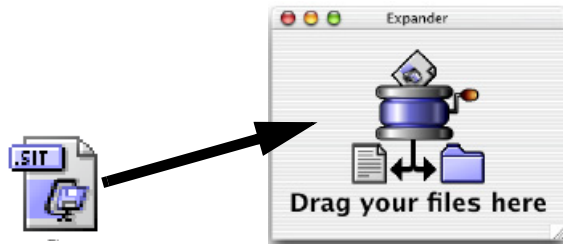
Your web browser may have automatically decompressed the Witango installer into the same directory as the archive. If this is the case you may skip to the next section of the install process, Create a Witango user on page 10.

Drag the archive that you downloaded from the Witango web site onto the Stuffit Expander icon. This will automatically start Stuffit Expander and extract the Witango installer package into the same directory where you saved the Stuffit archive.

- I Open the Stuffit Expander utility by double clicking on its icon. The following window will appear:

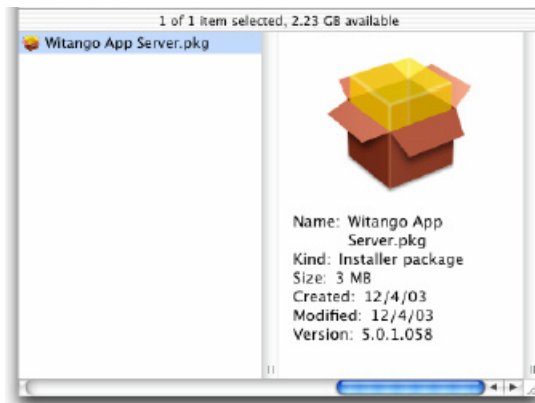


- 2 Drag the archive file you downloaded from the web site onto the window.



Witango_App_Server_for_OSX
_5.x.xxx.pkg.sit

- 3 The extraction process will begin automatically and the Witango App Server package will be saved to the same directory as the archive file.



- 4 Exit the Stuffit Expander utility on completion of the extraction process

Create a Witango user

Before starting the Witango Server installation make sure that a witango user has been created. This user will be the owner of the Witango process and will have all the access rights and privileges that the groups that it belongs to have. The user can be given any password but it is recommended for the above reason that this user has a highly secure password.

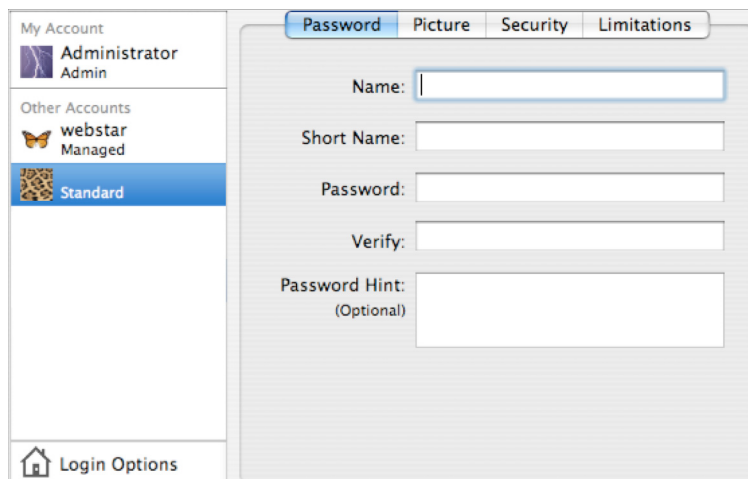
- 1 In the System Preferences choose the Accounts pane.



- 2 Click on the + button



- 3 A new blank user will appear in the list.

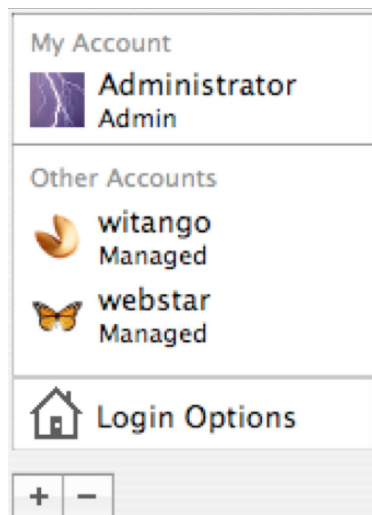


- 4 The following sheet will be displayed. Rename the user to witango and add give it a password. Close the window. A new witango user will be added to the user list.

The image shows a user creation dialog box with four tabs: Password, Picture, Security, and Limitations. The 'Password' tab is selected. The fields are as follows:

Field	Value
Name:	witango
Short Name:	witango
Password:	
Verify:	
Password Hint: (Optional)	

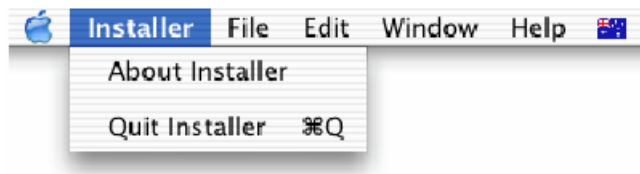
- 5 Once the witango user has been created, close the Accounts pane and close the System Preferences.



Cancelling Witango Application Server 5.5 Installer Processes



It is possible to cancel any process that the installer is engaged in and escape the Installer application at any time by selecting Quit Installer from the Install menu.



The Witango Application Server installer application will close.

Installing the Witango Application Server 5.5 Software

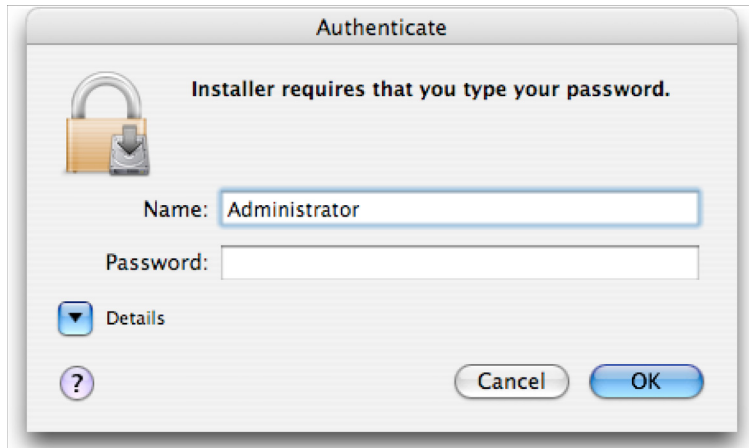
Installation of the Witango Application Server software is an automated process. However it requires you to make several decisions regarding the actions the installer should take prior to the software install beginning. This section will guide you through this installation setup process.



Caution Your web server should be shut down while the software is installed and configured on your system. You will need to restart the web server after the install process has finished. The install process only takes a few minutes and does not require you to reboot your machine.

Starting the Witango Installer

- 1 Double click the Witango App Server.pkg file. The Witango Server 5.5 For OS X installer will start extracting the files required for Witango Application Server installation. A message indicating that you need to enter an administrative username and password for the Installer to use during the installation is displayed.



- 2 Enter an administrator user name and password and click OK.

The Welcome screen



Click the Continue button to proceed to the next screen.

The Important Information screen

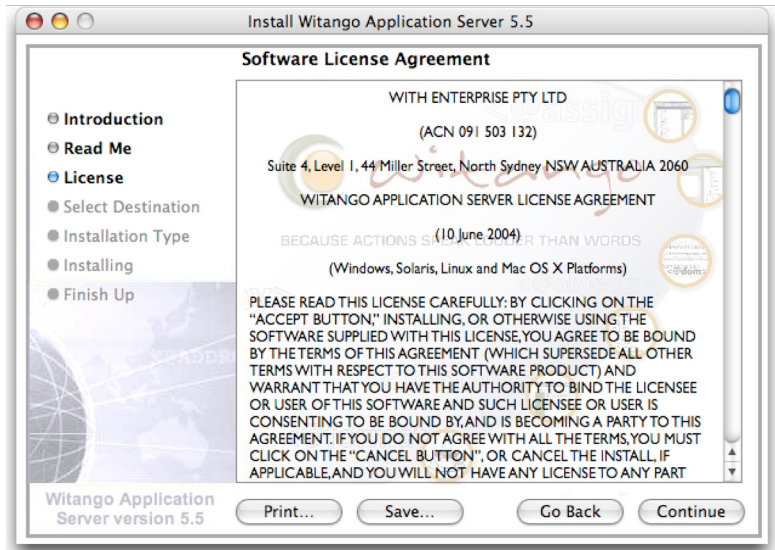
Read the Witango Application Server **Important Information** carefully. This screen contains last minute information regarding the build of the server you are installing.



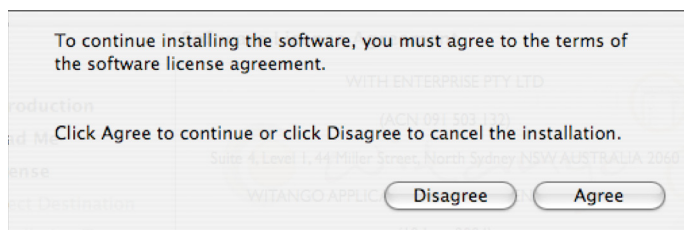
Once you have read the Important Information, click the Continue button to proceed to the next screen.

The License Agreement screen

Read the Witango Application Server licensing information carefully - in case it states that you are signing over your house to With Enterprise. ;-)



Once you have read the license agreement, click the Continue button to proceed to the next screen. After you have clicked Continue a sheet will be displayed asking you to Agree or Disagree with the terms of the license.



Now either:

Click **Agree** to accept the terms of the license agreement and continue with the installation.

Click **Disagree** to reject the terms of the license agreement and cancel installation.

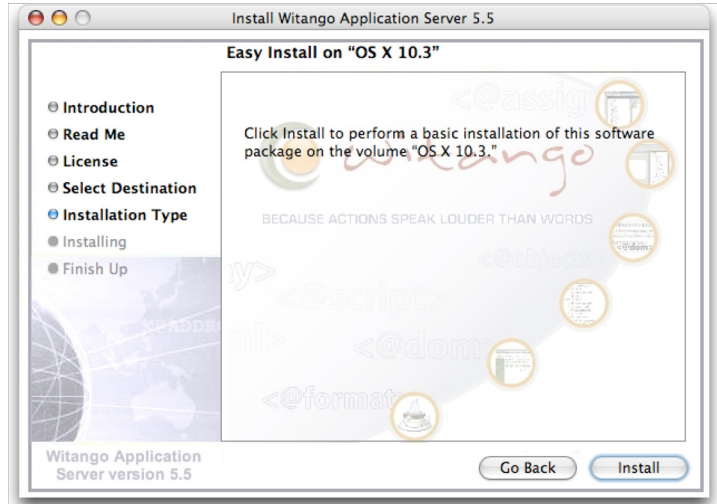
The Choose Destination Location screen

The Witango server will be installed in the /Applications/Witango directory of the root volume of your system. This cannot be changed with the installer.



The Ready To Install screen

- I Click Install to start the installation of Witango Application Server.



Note If you are upgrading a previous version of Witango 5.5 the **Install** button will read **Upgrade**.



Note During the upgrade process the installer will attempt to backup the witango.ini files before it begins the upgrade and configure the Witango Application Server with the same configuration. This does not occur if Witango Application Server has not been previously installed in a directory other than /Applications/Witango.

The Software Install Process

During the software installation process the installer performs a number of tasks and several messages informing you of the installer's current action will appear as these tasks take place. The installer performs the following tasks:

Stopping existing Witango Server (Upgrade Only)

The Witango installer must stop any existing Witango Servers before it can begin to install and configure the new Witango Application Server. If the installer is not able to stop all the existing Witango Server processes the new server executable and libraries may not be installed. If this occurs manually stop all server processes and run the installer again.

Checking for the Presence of Prior Installations

The Witango Installer checks for prior installations of Witango Application Server. Should it find one it will attempt to backup the witango.ini file before it begins to install and configure the Witango Application Server software again.

File Transfer

The witango installation files are copied to the hard disk.

Configuring the Witango Web Server Plug-Ins

On completion of the transfer of software to your hard disk, the Witango installer will configure the Witango web server plug-ins for Apache server and WebSTAR if it is present in the /Applications/4DWebSTAR directory.

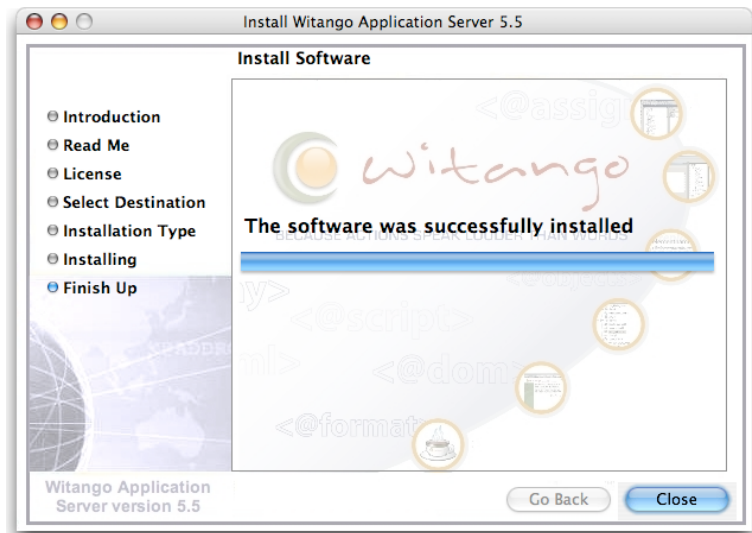
Restore the Server configuration (Upgrade Only)

The Witango Installer will restore the original configuration file.

Restarting the Witango Application Server

The Witango Installer will then start the newly installed Witango Application Server.

The Install Completion screen



Click Close to exit the installer. The installer will clear any temporary files from your hard disk that it has created and then close down.

At this point the Witango Application Server has been successfully installed onto your system.

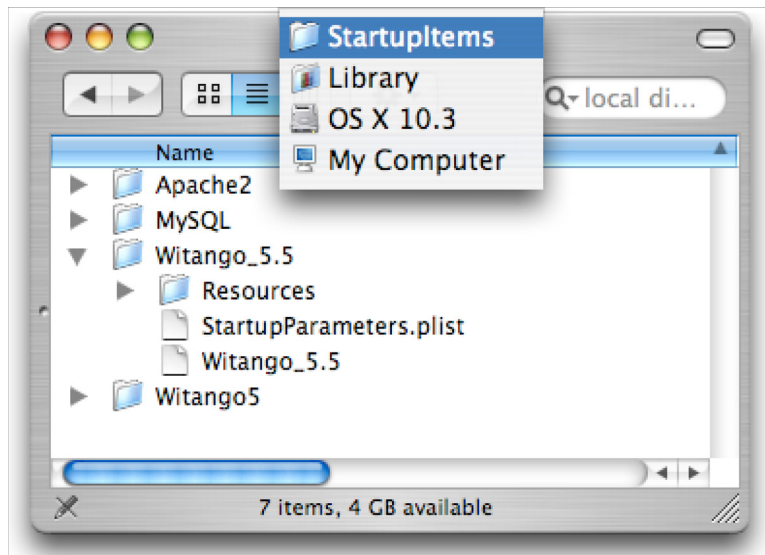
The Witango server will automatically start up as part of the start up sequence of OS X.

Witango Server Startup process

The startup script The Witango process can be started manually, by a login item, or by a startup script depending on the configuration you require.

If FileMaker datasources are to be used the witango server process must be started in the same user space as the FileMaker process. As the FileMaker client is opened via a double click, th process is owned by the currently logged in user. To change the Witango server to start up via a login item, the startup script should be removed and a startup command created and setup through the “Login Items” preference pane (see the section *Configuring Witango Server to Connect to FileMaker Client* on page 24).

The Witango installer will configure a default start up script to start a single server proces as a daemon (background process) that will start up in the user space for the **witango** user. The startup script is located in the `/Library/StartupItems/Witango_5.5` directory and is called `Witango5`.



The following example Witango5.5 startup script would initialise the environment variables necessary to use the ODBC, OCI, JDBC and JavaBean handler with the server (Professional Edition). The script would also start and stop 4 different witango server processes each from a different configuration stanza in the `witango.ini` file.


```

#!/bin/sh
#
# Witango Application Server Startup Script
#
# Startup the common code.
. /etc/rc.common
#
# InitWitangoEnv
# Use this procedure to set all environment variables to be used by the
# witangod process
#
InitWitangoEnv ()
{
# Set up environment for Witango server
#
WITANGO_PATH=/Applications/Witango/Server
export WITANGO_PATH
WITANGO_CONFIG=$WITANGO_PATH/configuration
export WITANGO_CONFIG

# Set up environment for Java Virtual Machine
#
JAVA_HOME=/System/Library/Frameworks/JavaVM.framework/Versions/1.4.1/
Libraries/
export JAVA_HOME
LD_LIBRARY_PATH=/usr/lib:/usr/local/lib:$WITANGO_PATH:$JAVA_HOME
export LD_LIBRARY_PATH
CLASSPATH=$WITANGO_PATH/jdbc_drivers/mysql-connector-java-3.0.6-
stable-bin.jar:
$WITANGO_PATH/jdbc_drivers/:
$WITANGO_PATH/jdbc_drivers/msbase.jar:
$WITANGO_PATH/jdbc_drivers/mssqlserver.jar:
$WITANGO_PATH/jdbc_drivers/msutil.jar:
$WITANGO_PATH
export CLASSPATH
# Set up environment for ODBC driver manager
#
ODBCINI=/Library/ODBC/odbc.ini
export ODBCINI
ODBCINSTINI=/Library/ODBC/odbcinst.ini
export ODBCINSTINI

# Set up environment for Oracle Call Interface
#
ORACLE_HOME=/Library/Application\ Support/Oracle/OCI/8.1.7.1
export ORACLE_HOME
ORA_NLS33=$ORACLE_HOME/ocommon/nls/admin/data
export ORA_NLS33
TNS_ADMIN=$ORACLE_HOME/network/admin
export TNS_ADMIN

# Set up general environment variables
#
DYLD_LIBRARY_PATH=/usr/lib:/usr/local/lib:/Library/Application\
Support/Oracle/OCI/8.1.7.1/lib
export DYLD_LIBRARY_PATH

```

```
PATH=$WITANGO_PATH:$PATH
export PATH
LC_ALL=en_US
export LC_ALL
}
StartService ()
{
if [ "${WITANGO5:=-NO-}" = "-YES-" ]; then
ConsoleMessage "Starting Witango Server"
InitWitangoEnv
/Applications/Witango/Server/witangod -c "WitangoServer1"
/Applications/Witango/Server/witangod -c "WitangoServer2"
/Applications/Witango/Server/witangod -c "WitangoServer3"
/Applications/Witango/Server/witangod -c "WitangoServer4"
fi
}

StopService ()
{
ConsoleMessage "Stopping Witango server"
/Applications/Witango/Server/witangod -c "WitangoServer1" -k
/Applications/Witango/Server/witangod -c "WitangoServer2" -k
/Applications/Witango/Server/witangod -c "WitangoServer3" -k
/Applications/Witango/Server/witangod -c "WitangoServer4" -k
/bin/sleep 3
}

RestartService ()
{
StopService
StartService
}

RunService "$1"
```

For more information

For more information see the following URLs:

Mac OS X Server: How to Locate and Edit Configuration Files

<http://docs.info.apple.com/article.html?artnum=106619>

The Boot Sequence

http://developer.apple.com/documentation/MacOSX/Conceptual/SystemOverview/BootingLogin/chapter_4_section_2.html#//apple_ref/doc/uid/20000981/CJBDEDDD

Customization Techniques

http://developer.apple.com/documentation/MacOSX/Conceptual/SystemOverview/BootingLogin/chapter_4_section_4.html#//apple_ref/doc/uid/20000981/CJBEABGI

Startup sequence control

Which background processes startup when OS X boots up is controlled by the `/etc/hostconfig` file. This file is located in a hidden directory and contains a flags that control which processes start and which don't.

A `-YES-` will start the process a `-NO-` will not start the process. You will see the `Witango 5` entry towards the end of the file.

A typical setup for this file looks like this:

```
##
# /etc/hostconfig
##
# This file is maintained by the system control panels
##

# Network configuration
HOSTNAME=-AUTOMATIC-
ROUTER=-AUTOMATIC-

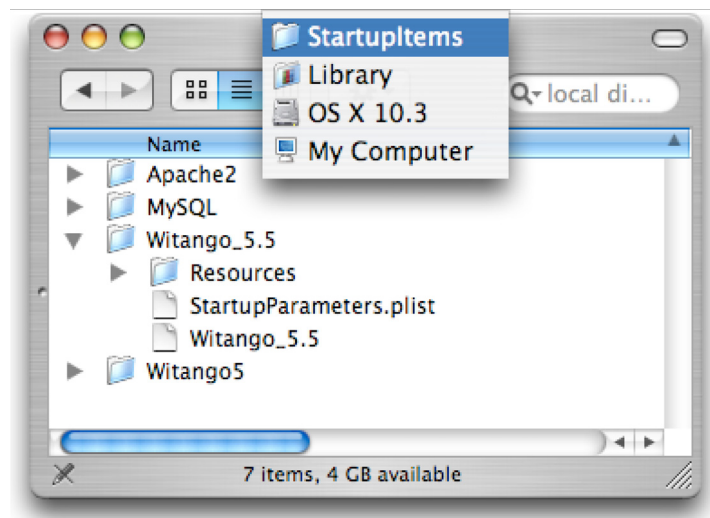
# Services
AFP_SERVER=-YES-
APPLETALK=-NO-
AUTH_SERVER=-NO-
AUTOMOUNT=-YES-
CONFIG_SERVER=-NO-
CUPS=-YES-
IP_FORWARDING=-NO-
IPV6=-YES-
MAIL_SERVER=-NO-
NETBOOT_SERVER=-NO-
NETINFO_SERVER=-AUTOMATIC-
NIS_DOMAIN=-NO-
RPC_SERVER=-AUTOMATIC-
TIME_SYNC=-NO-
QTSS_SERVER=-NO-
SSH_SERVER=-NO-
WEB_SERVER=-YES-
SMB_SERVER=-YES-
DNS_SERVER=-NO-
CRASH_REPORTER=-YES-
WITANGO5=-YES
```

Configuring Witango Server to Connect to FileMaker Client

At the time this Guide was prepared there exists a limitation in the implementation of AppleEvents on OS X that prevents a background process sending events to another process if the two processes are owned by a different users. To be able to use FileMaker Client as a datasource to the Witango Server the Witango server process and the FileMaker Client must be run in the same user space. To achieve this you will need to modify the startup sequence of the Witango server process that the installer configures.

Remove the startup script

The first step is to delete the startup script that starts the Witango Server at system start up. This is done by deleting the `/Library/StartupItems/Witango_5.5` directory and its content.



Create a startup command script

Create a command file (e.g. StartWitangoForFM.command) with a text editor. Command files are double clickable shell scripts that will run in a terminal window. Now add the following commands to it:



Note This example file can be found in the /Applications/Witango/Server/install directory. The command script when run in a shell will open FileMaker Pro and tell it to open the Sample02.fp5 database without a password. You can add other similar commands to this script to have FileMaker open other databases automatically for you as the user logs on.

```
#!/bin/sh

# Witango Application Server Startup Command
# Sample of how to start FileMaker and Witango from a command line
# in the same user space.

#####
####  Start FileMaker and Open Databases  ####
####  Modify these lines to automatically  ####
####  start FileMaker Pro databases        ####
####  osascript is used to send AppleEvents ####
####  it uses an AppleScript syntax to do  ####
####  to do this. Type 'man osascript' in  ####
####  the terminal to get more info.      ####
#####
echo "Starting FileMaker"
/usr/bin/osascript -e 'Tell application "FileMaker Pro" -e \
'activate' -e \
'open file "Mac HD:Applications:FileMaker Pro 5.5 Folder:FileMaker
Tutorial:Sample Files:Sample02.fp5" with password ""' -e \
'end tell'
/usr/bin/osascript -e 'Tell application "Finder" -e \
'activate' -e \
'end tell'
#####
####  set environment variables              ####
####  The following lines setup the variables ####
####  the server need to locate the odbc    ####
####  initialization file                   ####
#####
echo "Initialise user environment"
ODBCINI=/Library/ODBC/odbc.ini
export ODBCINI
ODBCINSTINI=/Library/ODBC/odbcinst.ini
export ODBCINSTINI
```

```
#####
####  Start the Witango Application Server  ####
####  The witango server is started with a -u ####
####  option to force the process to be owned ####
####  by the logged in user.  The & releases  ####
####  the terminal so it can be reused        ####
####  The next lines show the process info    ####
####  for the FileMaker and Witango processes #####
#####
echo "Starting Witango Server in user environment"
/Applications/Witango/Server/witangod -u &
ps -aux | grep witangod | grep -v grep
ps -aux | grep FileMaker | grep -v grep
```

There are 4 sections to this script.

- 1 The first line (`#!/bin/sh`) is a special sequence that tells the operating system which program to use to execute the commands with
- 2 The FileMaker Startup section
- 3 The environment variable setup section
- 4 The Start Witango Server section

The script is commented so you can find each of these sections and modify it to suit your environment.

Make sure that you save the file as a unix text file.



Caution If the file is saved as a classic OS 9 text file in MacRoman format, OS X will be unable to execute the command file.

Next make sure it has execute permissions on the file (See <http://docs.info.apple.com/article.html?artnum=10612> for information on changing file permissions) for the user that will use the script.

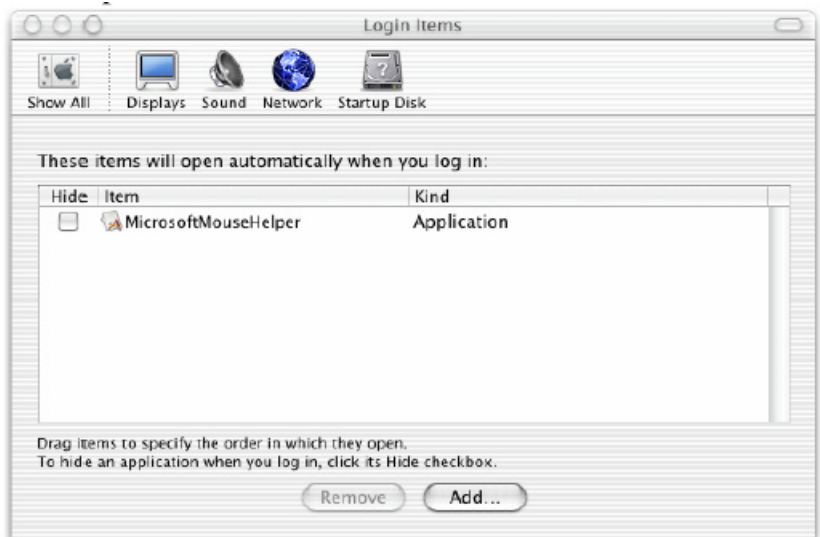
Make the startup command run automatically on login

Now you need to make this command file execute every time the current user logs in.

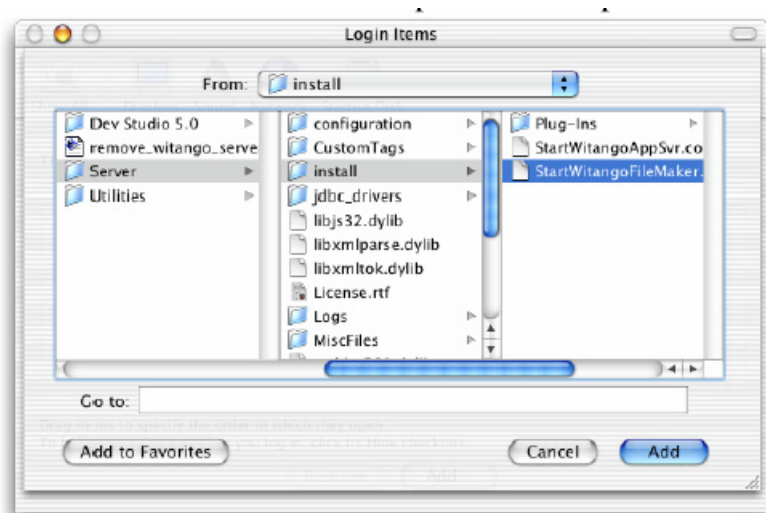
- 1 Go to the System Preferences.



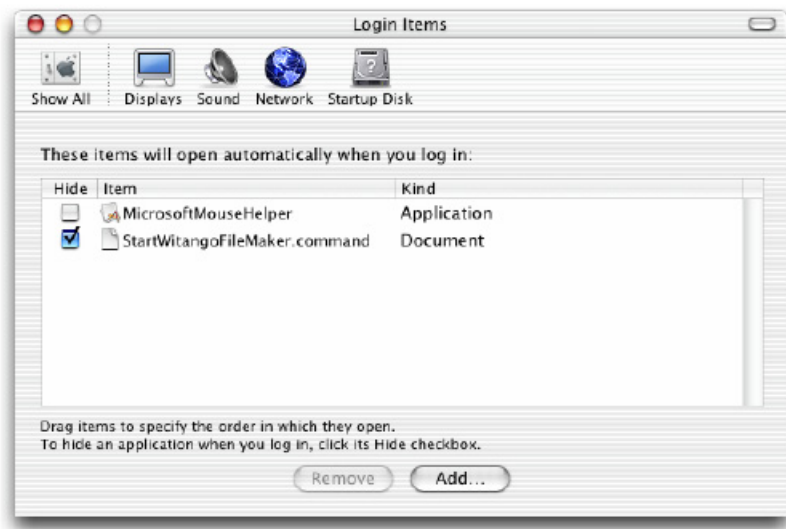
2 Open the “Login Items” pane and click on the “Add” button



3 Select the command file that was saved in the previous step and click the “Add” button.



The command file will now appear in the list of login items that will automatically start when the user logs in.



Log out and then log back in. Witango server and FileMaker should automatically start.

Configuring Witango Server with Java Integration

The Witango Server is able to use the Java Virtual Machine (JVM) to extend the Server capabilities with JDBC and Java Beans. For the Server to be able to load the JVM the correct environment variables must be set up in the script that starts the Witango Application Server. This is typically found in the following directory:

`/Library/StartupItems/Witango_5.5`

The J2RE implementation must be version 1.4.2 or greater for the Server to be able to load the JVM.



Note If you have configured the Server to use FileMaker Client as a data source you will need to add these variables to the command file you use to start the Witango Server.

Configuring the Java environment for Witango Server

The following environment variables must be set for the Java Virtual Machine to be able to start up successfully.

`JAVA_HOME`

`LD_LIBRARY_PATH`

`CLASSPATH`

Consult your Java documentation for a complete explanation of these environment variables:

<http://www.apple.com/java>

<http://developer.apple.com/java>

Typical Java environment variables for Witango Server

The following environment variable would configure the Java environment to use the 1.4.2 JVM and provide access to the mysql and MS SQL jdbc drivers.

```
JAVA_HOME=/System/Library/Frameworks/JavaVM.framework/Versions/1.4.2/Libraries/
```

```
export JAVA_HOME
```

```
LD_LIBRARY_PATH=/usr/lib:/usr/local/lib:$WITANGO_PATH:$JAVA_HOME
```

```
export LD_LIBRARY_PATH
```

```
CLASSPATH=$WITANGO_PATH/jdbc_drivers/mysql-connector-java-3.0.6-stable-bin.jar:$WITANGO_PATH/jdbc_drivers/mysql:$WITANGO_PATH/jdbc_drivers/mssql/msbase.jar:$WITANGO_PATH/jdbc_drivers/mssql/mssqlserver.jar:$WITANGO_PATH/jdbc_drivers/mssql/msutil.jar
```

```
export CLASSPATH
```



Caution These above commands should not be split over multiple lines when used in a shell script.

JAVA VIRTUAL MACHINE parameter in witango.ini

The `witango.ini` file located in the configuration directory of the Server will also need to be modified so the server can locate the correct JVM to load. Locate the JAVA VIRTUAL MACHINE parameter and set it to the path where the JVM has been installed. This is typically as follows:

```
/System/Library/FrameworksJavaVM.framework/JavaVM
```

Configuring Witango Server for JDBC

The JVM interface is loaded when the server starts up. If the JVM functionality fails to load an error message will be written to the `witangoevents.log` file in the configuration directory.

If the JVM loads successfully, you will see the following message:

```
Java VM Service is enabled.
```

If the JVM fails to load you will see the following message:

```
Unable to load Java VIRTUAL MACHINE, Java VM service is disabled.
```

OS X Java Documentation

For full Java for OS X documentation please consult the following sites:

```
http://www.apple.com/java
```

```
http://developer.apple.com/java
```

Setting up a JDBC driver

JDBC drivers may only be used once the JVM has been loaded by the Witango Server. If the JVM fails to load so too will the JDBC interfaces fail to load.

Configure the Classpath

The Witango Server can only use a JDBC driver once the driver files have been added to the CLASSPATH in the Witango Servers startup script (See Witango Server Startup process on page 20).

Before you attempt to setup a JDBC connection you should consult the documentation that came from the supplier of the JDBC driver. It should as a minimum provide you with the driver class and connection string information. The connection string informations should be used to form the URL to access the JDBC connection to the database.

e.g. A connection to a database using MySQL's Connector/J may have the following parameters:

DriverClass	com.mysql.jdbc.Driver
Connection Type	jdbc:mysql
Host	127.0.0.1
Port	3306
Database Name	Testdb

This information would translate to the following `jdbc.ini` settings:

```
<DriverClass>com.mysql.jdbc.Driver</DriverClass>  
<URL>jdbc:mysql://127.0.0.1:3306/Testdb</URL>
```

These settings are then added to the Server's `jdbc.ini` file which is located in the server's configuration directory. This file has the same form as the `jdbc.ini` file used by the Witango 5.5 Development Studio which has a graphical interface to add/delete/modify and test a JDBC datasource. A screen shot of this interface is shown below:

The screenshot shows the 'JDBC Datasource' configuration window. It contains the following fields and controls:

- Name:** Testdb
- Driver Class:** com.mysql.jdbc.Driver
- URL:** jdbc:mysql://127.0.0.1:3306/Testdb
- Properties:** A table with two columns: Name and Value.
- Testing...** A dropdown menu.
- Delete**, **Cancel**, and **Modify** buttons.
- Username:** root
- Password:** A field with 10 dots representing a masked password.
- Results:** Connection established.
- Test** button.

The jdbc.ini file

The `jdbc.ini` file is a simple XML structure. It is saved as a UNIX text file with Latin-1 encoding.



Caution If the file is saved as an OS 9 text file with Mac Roman encoding the server will fail to parse the datasources.

The structure for the `jdbc.ini` file looks like this:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE JDBCINI SYSTEM "jdbcini.dtd" >
<JDBCINI Version="0x00010000">
  <DataSources>
    <DataSource ID=" ">
      <DSN></DSN>
      <DriverClass></DriverClass>
      <URL></URL>
      <Properties>
        <Property ID=" ">
          <Name></Name>
          <Value></Value>
        </Property>
      </Properties>
    </DataSource>
  </DataSources>
</JDBCINI>
```

Note: The Datasource ID is the same as the DSN name.

Note: The Property ID is the same as the Name value.

Configuring Witango Server for ODBC

iODBC

The server dynamically loads the odbc driver manager specified in the ODBCDBLIBRARY parameter of the witango.ini file. The default setting for this parameter is the location of the iodbc driver manager that ships with OS X 10.2.x. This driver manager is located at `/usr/lib/libiodbc.2.1.6.dylib`. It is not recommended to use a symbolic link in this parameter as unpredictable results may occur.

If the specified driver manager is loaded the event is recorded in the `witangoevents.log` file.

```
[ 412] 2003-05-22 10:38:01 START      INFO      Initialized ODBC
environment
```

To ensure correct behaviour of the iODBC driver manager it is recommended that the following environment variables be set in the startup script of the Witango Server. These environment variable should point to the files that you store your odbc data source and odbc driver information in.

```
ODBCINI=/Library/ODBC/odbc.ini
export ODBCINI
ODBCINSTINI=/Library/ODBC/odbcinst.ini
export ODBCINSTINI
```

The above example would instruct iODBC to use the `odbc.ini` and `odbcinst.ini` files in the `/Library/ODBC/` directory.

Further information on iODBC

It is recommended that you read the iODBC documentation supplied by OpenLink Software regarding iodbc on OS X:

<http://www.openlinksw.com/support/macosex-faq.htm>

Support for iodbc is via a web forum:

<http://www.iodbc.org/wwwboard/wwwboard.html>

Configuring Witango Server for Oracle Call Interface (OCI)

With the Witango Application Server and Dev Studio for OS X have an updated OCI interface which is now optimised for the Oracle 8i v8.1.7.1 OCI for Mac OS X driver. This delivers a faster and more reliable interface for connecting to Oracle databases.

To install the OCI Driver on OS X 10.3 following step by step procedure.

Install the OCI driver

- 1 Download the oci driver driver file (MacOSX_8171.cpio) from:

```
http://otn.oracle.com/software/htdocs/  
distlic.html?/software/tech/java/sqlj_jdbc/htdocs/  
macsoft.html
```

Before you can download this driver you will need to register with the site. Membership is free.

- 2 Create a new folder.
- 3 Drag the MacOSX_8171.cpio file into that folder.
- 4 Open a terminal window.
- 5 Type in `cd` followed by a space.
- 6 In the Finder navigate to the folder you placed the MacOSX_8171.cpio file in and drag the folder onto the terminal window (this should put the path to the directory in the terminal window) and hit return.
- 7 In the terminal window, type:

```
cpio -idvnc < MacOSX_8.1.7.1.cpio
```
- 8 The command in step 7 will result in three files being generated:
 - **Oracle_8.1.7.1_Client.zip** which will expand into Oracle_8.1.7.1_Client Folder;
 - **MacOSX_8.1.7.1_OCI_Demo.zip** which will expand into ocidemo folder; and
 - **Release_Notes.zip** which will expand into a Release notes folder
- 9 Doubleclick the Oracle_8.1.7.1_Client.zip file or drag its image over the icon for Stuffit Expander.
- 10 In the `/Library/` directory at the root of OS X boot disk, create an Oracle directory.
- 11 Inside that directory, create another, called OCI.

- 12 Inside OCI, create another called 8.1.7.1.
- 13 Inside 8.1.7.1, place all of the contents from Oracle_8.1.7.1_Client Folder.
- 14 Edit /Library/Oracle/OCI/8.1.7.1/network/admin/tnsnames.ora to match your oracle setup

Set up the Server's OCI environment

- 1 The server needs to know where to find the OCI driver library to load. The OCI_LIB_PATH=parameter in the witango.ini file should be configured with the absolute path to the directory where the OCI dylib file has been installed. If you have been using the paths in the previous section this would be /Library/Oracle/OCI/8.1.7.1/lib/
- 2 For the library to load successfully you will need to configure four environment variables in the witango server's startup script. If you are using the standard startup script that the installer created make sure that the following lines are present and are not commented out:

```
ORACLE_HOME=/Library/Oracle/OCI/8.1.7.1
export ORACLE_HOME
ORA_NLS33=$ORACLE_HOME/ocommon/nls/admin/data
export ORA_NLS33
TNS_ADMIN=$ORACLE_HOME/network/admin
export TNS_ADMIN
DYLD_LIBRARY_PATH=/Library/Oracle/OCI/8.1.7.1/lib:/usr/lib:/usr/local/lib
export DYLD_LIBRARY_PATH
```

Development Studio Setup for the OCI driver

- 1 We now need to create a file to automatically set some environment variable for the oracle driver. If it does not exist already, in your home directory (example: ~ is the path to your home directory), create a directory named "~/.MacOSX", (don't forget the period "~" before the MacOSX)
- 2 Create a text file named environment.plist and copy and paste the text of the sample environment.plist into it (see below). Save the file to the ~/.MacOSX/ directory. You may need to edit the file paths to match the location where you installed the OCI driver.

If you do not wish to use the environment.plist, you will need to set the following environment variables and values with an appropriate mechanism:

```
ORACLE_HOME /Library/Oracle/OCI/8.1.7.1
DYLD_LIBRARY_PATH /Library/Oracle/OCI/8.1.7.1/lib
ORA_NLS33 /Library/Oracle/OCI/8.1.7.1/ocommon/nls/admin/data
TNS_ADMIN /Library/Oracle/OCI/8.1.7.1/network/admin
```


- 3 For the Dev Studio to be able to access the native unix OCI driver shared library, you need to make an OracleOCI framework. To do this type the following commands into a terminal window. You will need to change the paths if you did not use the paths in the previous steps. This involves two steps, performed in your terminal window:

- Make OracleOCI.framework directory, type the following in your terminal:

```
sudo mkdir /Library/Frameworks/OracleOCI.framework
```

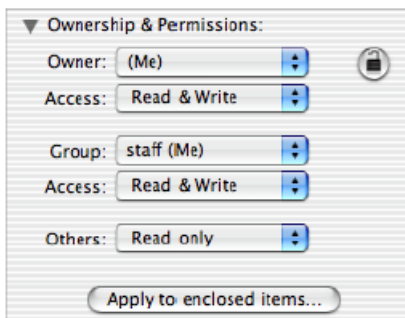
- Make a link to libcIntsh.dylib, type the following in your terminal:

```
sudo ln -s /Library/Oracle/OCI/8.1.7.1/lib/libcIntsh.dylib  
/Library/Frameworks/OracleOCI.framework/OracleOCI
```

Tightening the permissions on the OCI Driver

This is an optional step to secure the permissions on the OCI installation.

In order to do this, change the permissions on /Library/Frameworks/OracleOCI.framework, /Library/Oracle and ensure that read/write access has been given to the file owner and staff. (the user 'witango' which owns the witangod server process if it is installed should belong to the 'staff' group, this is automatic upon user creation



Sample ~/.MacOS/environment.plist

```
<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE plist SYSTEM "file:///localhost/System/Library/DTDs/PropertyList.dtd">  
<plist version="0.9">  
<dict>  
  
    <key>DYLD_LIBRARY_PATH</key>  
    <string>/Library/Oracle/OCI/8.1.7.1/lib</string>  
    <key>ORACLE_HOME</key>  
    <string>/Library/Oracle/OCI/8.1.7.1</string>  
    <key>ORA_NLS33</key>
```

```

                                <string>/Library/Oracle/OCI/8.1.7.1/ocommon/nls/
admin/data</string>
                                <key>TNS_ADMIN</key>
                                <string>/Library/Oracle/OCI/8.1.7.1/network/admin</
string>
                                <key>WITANGO_PATH</key>
                                <string>/Applications/Witango/Server</string>

</dict>
</plist>

```

Sample tnsnames.ora file

```

# This file contains the syntax information for
# the entries to be put in any tnsnames.ora file
# The entries in this file are need based.
# There are no defaults for entries in this file
# that Sqlnet/Net3 use that need to be overridden
#
# Typically you could have two tnsnames.ora files
# in the system, one that is set for the entire system
# and is called the system tnsnames.ora file, and a
# second file that is used by each user locally so that
# he can override the definitions dictated by the system
# tnsnames.ora file.
#
# The entries in tnsnames.ora are an alternative to using
# the names server with the onames adapter.
# They are a collection of aliases for the addresses that
# the listener(s) is(are) listening for a database or
# several databases.
#
# The following is the general syntax for any entry in
# a tnsnames.ora file. There could be several such entries
# tailored to the user's needs.

ORCL =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = 192.168.73.3) (PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = ORCL)
    )
  )

```

Manually Configuring Apache Web Server

Before your Apache web server will serve Witango files, you must configure your Apache Web server to use it with the Witango Server plug-in (witango55_apache1.so or witango55_apache2.so). By default, these plug-ins are located in \$WITANGO_PATH/Install/Plugins/.

Configuring Apache 1.3

To build a new Apache 1.3.x Web server with the Witango Server plug-in



Note The Apache 1 plugin will only work with the Apache 1.3.26 or later due to changes for Apache API.

- 1 Copy \$WITANGO_PATH/Install/Plugins/witango55_apache1.so to your Apache modules directory.
- 2 Add the following lines to the httpd.conf file.

```
Add Module WitangoModule <Apache module directory path>/
witango55_apache1.so
```

```
Load Module WitangoModule
```

- 3 Ensure that the permissions on the witango55_apache1.so file are appropriate for your system apache configuration.

Configuring Apache 2.0.x

To set up the new Apache 2.0.44 Web server to use the Witango Server plug-in



Note The Apache 2 plugin will only work with the Apache 2.0.44 or later due to changes for Apache2 API.

- 1 Copy \$WITANGO_PATH/Install/Plugins/witango55_apache2.so to your Apache modules directory.
- 2 Add the following lines to the httpd.conf file.

```
Add Module WitangoModule <Apache module directory path>/
witango55_apache2.so
```

```
Load Module WitangoModule
```

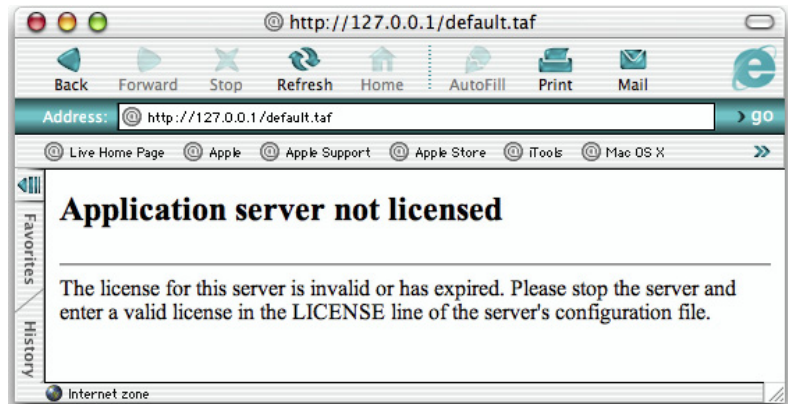
Ensure that the permissions on the witango55_apache2.so file are appropriate for your system apache configuration.



Note The mod_mime_magic module may interfere with the mime types that the wapache module registers with the server. The mod_mime_magic looks at the contents of the taf and overrides the wapache settings which will send the XML of the taf back to the browser. To overcome this issue you can modify the mod_mime_magic module Settings or make the wapache module load before the mime_magic module.

Troubleshooting

The Unlicensed Server Message



Your web browser may return this message when you attempt to access your web site. This indicates that either:

- 1 The 30 day evaluation period for the installation has expired.

You must purchase a valid serial number to continue operation of the software. Witango software licenses can be purchased on-line at <http://www.witango.com>

- 2 An invalid serial number has been entered.

This can be rectified by editing the `witango.ini` file

Use a text editing utility to edit the `witango.ini` file. Ensure that the file is saved as a Unix text file and not a Mac OS 9 text file.

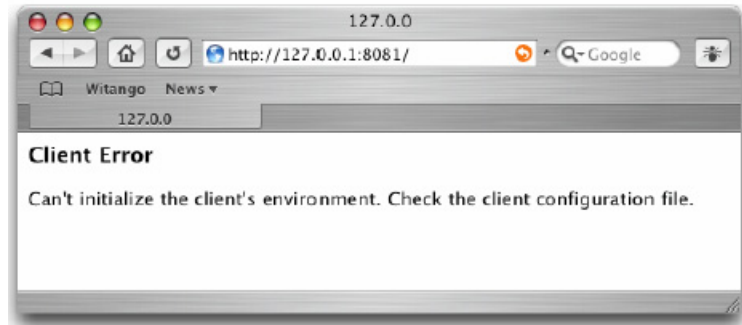
`/Applications/WitangoServer/5.5/configuration/witango.ini`

Locate the line in the file that starts:

```
LICENCE= <blank space or invalid serial number will be here>
```

Replace the text after `LICENCE=` with the valid serial number.

The Web Server plug-in does not connect to the Server



This error is caused by the witango client not being able to read the clients.ini file. There are 2 causes for this error:

- 1 The clients.ini file is missing, incorrectly formatted or has not been saved as a Mac OS X unix text file.
- 2 A typical clients.ini file looks like this and must be saved a text file with unix style line breaks. If it is saved with Macintosh line breaks the client will not be able to parse the file.

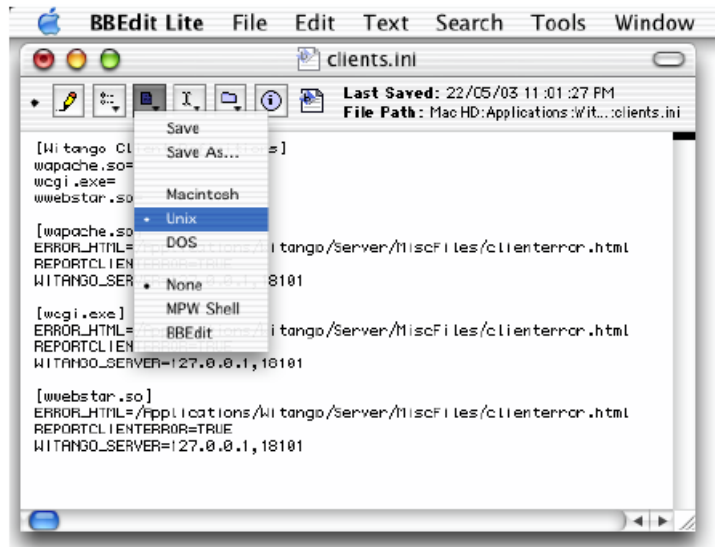
```
[Witango Client Definitions]
witango55_apache1.so=
wsgi.exe=
webstar.so=

[witango55_apache1.so]
ERROR_HTML=/Applications/WitangoServer/5.5/MiscFiles/clienterror.html
REPORTCLIENTERROR=TRUE
WITANGO_SERVER=127.0.0.1,18100

[wsgi.exe]
ERROR_HTML=/Applications/WitangoServer/5.5/MiscFiles/clienterror.html
REPORTCLIENTERROR=TRUE
WITANGO_SERVER=127.0.0.1,18100

[witango55_webstar5.so]
ERROR_HTML=/Applications/WitangoServe/5.5r/MiscFiles/clienterror.html
REPORTCLIENTERROR=TRUE
WITANGO_SERVER=127.0.0.1,18100
```

If you are unsure how the file was saved open it in a text editor like BBEdit and check the options of the file as shown below.



The permissions on of the clients.ini file or a directory leading to the clients.ini file is preventing the web server process from reading the file

The witango client (plug in) loads in the process space of the web server so the user that owns the web server process must be able to read the clients.ini file in /Applications/Witango/Server/configuration.

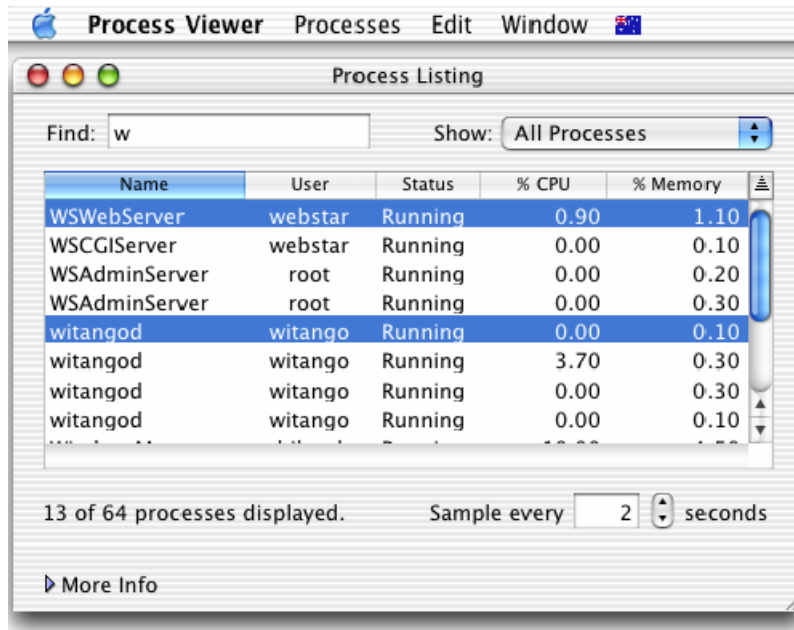
You can see the permissions on the directory hierarchy of the Witango server by running the following command in the terminal window.

```
ls -al /Applications/WitangoServer
ls -al /Applications/WitangoServer/5.5
ls -al /Applications/WitangoServer/5.5/configuration
```

Typical output for the directory listing should look like this:

```
[UserName:~] UserName % ls -al /Applications/WitangoServer
total 32
drwxrwxrwx  8 UserName  admin   272 May 20 16:57 .
drwxrwxr-x 43 root      admin   1462 May 22 02:13 ..
-rwxrwxrwx  1 UserName  admin   6148 May 20 16:57 .DS_Store
drwxrwxr-x  7 root      staff    238 May 20 00:12 Dev Studio 5.0
drwxrwxr-x 20 root      staff    680 May 20 12:49 Server
drwxrwxr-x  8 root      staff    272 May 19 10:40 Utilities
```

To work out which user owns the web server process open the Process Viewer in your /Applications/Utilities directory. This will list all the processes running on your system.



Use the Find field to narrow down the number of processes in the list. For Apache type “htt” and for WebStar type “w”. The user name that corresponds to the web server process is the owner of the processes. This user must be able to read the clients.ini file.

Once you have the name of the user, in a terminal window type the following command substituting in the correct user name:

```
id <username>
```

This will output the following information about the user. The following example for the webstar user shows that webstar is also a member of the staff group.

```
uid=504(webstar) gid=20(staff) groups=20(staff)
```

Now you need to check that the user and/or group of the web server user can access the clients.ini file. Be careful not to change permissions on the directories or files that will prevent the witango user from having read and write access to the directories.

A typical fix for file permissions would be like this:



Note You will require an administrative password to run these commands in the terminal

```
sudo chown -R witango /Applications/WitangoServer/5.5
sudo chgrp -R staff /Applications/WitangoServer/5.5
sudo chmod -R 775 /Applications/WitangoServer/5.5
sudo chmod 770 /Applications/WitangoServer/5.5/configuration/*
```

Now stop the web server. Check the process has terminated. Start the web server again. The witango client should now be able to initialise itself and the error should be gone.

If you are still getting the same error message, run the following commands:

```
sudo chown -R witango /Applications/WitangoServer/5.5
sudo chgrp -R staff /Applications/WitangoServer/5.5
sudo chmod -R 775 /Applications/WitangoServer/5.5
sudo chmod -R 770 /Applications/WitangoServer/5.5/configuration/
*.ini
sudo chmod -R 774 /Applications/WitangoServer/5.5/configuration/
clients.ini
```

Unable to connect to the specified data source

If you receive the following error message from the server while trying to access an ODBC data source it is most commonly caused by the driver manager finding the first `odbc.ini` file in its search path. This may not be the same `odbc.ini` file that you have configured.

Error

An error occurred while processing your request:

File: **testdb_o.taf**

Position: **RecordList**

Class: **Internal**

Main Error Number: **-4**

Unable to connect to the specified data source.

Verify that data source is properly configured and that database server is online.

To ensure the correct behaviour of the ODBC driver manager it is recommended that the following environment variables be set in the startup script of the Witango Server. These environment variable should point to the files that you store your odbc data source and odbc driver information in.

```
ODBCINI=/Library/ODBC/odbc.ini
export ODBCINI
ODBCINSTINI=/Library/ODBC/odbcinst.ini
export ODBCINSTINI
```

The above example would instruct iODBC to use the odbc.ini and odbcinst.ini files in the /Library/ODBC/ directory.

The server is starting up message

While the witango server is starting up it moves through several phases of operation. While the server is loading its support libraries and initialising its subsystems it will not accept requests from the witango clients. To allow the client to display a message to the end user the server sends back the following message. This is normal and as soon as the server has finished its initialisation phase it will start accepting user requests.

Error

An error occurred while processing your request:

File:

Position:

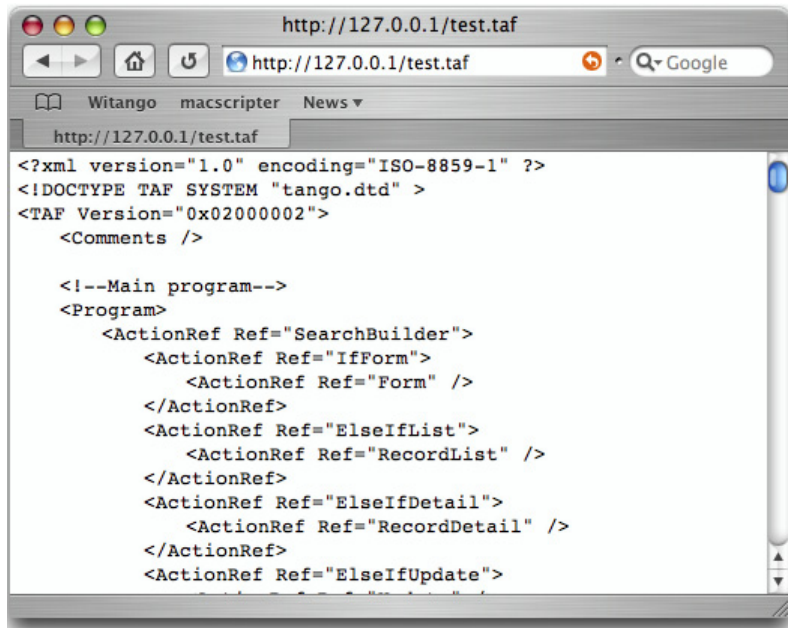
Class: Internal

Main Error Number: -1070

The server is starting up and can not process regular requests.

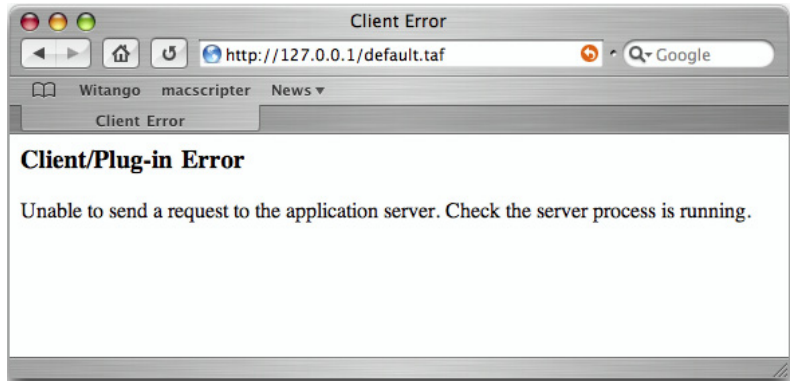
The XML of the taf file is returned

If you receive the XML content of your taf file when requesting a URL this is an indication that the web server plugin is not functioning or loaded. This will most likely be due to a misconfiguration or missing configurations. You should check you web server documentation on how to configure plugins. The Witango plugins for Apache and WebStar are located in the install directory of the server. A screenshot of the XML content of your taf file is shown below:

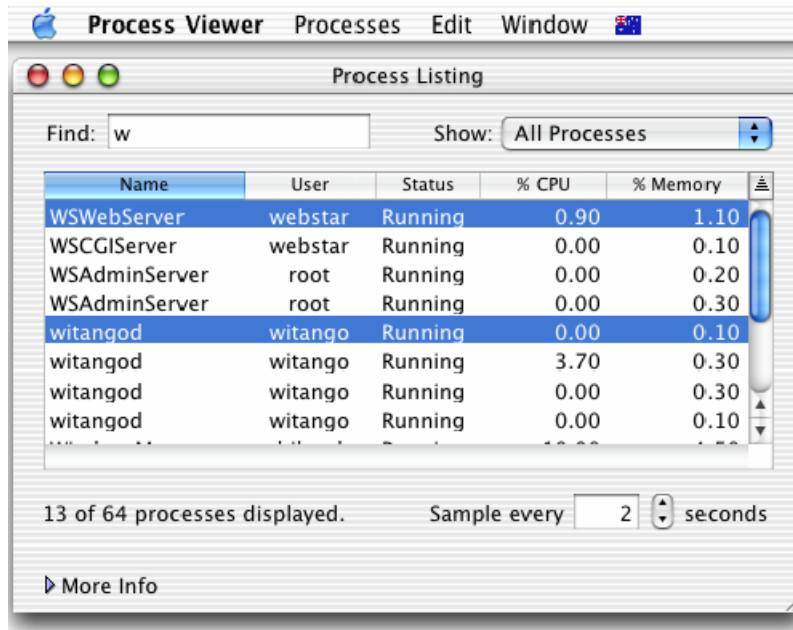


The server is not started

If the server has been configured to listen on a different port to a client, OR, the server process is not running, the above error message will be generated.



The administrator should first check that the Witango Server process is running in the Process Viewer as shown below.



If the Witango Server process is not visible in the list the administrator should start the server process as outlined in Witango Server Startup process on page 20.

If the process is running the administrator should check that the WITANGO_SERVER in the clients.ini file matches the values in LISTENERPORT and LISTENERADDRESS in the witango.ini file.

[wiis.dll]

ERROR_HTML=C:\\Program
Files\\WitangoServer\\5.5\\MiscFiles\\clienterror.html

REPORTCLIENTERROR=TRUE

WITANGO_SERVER=127.0.0.1,18100

LISTENERADDRESS=

LISTENERPORT=18100

The Unix Command Line

Changing unix file permissions

See Apple's Knowledge Base article "Mac OS X: Troubleshooting Permission Issues" at <http://docs.info.apple.com/article.html?artnum=10612>

Witango command line options

Witango Server has several command line options:

```
witangod [ -ksorvh ] [-u username] [-c config_name]
-k: Kill the running process
-s: Request status information of the running process
-u: Run process in user space 'username' (Default User:
witango), and not as a daemon
-o: Send output to standard out window
-r: Restart the daemon
-c: Start the daemon with 'config_name'
-v: Version Information
-h: This help message
If no arguments are present, process will attempt to start
as a daemon
```

Global User environment variables

If you need to set environment variable for all the users that will log onto the system, you can add these commands to the following file:

```
/etc/profile
```

Be sure to keep a backup of this file as Apple have in the past written over this file during system updates. If this occurs you will need to edit the new file and add the additional environment variable again.

A typical /etc/profile file would look like this:

```
# System-wide .profile for sh(1)

PATH="/bin:/sbin:/usr/bin:/usr/sbin"
export PATH

ODBCINI=/Library/ODBC/odbcinst.ini
export ODBCINI

ODBCINSTINI=/Library/ODBC/odbcinst.ini
export ODBCINSTINI
```

Useful terminal commands and what they do

Change to the Witango Server Directory

```
cd /Applications/WitangoServer/5.5
```

Start the Witango Server in a local user mode with output to the screen

```
/Applications/WitangoServer/5.5/witangod -uo
```

List all running Witango Server processes

```
ps -aux | grep witango | grep -v grep
```

Start the Witango Server via the startup script

```
sudo SystemStarter start WITANGO55
```

Stop the Witango Server via the startup script

```
sudo SystemStarter stop WITANGO55
```

View the last 512 bytes of the Witango log

```
tail /Applications/WitangoServer/5.5/configuration/witangoevents.log
```

Show the status of the most active processes

```
top -w -l 1 | grep witangod
```

Adding commands shortcuts to your terminal session

UNIX provides the ability to alias commands to make long or complex commands easy to remember and use. These aliases can be added to your users .tcshrc file so that they will be available with every terminal window. A few suggested commands to add to your ~/.tcshrc file are:

```
alias ll "ls -al"
```

```
alias wcd "cd /Applications/WitangoServer/5.5"
```

```
alias witango "/Applications/WitangoServer/5.5/witangod -u"
```

```
alias wpid "ps -aux | grep witango | grep -v grep"
```

```
alias wspid "ps -aux | grep webstar | grep -v grep"
```

```
alias wstart "sudo SystemStarter start WITANGO55"
```

```
alias wstop "sudo SystemStarter stop WITANGO55"
```

```
alias wlog "tail /Applications/WitangoServer/5.5/configuration/  
witangoevents.log"
```