

# Omnis Data Bridge

Allows cross-platform access to Omnis data files on Mac OS X servers

Omnis Software

April 2018

09-042018-06

The software this document describes is furnished under a license agreement. The software may be used or copied only in accordance with the terms of the agreement. Names of persons, corporations, or products used in the tutorials and examples of this manual are fictitious. No part of this publication may be reproduced, transmitted, stored in a retrieval system or translated into any language in any form by any means without the written permission of Omnis Software Ltd.

© Omnis Software Ltd, and its licensors 2018. All rights reserved.

Portions © Copyright Microsoft Corporation.

Regular expressions Copyright (c) 1986,1993,1995 University of Toronto.

© 1999-2018 The Apache Software Foundation. All rights reserved.

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

OMNIS® and Omnis Studio® are registered trademarks of Omnis Software Ltd.

Microsoft, MS, MS-DOS, Visual Basic, Windows, Windows 95, Win32, Win32s are registered trademarks, and Windows NT, Visual C++ are trademarks of Microsoft Corporation in the US and other countries.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world.

IBM, DB2, and INFORMIX are registered trademarks of International Business Machines Corporation.

ICU is Copyright © 1995-2003 International Business Machines Corporation and others.

UNIX is a registered trademark in the US and other countries exclusively licensed by X/Open Company Ltd.

Sun, Sun Microsystems, the Sun Logo, Solaris, Java, and Catalyst are trademarks or registered trademarks of Sun Microsystems Inc.

J2SE is Copyright (c) 2003 Sun Microsystems Inc under a licence agreement to be found at: <http://java.sun.com/j2se/1.4.2/docs/relnotes/license.html>

MySQL is a registered trademark of MySQL AB in the United States, the European Union and other countries ([www.mysql.com](http://www.mysql.com)).

ORACLE is a registered trademark and SQL\*NET is a trademark of Oracle Corporation.

SYBASE, Net-Library, Open Client, DB-Library and CT-Library are registered trademarks of Sybase Inc.

Acrobat is a trademark of Adobe Systems, Inc.

Apple, the Apple logo, Mac OS, and Macintosh are registered trademarks of Apple, Inc.

HP-UX is a trademark of Hewlett Packard.

OSF/Motif is a trademark of the Open Software Foundation.

CodeWarrior is a trademark of Metrowerks, Inc.

This software is based in part on ChartDirector, copyright Advanced Software Engineering ([www.advsofteng.com](http://www.advsofteng.com)).

This software is based in part on the work of the Independent JPEG Group.

This software is based in part of the work of the FreeType Team.

Other products mentioned are trademarks or registered trademarks of their corporations.

# Table of Contents

<b>INTRODUCTION .....</b>	<b>4</b>
<b>CONFIGURATION .....</b>	<b>5</b>
<i>Note to existing ODB users .....</i>	<i>5</i>
<i>Omnis SQL and DML connections .....</i>	<i>5</i>
<i>Ancillary files.....</i>	<i>5</i>
<i>Multiple ODB Installations.....</i>	<i>5</i>
EDITING THE CONFIG FILE.....	6
<i>Adding Data Files.....</i>	<i>6</i>
<i>Setting the Port.....</i>	<i>12</i>
<i>Other configuration items .....</i>	<i>12</i>
<i>Error Configuration .....</i>	<i>12</i>
<b>USING THE ODB.....</b>	<b>13</b>
MANAGING THE ODB SERVER.....	13
<i>On Mac OS X .....</i>	<i>13</i>
<i>On Linux.....</i>	<i>14</i>
<i>On Windows.....</i>	<i>15</i>
<i>Debugging the ODB .....</i>	<i>16</i>
OPENING AN ODB DATA FILE .....	17
<i>Using DML.....</i>	<i>17</i>
<i>Using OmnisSQL .....</i>	<i>17</i>
<i>Using the SQL Browser .....</i>	<i>18</i>
<i>ODB Server preference.....</i>	<i>18</i>
<i>Omnis Datafile List.....</i>	<i>19</i>
ODB ERROR MESSAGES.....	19
<i>General Errors .....</i>	<i>19</i>
<i>Client Errors.....</i>	<i>20</i>
<i>Server Errors .....</i>	<i>21</i>

# Introduction

The **Omnis Data Bridge** (ODB) enables you to connect to and share native Omnis data files on Mac OS X servers, from clients across all platforms supported by Omnis. The Omnis Data Bridge works in Omnis Studio 3.3 and above, and Omnis 7.3 version 8.2 only: earlier versions of Omnis Studio or Omnis 7 do not support the ODB.

Mac OS X servers and some other Unix servers do not fully support the advisory byte locking employed by Omnis in a cross platform manner. The specific problem the ODB resolves is that of clients connected via one type of file protocol not respecting locks of clients connected via a different file protocol.

The Omnis Data Bridge is a background system daemon, that runs invisibly on a server and listens for requests from Omnis clients on a given port. The ODB uses TCP/IP and therefore bypasses the standard file protocols normally employed in a multi-platform environment. Omnis clients can open and use data files by specifying the ODB server's IP address and port instead of a file name and path to the Omnis data file, or the IP-address and port can be stored in the \$odbserver Omnis preference.

# Configuration

After installation, you must edit the configuration file before you can use the ODB. In the ODB folder you will find the configuration file named *config.xml*. The config.xml file contains a number of configuration items, including the details about the Omnis data files to be accessed by the ODB.

## Note to existing ODB users

In versions of the ODB prior to 1.69, error messages were stored in the sysconfig.xml file. With v1.69 onwards the errors are now hard-coded making this file redundant.

As of version 1.69, the dependency on libxerces has also been removed. The ODB now employs a built-in XML parser.

## Omnis SQL and DML connections

From Omnis Studio 5 onwards, connections to an Omnis datafile can now be made concurrently using both Omnis SQL and the Omnis Data Bridge using a single serial number quota. Omnis no longer treats host names and IP addresses from the same machine as different entities.

## Ancillary files

The ODB 1.75 on Windows installs all ancillary files into the following user directory:

```
C:\Users\[username]\AppData\Roaming\OmnisDataBridge1.75
```

On macOS the ancillary files are installed into the following user directory:

```
/Library/Application Support/Omnis/OmnisDataBridge 1.75
```

NB) When installing as *Administrator* on Windows, the ancillary files folder is located in `C:\Windows\system32\config\systemprofile\AppData\...`

## Multiple ODB Installations

It is possible to install multiple copies of the ODB on the same machine provided that certain conditions are followed, namely:

- Each ODB must maintain its own writable files (i.e. config.xml & messages.txt).
- Each ODB must manage its own data files; two ODBs cannot share the same file(s).
- Each ODB must run using a different port number to avoid a conflict.

### macOS

Using a central location for macOS writable files has implications if you want to use multiple ODB installations. As of version 1.75, the ODB looks in the **/Library/Application Support** folder whenever the ODB is installed to the **/Applications** folder. The recommended approach when using multiple ODB installations is to move each copy of the ODB to a root folder, e.g. `/opt/ODB175a`, `/opt/ODB175b`, etc. As a minimum, you should copy a separate `config.xml` file into each folder.

Hint: as long as the word “Applications” does not appear in the full path name of the odb executable, the ODB ignores the writable files folder and accesses its `config.xml` & `messages.txt` files locally (i.e. local to the odb executable).

### Windows

The same principle applies when the ODB is installed to the “Program Files” folder on Windows. If you need multiple copies of the ODB, you should create separate folders in a root directory, e.g. `c:\DataBridge\ODB175a`, `c:\DataBridge\ODB175b`, etc.

In this mode, the Windows ODB ignores the user’s AppData folder and accesses each `config.xml` and `messages.txt` locally (i.e. local to the `odbridge.exe`).

### Linux

The above discussion does not apply to Linux ODB installations, since on this platform the ODB always accesses its writable files locally (i.e. local to the odb executable). Thus to use multiple copies of the ODB, simply duplicate the ODB folder as needed, e.g. `/usr/local/odb175a`, `/usr/local/odb175b`, etc.

## Editing the config file

You can modify the `config.xml` file with the Property List Editor or any other XML editor or text editor. The Property List Editor is supplied by Apple and is installed as part of the developer tools.

### Adding Data Files

Before a data file can be accessed by the ODB, you need to edit the `config.xml` configuration file by adding details of the Omnis data file and its segments.

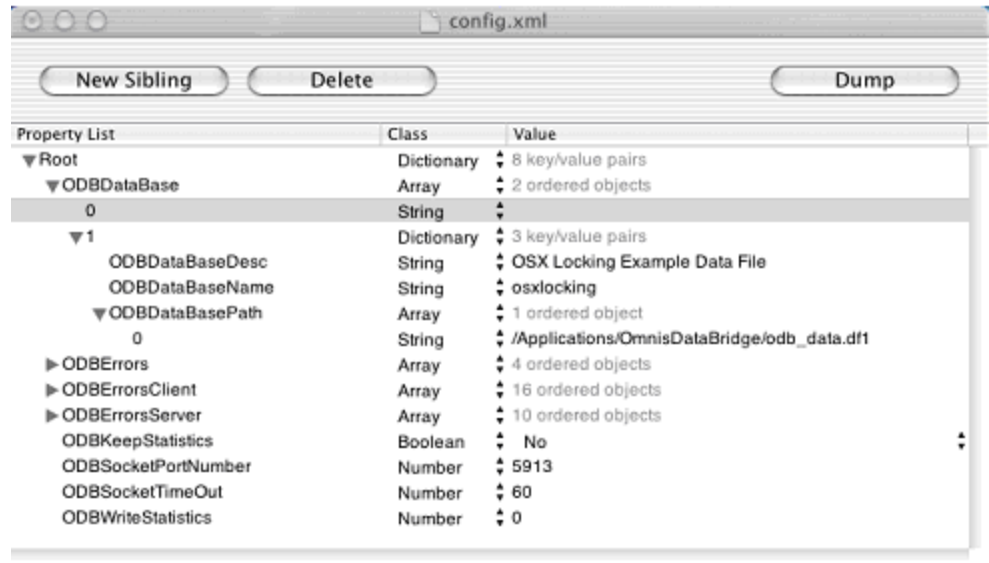
### Special Characters

Special characters such as " (double quote), ' (single quote), <, >, and &, require escaping for use in XML files. In the case of & (ampersand) you need to use `&amp;`; and similarly in the case of " (double quote) you need to use `&quot;`; and so on. Failing to escape such characters will cause a fatal error in the DOM parser code which is a vendor limitation.

## Using Property List Editor

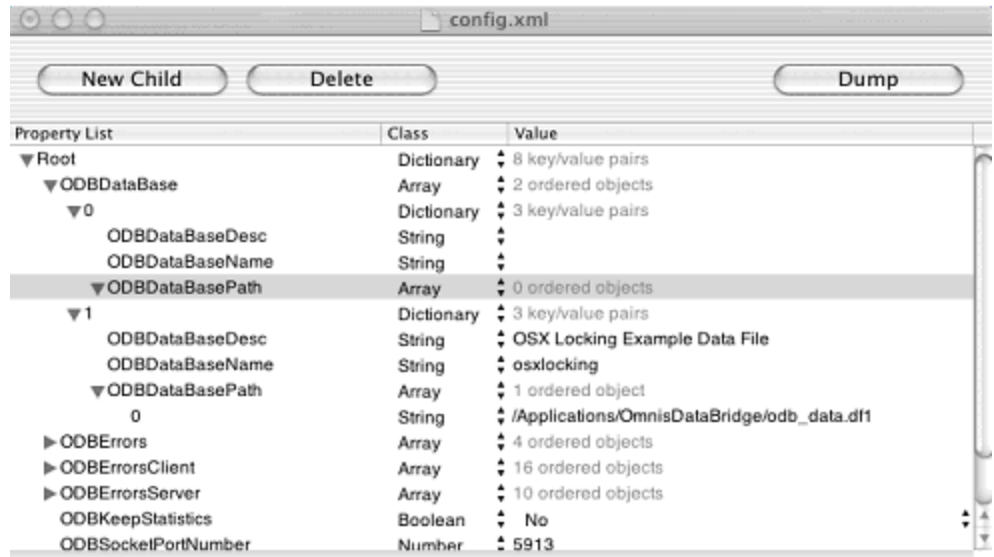
If you have the Property List Editor or a similar property editor installed on your system, you can simply double click the *config.xml* file to open it.

To add details of your data file, expand and select the key *ODBDataBase* and click on *New Child*.



You should now have a new array entry for your data file. There is no fixed limit to the number of data files you can add.

Change the *Class* of the entry to *Dictionary* and add three children to the array entry. The children must be named *ODBDataBaseDesc*, *ODBDataBaseName* and *ODBDataBasePath*. The *Class* of the three children should be *String*, *String* and *Array* respectively.



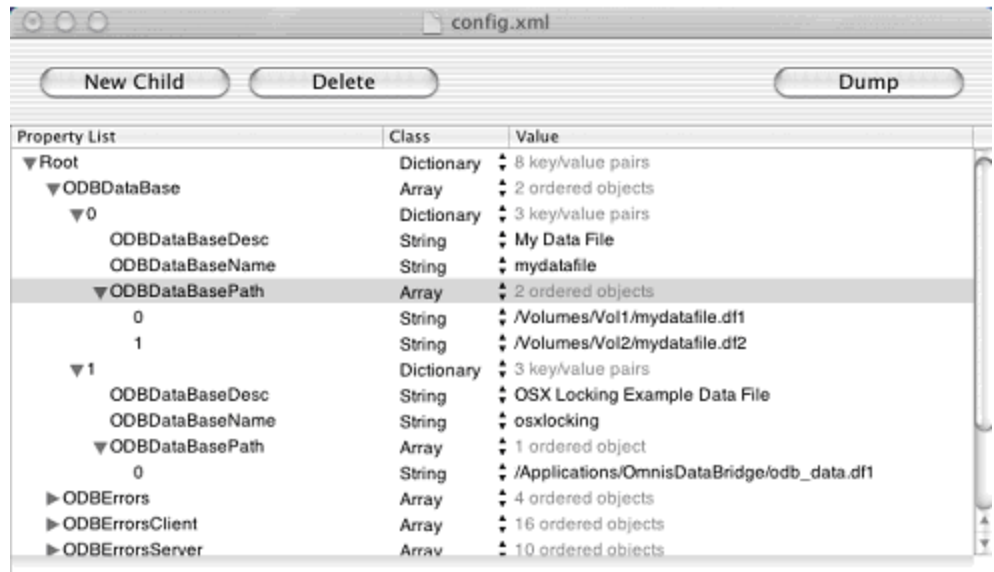
For the value of *ODBDataBaseDesc* you should enter a sensible description for the data file. This description is displayed to the user if you use *Prompt for data file* to allow the user to select a data file from a list supplied by ODB.

The value of *ODBDataBaseName* must be a unique name that you will use to identify the data file when opening a data file using the *Open data file* command.

The last item *ODBDataBasePath* is an array of full paths pointing to your data file segments on disk. You must enter a full path for each data file segment if the segments are located in different folders, volumes or machines. If all the segments are located in the same folder and are named using the default Omnis data file naming convention, you only need to supply the full path to the first segment.



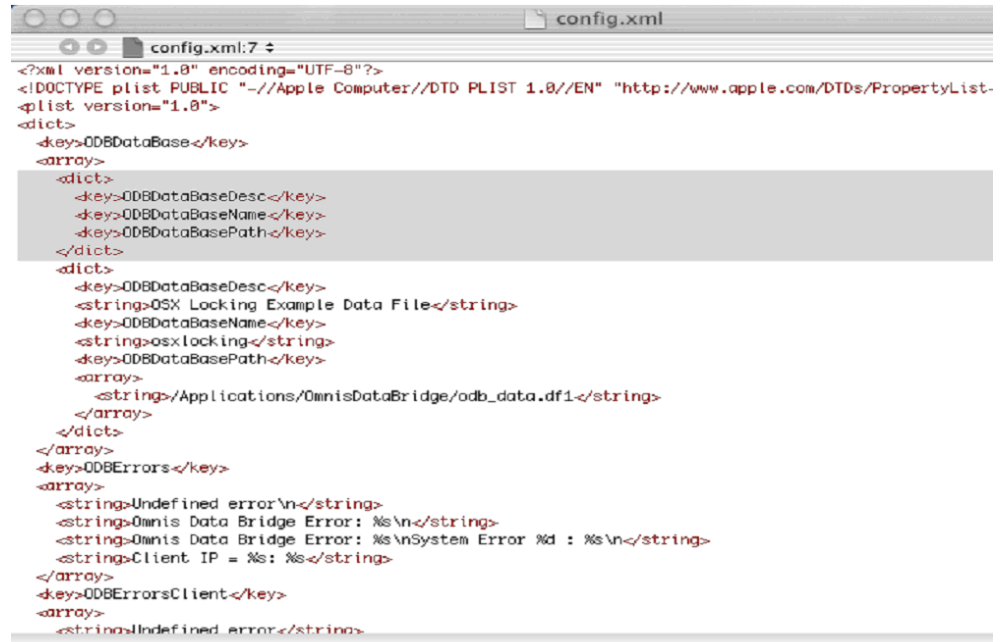
To add a path select *ODBDataBasePath* and click on *New Child*.



Make sure the Class is set to String and enter the full path to your data file segment. The pathname to the Omnis data file can only contain Ascii characters.

## Using a text editor

To add details of your data file using a text editor, create a dictionary array entry for the *ODBDataBase* key. There is no fixed limit to the number of data files you can add. The dictionary entry has to contain the keys *ODBDataBaseDesc*, *ODBDataBaseName* and *ODBDataBasePath*.



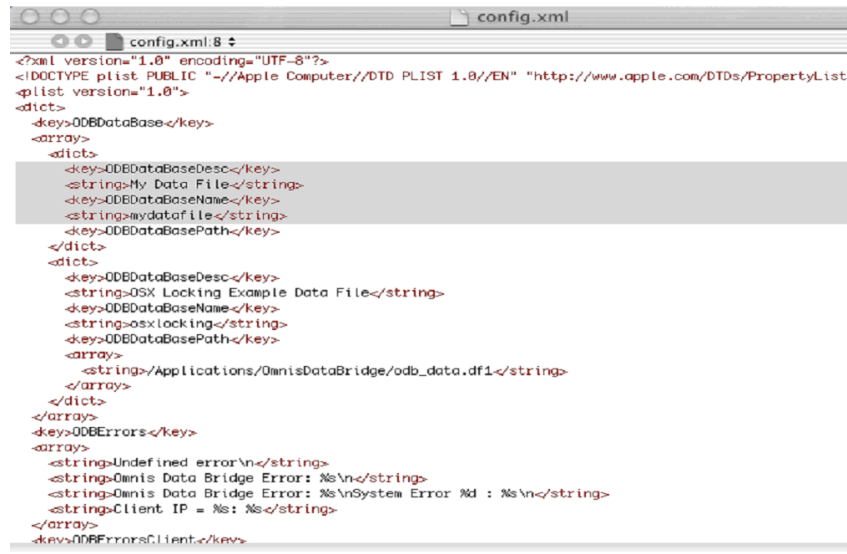
```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>ODBDataBase</key>
  <array>
    <dict>
      <key>ODBDataBaseDesc</key>
      <key>ODBDataBaseName</key>
      <key>ODBDataBasePath</key>
    </dict>
    <dict>
      <key>ODBDataBaseDesc</key>
      <string>OSX Locking Example Data File</string>
      <key>ODBDataBaseName</key>
      <string>osxlocking</string>
      <key>ODBDataBasePath</key>
      <array>
        <string>/Applications/OmnisDataBridge/odb_data.df1</string>
      </array>
    </dict>
  </array>
  <key>ODBErrors</key>
  <array>
    <string>Undefined error\n</string>
    <string>Omnis Data Bridge Error: %s\n</string>
    <string>Omnis Data Bridge Error: %s\nSystem Error %d : %s\n</string>
    <string>Client IP = %s: %s</string>
  </array>
  <key>ODBErrorsClient</key>
  <array>
    <string>Undefined error</string>
  </array>
</dict>
</plist>

```

For the value of *ODBDataBaseDesc* you should enter a sensible description for the data file. This description is displayed to the user if you use *Prompt for data file* to allow the user to select a data file from a list supplied by ODB.

The value of *ODBDataBaseName* must be a unique name that you will use to identify the data file, when opening a data file using the *Open data file* command.

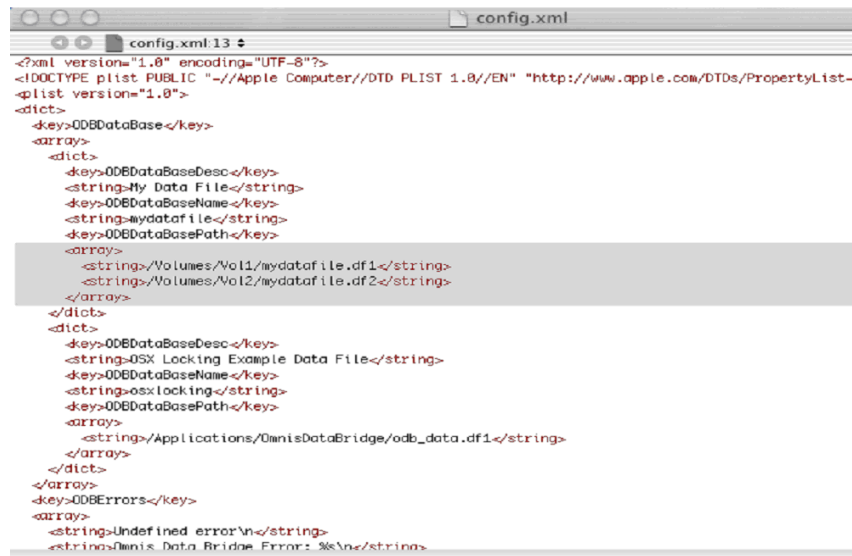


```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>ODBDataBase</key>
  <array>
    <dict>
      <key>ODBDataBaseDesc</key>
      <string>My Data File</string>
      <key>ODBDataBaseName</key>
      <string>mydatafile</string>
      <key>ODBDataBasePath</key>
    </dict>
    <dict>
      <key>ODBDataBaseDesc</key>
      <string>OSX Locking Example Data File</string>
      <key>ODBDataBaseName</key>
      <string>osxlocking</string>
      <key>ODBDataBasePath</key>
      <array>
        <string>/Applications/OmnisDataBridge/odb_data.df1</string>
      </array>
    </dict>
  </array>
  <key>ODBErrors</key>
  <array>
    <string>Undefined error\n</string>
    <string>Omnis Data Bridge Error: %s\n</string>
    <string>Omnis Data Bridge Error: %s\nSystem Error %d : %s\n</string>
    <string>Client IP = %s: %s</string>
  </array>
  <key>ODBErrorsClient</key>

```

The last item *ODBDataBasePath* is an array of full paths pointing to your data file segments on disk. You must enter a full path for each data file segment if the segments are located in different folders, volumes or machines. If all the segments are located in the same folder and are named in the default Omnis naming convention, you only need to supply the full path to the first segment. The pathname to the Omnis data file can only contain Ascii characters.



```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>ODBDataBase</key>
  <array>
    <dict>
      <key>ODBDataBaseDesc</key>
      <string>My Data File</string>
      <key>ODBDataBaseName</key>
      <string>mydatafile</string>
      <key>ODBDataBasePath</key>
      <array>
        <string>/Volumes/Vol1/mydatafile.df1</string>
        <string>/Volumes/Vol2/mydatafile.df2</string>
      </array>
    </dict>
    <dict>
      <key>ODBDataBaseDesc</key>
      <string>OSX Locking Example Data File</string>
      <key>ODBDataBaseName</key>
      <string>osxlocking</string>
      <key>ODBDataBasePath</key>
      <array>
        <string>/Applications/OmnisDataBridge/odb_data.df1</string>
      </array>
    </dict>
  </array>
  <key>ODBErrors</key>
  <array>
    <string>Undefined error\n</string>
    <string>Omnis Data Bridge Error: %s\n</string>

```

## Setting the Port

### ODBSocketPortNumber

In addition to adding details of your data files to the config.xml file, you may also need to change the port on which ODB listens to requests from clients. The port number by default is set to 5913. You need to change it, if that port is already in use.

Simply change the number value of the *ODBSocketPortNumber* key.

If users are accessing your Omnis data file via the ODB from a remote location, your Firewall should allow access to the port you have chosen.

## Other configuration items

### ODBSocketTimeOut

This time out value specifies the number of seconds the ODB will wait for data once it has received the first part of a message from a client. If the remainder of the message is not received within this time ODB will give up and close the connection. By default this is set to 60 seconds, which should be more than enough time to account for network slowdowns.

### ODBKeepStatistics

If set to YES (/true if editing manually), this tells the ODB to maintain statistics of all server access. However this may have a substantial impact on performance, so is /false by default. This option should not be enabled unless it is required to track down some problems. See also *ODBWriteStatistics*.

### ODBWriteStatistics

This number value, if non-zero, tells the ODB to write statistics to the file *info.txt* every *n* seconds. This file can be viewed from any text editor. However this may have a substantial impact on performance, so is zero by default. This option should not be enabled unless it is required to track down some problems. See also *ODBKeepStatistics*.

## Error Configuration

### ODBLogLevel

In the config.xml, the ODBLogLevel number key lets you set the level of error logging reported by the ODB, with the following levels:

0	No Logging except loading of config
1	Startup, shutdown and errors
2	as above plus warnings and client connections and disconnections
3	as above plus all other client activity

# Using the ODB

Once you have updated the configuration file with the details of your data files, you can launch the ODB using the Terminal. This section explains how you launch, shutdown and connect to the Omnis Data Bridge.

## Managing the ODB server

### On Mac OS X

#### Launching ODB

To launch the Terminal program, type:

```
cd /Applications/OmnisDataBridge
```

and press return. Then type

```
./odb start
```

and press return.

If ODB has started you should see

```
Executing start...  
...see messages.txt for success
```

You can check the file *messages.txt* in the ODB folder to check if ODB has started successfully. If an error has occurred this file will contain details of the error.

ODB is now listening for requests from your clients.

#### Shutting down ODB

To shut down ODB from the Terminal, type:

```
cd /Applications/OmnisDataBridge
```

and press return. Then type

```
./odb shutdown
```

and press return.

If ODB has received the shut down request, you should see

```
Executing shutdown...  
...see messages.txt for success
```

If there are users connected, shutdown will fail. It is not safe to shutdown ODB while users are still connected. You should ask the users to disconnect (close their data files) before trying again.

It is however possible to force ODB to shutdown with users still connected.

Type

```
./odb kill
```

and press return.

You may risk data file corruption if a user is writing to the data file while shutting down.

### Removing the ODB

To remove (uninstall) the ODB on Mac OS X do the following:

1. Remove (Trash) the OmnisDataBridge from the Applications Folder
2. Use a Terminal session and enter the following command (as Superuser)

```
pkgutil --forget com.omnissoftware.omnisdatabridge
```

## On Linux

### Launching ODB

There is a helper ODBSet script for the Linux platform which starts the ODB along with any command line parameters. The resulting script is named odbX64.

To run the script and launch the ODB, type:

```
cd /usr/local/omnissoftware/databridgeXXX  
./odbX64      ( or  ./odbX64 start )
```

### Shutting down ODB

To stop the ODB, type:

```
./odbI386 shutdown
```

If there are users connected, shutdown will fail. It is not safe to shutdown ODB while users are still connected. You should ask the users to disconnect (close their data files) before trying again.

It is however possible to force ODB to shutdown with users still connected.

Type

```
./odbI386 kill
```

and press return.

You may risk data file corruption if a user is writing to the data file while shutting down.

## On Windows

### Launching ODB

Once you have updated the configuration file with the details of your data files, you can launch the ODB by running the `odbridge.exe` located in the ODB folder. All commands must be executed via the Administrator user.

Open the command prompt as Administrator and navigate to the ODB folder.

Type

```
odbridge      (or  odbridge start)
```

And press return.

If ODB has started you should see

```
Executing start...  
...see messages.txt for success
```

You can check the file *messages.txt* in the ODB folder to check if ODB has started successfully. If an error has occurred this file will contain details of the error, or

ODB is now listening for requests from your clients

Note: You can also start the ODB by double-clicking `odbridge.exe` in the window view. In this case the data bridge process starts silently, although messages are still written to the *messages.txt* file.

### Shutting down ODB

To stop the data bridge process, open the command prompt as Administrator and type:

```
odbridge shutdown
```

And press return.

If ODB has received the shut down request, you should see

```
Executing shutdown...  
...see messages.txt for success
```

If there are users connected, shutdown will fail. It is not safe to shutdown ODB while users are still connected. You should ask the users to disconnect (that is, close their data files) before trying again.

However, it is possible to force the ODB to shutdown with users still connected.

Open the command prompt as Administrator and type:

```
odbridge kill
```

and press return.

Note: You risk potential data file corruption if a user is writing to the data file while shutting down.

## Installing as a Service

You can install the ODB as a Windows service which starts automatically each time Windows loads. To do this, make sure that the ODB is not running, open the command prompt as Administrator and type:

```
odbridge install
```

and press return.

If the installation is successful, you should see

```
Installing "C:\...\odbridge.exe" as a service
The operation completed successfully.
```

If an error occurs, the error message will be displayed instead. By default, the service (named "Omnis Data Bridge") is configured to start automatically. You can modify this setting via the Windows Control Panel->Administrative Tools->Services panel if required.

## Removing the Service

To stop and remove the "Omnis Data Bridge" service, open the command prompt as Administrator and type:

```
odbridge remove
```

And press return.

If removal is successful, you should see

```
Uninstalling service: odbridge
The operation completed successfully.
```

If an error occurs, the error message will be displayed instead.

Note: Removing the service is equivalent to executing the "odbridge kill" command. You should therefore ensure that there are no clients connected before removing the service.

## Debugging the ODB

Additional debug information on connection attempts, active connections and incoming requests can be obtained by running the ODB in debug mode. To start the ODB in debug mode, specify **debug** instead of **start**. For example:

```
odbridge debug ( or ./odb debug )
```

In this mode, the ODB runs single-threaded and outputs directly to the terminal window. Make sure that there are no active connections to the ODB before terminating the ODB (CTL+C) since this is equivalent to executing `odbridge kill`.



# Opening an ODB data file

You can access an Omnis data file using the DML or via SQL using the Omnis SQL DAM. When using the ODB you need to use the 'IP-address:port,internal-name' string pointing to the ODB rather than using the file path to the Omnis data file. See below for examples.

## Using DML

There are two ways to connect to an Omnis data file from your Omnis library using the Omnis Data Manipulation Language (DML).

You can use the *Open data file* command and enter 'odb://IP-address:port:name,internal name', for example:

```
Open data file {odb://180.70.120.146:5913:mydatafile,mydatafile}
```

or you can use the *Prompt for data file* command and enter 'odb://IP-address:port,internal name', for example:

```
Prompt for data file {odb://180.70.120.146:5913,mydatafile}
```

## Creating new data files

You cannot use the *Create Data File* command in an application running against the ODB. To add a new data file, use *Create data file* (or \$datas.\$add...) on your local file system, move the new empty data file to the ODB server and add an entry to your config.xml file. When the ODB restarts, you can then open the data file, specifying the name as above.

## Using OmnisSQL

You can connect to your Omnis data file via Omnis SQL DAM using code methods, but depending on the version of Studio DAMs you are using (or if you are using Omnis 7.3 v8.2), you will need a different method, as follows.

### Omnis 7.3 v8.2 and Studio 3.3 Old style or 'V2' DAMs

```
Set username {SA}  
Set password {Lion}  
Set hostname {odb://180.70.120.146:5913:odb_test}  
Logon to host (Omnis Character Set)
```

### Object DAMs (Studio 3.3 or higher)

```
Calculate hostname as 'odb://180.70.120.146:5913:odb_test'  
Do SessObj.$logon(hostname,username,password)
```

## Using the SQL Browser

You can connect to your Omnis data file via Omnis SQL DAM using the SQL Browser, but you need to use the 'IP-address:port,internal-name' string pointing to the ODB rather than using the file path to the Omnis data file.

The screenshot shows a window titled "Modify OMNISSQL". Inside, there is a section "Session Definition Details :". The fields are as follows:

- Session Name: OMNISSQL
- DBMS Vendor: OMNIS (dropdown)
- Data Access Module: OMSQLDAM (dropdown)
- DB Version: (empty dropdown)
- Multi-threaded OmnisSQL dam (text label)
- Host Name: odb://180.70.120.146:5913:odb\_test
- User Name: User1
- Password: (masked with asterisks)
- Database: (empty)
- At Startup: ☐ Automatically Logon
- Initialization: (empty text area)
- Maximum rows: 0
- Session type: General (dropdown)
- Transaction mode: Automatic (dropdown)
- Logon character set: OMNIS (dropdown)

At the bottom right are "OK" and "Cancel" buttons.

## ODB Server preference

As an alternative to specifying the IP address and port name in the *Open/Prompt for data file* commands, the *Set hostname* command, or the Hostname field in the SQL Browser, you can store the IP address and port name of the ODB server in the Omnis preference \$odbserver (\$root.\$prefs) and omit the IP-address:port from the command parameters. The preference can contain the TCP/IP address and port of the ODB server in the form IP-address:port, e.g. 194.131.70.222:5900. To use this property, you write code such as:

```
Prompt for data file {odb://}
Open data file {odb://mydatafile}
or
Set hostname {odb://odb_test}
```

The commands use the server identified by \$odbserver. In other words, the address and port components of the odb://datafilename string can be omitted if your ODB server is specified in the \$odbserver property.

The value of \$odbserver is stored in a text file in the Studio folder, called odb.txt. This has one line, in the form:

```
server=address:port
```

This allows you to configure clients by either copying this file, setting up this file in an installer, or using the notation to set \$odbserver via your own dialog.

If the \$odbserver preference is specified, the Data File Browser includes an “Open ODB Datafile” option which will open the data file on the specified ODB server.

## Omnis Datafile List

The Omnis Root method \$getodbfilelist(IList,[cHostIP]) (available in Omnis Studio 5.0 or above) returns a list of Omnis datafiles registered with the Omnis Data Bridge. The cHostIP parameter should be of the form ‘odb://ip-address:port’, or if omitted the value in the \$odbserver property is used.

# ODB Error messages

Note you can set the error logging level in the config.xml in the ODBLogLevel key; see the Configuration section in this manual.

## General Errors

These are errors that occur on the Omnis Data Bridge server that are usually followed by either a server error or client error described below that will give more detail:

### Undefined error

Unknown error has occurred. Anyone receiving this error should contact Omnis technical support.

### Omnis Data Bridge Error: <ODB Error String>

An error has occurred on the Omnis Data Bridge server. Further details will be contained in the error string and may be related to another reported Server error.

### Omnis Data Bridge Error: <ODB Error String> System Error <System Error Code>: <System Error String>

An error has occurred on the Omnis Data Bridge server caused by an operating system function. Further details will be contained in the System error code and error string.

### Client IP = < Client IP Address>: <ODB Error String>

An error has occurred on the Omnis Data Bridge server caused by a function carried out by the client. The IP address will be the address of the Client making the function call. Further details will be contained in the ODB error string.

## Client Errors

Client errors are reported by the server to the client and occur after a datafile statement has been executed on the client.

### **Undefined error**

Unknown error has occurred. Anyone receiving this error should contact Omnis technical support.

### **Specified database does not exist on server**

The datafile specified in the connection string could not be found in the given location. Please check that the specified url is correct on the client and that path to the datafile specified in the server config.xml file is correct.

### **Specified database index is out of range**

The datafile segment requested does not exist. Please check that the specified url is correct on the client and that path to the datafile specified in the server config.xml file is correct.

### **Unable to open specified segment**

The attempt to open or add a new segment could not be completed as the databridge was unable to load the segment. Please check that the specified url is correct on the client and that path to the datafile specified in the server config.xml file is correct.

### **Error during file seek**

This error is caused when attempting to initiate a read of the datafile. This error is caused if the datafile does not exist or if the position that is being searched does not exist either because it is after the end of the file or before the beginning

### **Error during file read**

The error is returned when attempting to read the file. This is caused is the file is invalid, or the file is not open for reading, or the file is locked.

### **Error during file write**

The error is returned when attempting to write to the file. This is caused is the file is invalid, or the file is not open for writing, or the file is locked.

### **Error during resizing file**

This error is caused when attempting to resize the datafile. This error is caused if the datafile does not exist or if the new size is before the beginning of the file.

### **Data received is not valid**

The data sent by the client is invalid. This could be due to the length of data specified not matching the length of the data sent. This could also be caused by missing client headers being sent to the server. Please ensure you have the correct version of the client and server.

### **Byte/byte range already locked**

The data being written to data file has failed as the byte range is locked by another client.

**Specified byte/byte range lock was not found**

An attempt to free a lock on the datafile could not be completed. This could be because the lock does not exist or the datafile is invalid.

**Client is not compatible with server**

The client being used is not compatible with the server. Please ensure that you are using the compatible versions of both the client and server.

**No statistics available**

Therefore are no available server statistics available to be reported.

**Invalid command**

The command sent by the client is not recognised by the server.

**Client unexpectedly disconnected**

The server has disconnected the client. This can happen if the client does not respond to the server for a certain period of time.

**Unable to shut down, clients still connected to server**

An attempt was made to shutdown the server while some clients are still connected. Please ensure that all clients have completed before attempting to shutdown. Alternatively use the kill command to force the server to shutdown.

**Attempt to add segment which exists**

The segment that client is attempting to add already exists.

**Attempt to add segment without previous segment**

Attempt to add an invalid segment. This is caused when the index number of the segment being added is higher than the next available segment.

**Attempt to delete an invalid segment**

Attempt to delete a segment that does not exist.

**Unable to open database in current mode**

An attempt was made to open the datafile in exclusive mode when other users are already connected or an attempt was made to connected to a datafile when it is exclusively locked.

**Unexpected error, bad file number passed to data manager**

Indicates that the datafile may be corrupted and in need of repairing.

## Server Errors

These are errors that occur during server operations and are output to the messages.txt file.

**Undefined error**

Unknown error has occurred. Anyone receiving this error should contact Omnis technical support.

**Failed to create listener socket**

Unable to create listening socket for clients to connect too. This could be because the socket is already connected.

**Failed to bind listener socket**

Unable to bind server IP address to the current socket. This could be because the socket is already connected or the socket was not created properly.

**Failed to prepare listener socket**

Unable to set the socket into listening mode. This could be because the socket is not bound to a local address, the protocol does not support listening on an unbound socket, the socket is already connected or the socket was not created properly.

**Error while waiting for connections**

Error occurred while waiting for a connection from the client. This can be caused by an invalid timeout value or a Network failure.

**Failed to accept client socket**

This error occurs when a client has attempted to connect to the Omnis Data Bridge server. This error is returned if the socket being passed is invalid or if a network error has occurred.

**Invalid header received from client**

This error occurs when a client has sent invalid information in the data header. This is most likely to occur when the client is not an Omnis client.

**Invalid data received from client**

This error occurs when the data sent to the server from the client is a different length from that specified by the client. This can be caused by an alteration or corruption during the sending of the data.

**Error while sending data to client**

This error occurs when an attempt was made to return data back to the client. This error may have occurred if the socket has been disconnected due to a network failure.

**Error while waiting for messages**

The client has made a connection to the server but an error has occurred while waiting for the client to send a request. This error may have occurred if the socket has been disconnected due to a network failure.