

# **Amber Installation Guide**

# Table of Contents

Amber Installation Guide .....	<a href="#">1</a>
Table of Contents .....	<a href="#">2</a>
Introduction .....	<a href="#">3</a>
Server Requirements .....	<a href="#">4</a>
Java Version .....	<a href="#">4</a>
Web Server .....	<a href="#">5</a>
Database Requirements .....	<a href="#">5</a>
Installation Process .....	<a href="#">6</a>
Java Run time .....	<a href="#">6</a>
Win32 .....	<a href="#">6</a>
Sun Solaris .....	<a href="#">6</a>
Linux .....	<a href="#">7</a>
Amber Server Core Installation .....	<a href="#">8</a>
Win32 .....	<a href="#">8</a>
UNIX .....	<a href="#">8</a>
Operating Considerations .....	<a href="#">10</a>
Starting Amber .....	<a href="#">10</a>
Win32 .....	<a href="#">10</a>
UNIX .....	<a href="#">11</a>
Stopping Amber .....	<a href="#">11</a>
Win32 .....	<a href="#">11</a>
UNIX .....	<a href="#">11</a>
Using a Alternative Web Server .....	<a href="#">12</a>
Amber HTTP Interface (Enterprise edition only) .....	<a href="#">12</a>
Using an Alternative Database Server .....	<a href="#">13</a>
Appendix .....	<a href="#">15</a>
Configuration Files .....	<a href="#">16</a>
AmberServer.properties .....	<a href="#">16</a>
DatabaseManager.properties .....	<a href="#">24</a>
Directory Structure .....	<a href="#">27</a>

# Introduction

This document describes how to install and initially configure the Amber server system. The Amber server is designed to operate in a variety of operating systems and interface simply with a variety of web server systems.

The Amber server is a stand-alone server which operates in conjunction with and alongside almost all web servers. This document concentrates on the installation of the Amber Server itself describing the requirements and operating characteristics of the Amber server.

# Server Requirements

For operation Amber requires the following:

1. A Web server. This is used to serve the HTML pages which encase the Amber Java classes.
2. A Java development environment. The development environment is used to develop the Amber Application code.
3. An SQL database which contains information used by the Amber server to identify which Amber application to use for an incoming connection. As Amber must connect to this database the Java JDBC drivers for the database are also required. This is optional for small numbers of applications as a text file can be used to specify connection to application associations.

Included in the installation are a Web Server and a database. The Web Server is the Soma Web Server written by Rob Linwood ([rcl211@is9.nyu.edu](mailto:rcl211@is9.nyu.edu)). The database included in the Amber installation is the HypersonicSQL database engine ([www.hypersonicsql.com](http://www.hypersonicsql.com)). These additional components are included for convenience and to simplify the process of getting started with Amber.

To run and develop Amber applications a Java runtime environment is required for the operating system Amber is installed on. This can be found at a variety of locations however Java for the more common operating systems can be found at the Sun Java web site: [java.sun.com](http://java.sun.com).

## Java Version

The Java runtime must conform with JavaSoft's JDK version 1.1. Should you use a version 1.1 JDK environment the database server must be recompiled to support 1.1. Instructions on how to do this can be found in the amberserver/database directory. The Amber server itself is tested to run under versions of JavaSoft's JDK from 1.1 to 1.4.

## Web Server

Amber will work in conjunction with almost all web servers. The Amber Server uses a database to determine which application to spawn for incoming connections. In its simplest form the Web server is merely a mechanism for transmitting the encapsulating HTML page to the remote browser. For this reason Amber can be integrated with Web Servers which simply serve HTML pages all the way up to major Web/Application server environments.

## Database Requirements

Amber requires a database to define which application to spawn for incoming connections. While there is a database server included for convenience with the current installation Amber is not limited to just this database. Amber is designed to utilise the standard Java JDBC interface to interface to local or remote databases. Therefore any database with JDBC drivers can be used in conjunction with Amber. The configuration of Amber can be simply used to connect to different databases. As the configuration of the JDBC parameters can alter widely it is recommended that the user documentation which came with the JDBC drivers be consulted when configuring Amber to use the new drivers. A list of the various JDBC database drivers can be found using the following URL:

<http://industry.java.sun.com/products/jdbc/drivers>

# Installation Process

This section describes the installation process required to install the Amber server environment onto a specified machine. The process varies depending on the type of operating system that Amber is installed onto. For this reason, the documentation describes a variety of installation types. It is recommended that the one closest to the required operating system is selected and used to install the software.

The installation is defined in 2 steps. These are:

1. Installing the Java run time environment.
2. Installing the Amber server core.

Depending on the requirements and operating system that Amber is installed on several of these steps may be omitted depending on requirements. For example, should the operating system already have the Java run time environment installed this step can be omitted.

## Java Run time

Amber is a Java application. For this reason a Java run time environment is required for the Amber server system to operate. Java run time environments can be found for most operating systems.

The mechanism for installing the Java environments alter depending on the type of operating system. The following sections describe how to install the Java environments included with this installation.

### Win32

The installation images for Java in the Win32 environment are executable programs. The following steps will install the run time environment for Win32 environments. It is important to note that the Java run times are not compatible with Microsoft™ Windows NT 3.51 or earlier operating systems.

1. Download the required Java installation image. Unless there is an absolute requirement for backwards compatibility it is recommended that the JDK 1.3 or later be installed.
2. Run the installation executable. This can be done by double clicking on the file in the file viewer or by explicitly specifying the program in the Start/Run windows menu selection.
3. Follow the directions in the set up dialogs.

### Sun Solaris

The installation images for Solaris are package files which install the run times. The package file selected will depend on the processor of the computer. On the SPARC processors there are two possible choices depending on the version of Java required (1.1.8/1.3). To install these images use the following steps:

1. Download the required Java installation image. Again unless there is an absolute requirement for backwards compatibility it is recommended that the JRE 1.3 or later be installed.
2. Install the Java image using the following command line:

```
pkgadd -d filename
```

where filename is one of the three install images.

## Linux

The installation file for Linux is a Red Hat Package Manager (RPM) installation image for the Intel™ x86 processor range. To install these images use the following steps:

1. Download the required Java installation image.
2. Install the Java run time using the following command line:

```
rpm -i j2sdk-1_XX-linux.rpm where XX is the version of the runtime.
```

# Amber Server Core Installation

The installation sequence for installing the Amber server varies depending on the operating system used. The following sections describe how to install the Amber server on several operating systems. The installation instructions will also install the default database and web server packages.

## Win32

The Win32 installation program can be found in the install directory on the installation CD. To install the Amber Server use the following steps:

1. Find the installation program (**setup.exe**) located in the install/win32 directory on the CD.
2. Run the installation executable. This can be done by double clicking on the file in the file viewer or by explicitly specifying the program in the Start/Run windows menu selection.
3. Click the next button on the Welcome Screen.
4. Specify the destination directory by clicking on the Browse button or using the default setting. Click Next to continue.
5. Select the Program folder name (the name of the program group which appears in the Start/Programs menu) and press Next.
6. Verify your settings in the Start Copying Files Screen and click Next to copy the Amber files to the hard disk.
7. Click the Finish button to end the installation process.
8. Run the GUI Configure program by selecting the GUI Configure item under the Amber Program group. This brings up the configuration editor for the AmberServer.properties and DatabaseManager.properties files. These files configure and control the operating characteristics of the Amber Server. Special attention should be paid to the following properties:
  - AmberServer.AmberRootDirectory. This is the location of the current Amber Server install.
  - AmberServer.DocumentRootDirectory. This is the location of the Web pages used by Amber.
  - AmberServer.LicenseKey. Required for the Amber Server to run. This is filled in using the license key given during the purchase/registration process.

## UNIX

To install the Amber Server on UNIX systems the following is required:

1. Locate the installation image (amber.jar) in the install directory on the installation CD.
2. Copy the installation image to the file system in the required location (typically /opt

although /usr/local is also common) where the amber system will be installed. The amber server will be installed in a directory called amber off this directory.

3. Change directory to the base directory
4. Extract the installation using the following command:

```
jar xvf amber.jar
```

5. This will have created an amber directory containing the amber server files in the target directory.
6. Located in the amber/bin directory are the required script files to start the amber server.
7. Run either the Configure or GUI Configure programs by running either the bin/configure.sh shell script or the bin/guiConfigure.sh from the amber/amberserver directory off the install directory. The GUI configuration program is more complete however requires a working X server to operate. These programs set the AmberServer.properties file and the DatabaseManager.properties file (GUI Configure only) in the config/amberserver and config/database directories. Special attention should be paid to the following properties:
  - AmberServer.AmberRootDirectory. This is the location of the current Amber Server install.
  - AmberServer.DocumentRootDirectory. This is the location of the Web pages used by Amber.
  - AmberServer.LicenseKey. Required for the Amber Server to run. This is filled in using the license key given during the purchase/registration process.

# Operating Considerations

To run the amber server the following components need to be started:

- Database server. This contains configuration information required by the Amber core server to operate. This can be replaced by a text file for simple applications.
- Web server. This server serves the HTML pages containing the Amber components to the client browsers.
- Amber server. This is the core of the Amber system, it responds to the Amber client components and handles the majority of the application processing requirements.

Depending on your requirements two of these components may differ from those installed with Amber. Amber is capable of interacting with a wide variety of Web and database servers. Configuration and operating these other servers is outside the scope of this document.

This document describes the operating characteristics of the three servers installed in the Amber installation package. They are:

- Soma Web Server written by Rob Linwood ([rc1211@is9.nyu.edu](mailto:rc1211@is9.nyu.edu)).
- HypersonicSQL database engine ([www.hypersonicsql.com](http://www.hypersonicsql.com)).
- Amber Server which is the core server of the Amber system.

## Starting Amber

To start the Amber server all three parts of the Amber server must be started. The mechanism for this differs depending on the operating system used.

### Win32

When the Amber server was installed a program group called Amber was created on the Start/Programs menu. Contained in this group are the program items required to start the various parts of the Amber server. These items correspond to batch files located in the amberserver/bin directory relative to the installation directory.

The programs are:

- Start Web Server. This is the program which starts the Soma Web Server. It corresponds to the batch file startWebServer.bat.
- Start Database Server. This program starts the HypersonicSQL database server. It corresponds to the batch file startDatabaseServer.bat.
- Start Amber Server. This program starts the central Amber Server. It corresponds to the batch file startAmber.bat.
- Start All Servers. This program starts all the servers in the correct order. It effectively calls the above three batch files to start the three servers. It corresponds to the batch file startAll.bat.

Amber depends on the Web Server and Database server for its functioning. Typical starting

order of the servers is Database Server, Web Server followed by Amber Server. The order of the last two items can be reversed, however the database server must be operating *before* the Amber Server is started.

## UNIX

Scripts which start each individual server can be found in the amber/bin directory. They are:

- startDatabaseServer. This file starts the HypersonicSQL database engine.
- startWebServer. This file starts the Soma Web Server program.
- startAmber. This file starts the Amber server.
- startAll. This file starts all three servers in the correct order.

Amber depends on the Web Server and Database server for its functioning. Typical starting order of the servers is Database Server, Web Server followed by Amber Server. The order of the last two items can be reversed, however the database server must be operating *before* the Amber Server is started.

## Stopping Amber

Stopping the servers in the Amber system is again dependant on the operating system used. This document will treat each of these in turn.

### Win32

When the programs initiate they create console windows on the Program toolbar. Clicking the Ctrl C key in each window will terminate the corresponding server.

## UNIX

Located in the amberserver/bin directory are a number of scripts which terminate the various components of the server. These scripts correspond to the start scripts used to start the Amber server.

- stopDatabaseServer. This file stops the HypersonicSQL database engine.
- stopWebServer. This file stops the Soma Web Server program.
- stopAmber. This file stops the Amber server.
- stopAll. This file stops all three servers in the correct order.

## Using a Alternative Web Server

Amber will run with a variety of different web servers. Unless the enterprise HTTP interface is required for Amber any simple file serving Web Server can handle the Amber HTTP

requirements.

To use a different web server simply copy the files from the `amberserver/live/htdocs` directory to the required location in the web server html tree. These files can then be accessed by the web browser as required.

In the case where a different web server is required the Soma Web Server should not be run. Either start the Amber Server (and optionally the HypersonicSQL database) using the required `startXXX` scripts in the `amberserver/bin` directory. Once this occurs the HTML pages will correctly connect to the Amber Server and run.

In its purest form a working Amber connection can be created by simply copying the `AmberClient.jar` to the web server HTML root and creating an HTML page to start the Amber client. Examples of HTML starting pages can be found in the Development Guide.

## Amber HTTP Interface (Enterprise edition only)

One requirement of Amber is that the Amber client be able to initiate a socket connection to the Amber Server. In a number of cases this is not possible (for example, in the case of networks containing firewalls etc). While a firewall could be opened this is often too inconvenient to be practicable. In this case an HTTP connection module can be used in the Amber Client and information between the Amber Client and Server is transmitted using the HTTP protocol.

The Web Server requirements are more stringent in this case as the Server side HTTP interface requires a web server capable of running Java servlets.

The following classes require configuration in the Web Server.

Default Client URL	Server Class	Description
<code>http://&lt;server name&gt;/servlet/AmberReceiverConnect</code>	<code>amber.redirect.AmberReceiverConnect</code>	This is the servlet which handles incoming initial connections.
<code>http://&lt;server name&gt;/servlet/AmberReceiverReceive</code>	<code>amber.redirect.AmberReceiverReceive</code>	Receives incoming packets from the Amber Client.
<code>http://&lt;server name&gt;/servlet/AmberReceiverSend</code>	<code>amber.redirect.AmberReceiverSend</code>	Sends packets to the Amber Client from the server.

Other servlets are:

Default Client URL	Server Class	Description
<code>http://&lt;server name&gt;/servlet/AmberReceiverAdminConnect</code>	<code>amber.redirect.AmberReceiverAdminConnect</code>	This is the servlet which handles incoming initial administration connections.
<code>https://&lt;server name&gt;/servlet/AmberReceiverSecureConnect</code>	<code>amber.redirect.AmberReceiverSecureConnect</code>	Handles incoming initial secure connections.
<code>http://&lt;server name&gt;/servlet/AmberReceiverMonitor</code>	<code>amber.redirect.AmberReceiverMonitor</code>	A standard servlet designed to be directly accessed by the browser. It dumps information on the current HTTP connections.

The HTTP interface servlets are configured using the servlet properties mechanism. The properties used by the servlets are:

Property Name	Default Value	Description
<code>AmberServer.ServerAddress</code>	127.0.0.1	This is the IP address of the Amber Server which will handle the connection.
<code>AmberServer.ListenerPort</code>	21384	The port on the Amber Server listening for normal connections.
<code>AmberServer.SecureListenerPort</code>	21385	The port on the Amber Server listening for secure connections.
<code>AmberServer.AdminListenerPort</code>	21386	The port on which the Amber Server listens for administration connections.

## Using an Alternative Database Server

In its simplest form Amber will run with no database at all so long as the connection precache file is used and configured to link an incoming PageID (specified in the HTML page) to an application class.

For more comprehensive systems a database is more commonly used to configure the application characteristics for Amber. The installation includes the HypersonicSQL database however any database can be used. There are several SQL scripts matching several databases

which can be found in the `amberserver/database/scripts` directory. Either use these or create your own based on these examples.

Amber requires at least one database containing a single table **Pages**. This table is similar in form to the connection precache file and configures Amber with information required to determine which application class to use when an incoming Amber Client connection is received.

If the base configuration database is changed then the `config/amberserver/AmberServer.properties` file must be changed to point to the new database. This is done by altering the corresponding database JDBC entries:

- `AmberServer.DatabaseDriver`. The JDBC driver class used to connect to the required database.
- `AmberServer.DatabaseName`. The URL used to connect to the database/table.
- `AmberServer.DatabaseUsername`. The username used when authenticating the connection to the database. Depending on the database driver this parameter may not be used.
- `AmberServer.DatabasePassword`. The password used when authenticating the connection to the database. Depending on the database driver this parameter may not be used.

# Appendix

The following sections describe the characteristics of the installation. They are divided into sections corresponding to various logic components of the Amber Server.

# Configuration Files

The various system components of the Amber server are configured using a variety of property files. These files are used to alter the functioning of the Amber Server. The following sections describe these files.

## AmberServer.properties

This file is located in the `amberserver/config/amberserver` directory and contains the primary configuration options for the Amber Server. Some options may not be available depending on the type of installation. Available options are:

```
#
# Configuration file for the AmberServer Application
#

#
# Name of the AmberServer Object registered with RMI
#
# Default value: AmberServer
#
AmberServer.ServerName=AmberServer

#
# The base directory of the Amber log file.
#
# Default value:
#
AmberServer.LoggingDirectory=

#
# The base filename to log Amber Informational and Error messages to
#
# Default value: AmberServer.log
#
AmberServer.LoggingFilename=AmberServer.log

#
# The level of messages logged by the server.
# Levels are:
# LoggingOff    = -1 ;
# LoggingLow    = 0 ;
# LoggingMedium = 1 ;
# LoggingHigh   = 2 ;
# LoggingDebug  = 3 ;
# LoggingVerbose = 4 ;
```

```
#
# Default value: LoggingLow
#
AmberServer.LoggingLevel=0

#
# The port to listen for non-secure Amber connections.
#
# Default value: 21384
#
AmberServer.ListenerPort=21384

#
# The port to listen for secure Amber connections.
#
# Default value: 21385
#
AmberServer.SecureListenerPort=21385

#
# The port to listen for administration Amber connections.
#
# Default value: 21386
#
AmberServer.AdminListenerPort=21386

#
# The name to use for the AmberServer Connection Pool.
# This connection pool is used to connect to the Amber database which
# is used to determine the correct ApplicationHandler for incoming connections.
#
# Default value: AmberServerDb
#
AmberServer.ConnectionPoolName=AmberServerDb

#
# The database JDBC URL used to connect to the Amber Server database.
#
# Default value: jdbc:odbc:AmberPage
#
# AmberServer.DatabaseName=jdbc:odbc:AmberPage
AmberServer.DatabaseName=jdbc:HypersonicSQL:hsqldb://localhost

#
# The database JDBC driver to use to connect to the Amber Server database.
#
# Default value: sun.jdbc.odbc.JdbcOdbcDriver
```

```
#
# AmberServer.DatabaseDriver=sun.jdbc.odbc.JdbcOdbcDriver
AmberServer.DatabaseDriver=org.hsql.jdbcDriver

#
# The database username to use when logging into the Amber Server database.
#
# Default value:
#
AmberServer.DatabaseUsername=sa

#
# The database password to use when logging into the Amber Server database.
#
# Default value:
#
AmberServer.DatabasePassword=

#
# The number of connections to the Amber Server database which are created when
# the server is first started.
#
# Default value: 1
#
AmberServer.DatabaseMinConnectionNumber=1

#
# The maximum number of connections to the Amber Server database which can possibly be
# created.
#
# Default value: 10
#
AmberServer.DatabaseMaxConnectionNumber=10

#
# The logging level to use when logging the database connection to the Amber Server
# database.
# The values are the same as for AmberServer.LoggingLevel
#
# Default value: LoggingLow
#
AmberServer.DatabaseLoggingLevel=0

#
# The name of the table used in the database to define which application to start for
# normal connections.
#
```

```
# Default value: Pages
#
AmberServer.NormalTablename=Pages

#
# The name of the table used in the database to define which application to start for
# secure (encrypted) connections.
#
# Default value: SecurePages
#
AmberServer.SecureTablename=SecurePages

#
# This file is used either in addition to or in place of the application database
# used by the Amber Server to identify which application to start for an incoming
# normal connection. This file is preloaded into a cache for improved performance in
# responding
# to connections.
#
# Default value: none
#
#AmberServer.NormalPreCacheConnectionFile=

#
# This file is used either in addition to or in place of the application database
# used by the Amber Server to identify which application to start for an incoming
# secure connection. This file is preloaded into a cache for improved performance in
# responding
# to connections.
#
# Default value: none
#
#AmberServer.SecurePreCacheConnectionFile=

#
# This file is used either in addition to or in place of the application database
# used by the Amber Server to identify which application to start for an incoming
# administration connection. This file is preloaded into a cache for improved performance in
# responding
# to connections. This value is currently ignored.
#
# Default value: none
#
#AmberServer.AdminPreCacheConnectionFile=

#
# The connection pool refresh period to use when ensuring the database connections
```

```
# remain live. The value is specified in milliseconds
#
# Default value: 60000
#
AmberServer.ThreadInactivityDuration=60000

#
# The license key for this Amber Server. This is required for the Amber Server to
# start.
#
# Default value:
#
AmberServer.LicenseKey=

#
# The directory under which Amber runs.
#
# Default value:
#
AmberServer.AmberRootDirectory=.

#
# The directory used by the web server as it's document root.
#
# Default value:
#
AmberServer.DocumentRootDirectory=./live/htdocs

#
# The user name used when logging into the Administration System
#
# Default value: admin
#
AmberServer.Admin.Username=admin

#
# The password used when logging into the Administration System
#
# Default value: password
#
AmberServer.Admin.Password=password

#
# The class which is invoked when an administration connection is detected.
# This class must extend ApplicationHandler.
#
# Default value: amber.admin.Admin
```

```
#
AmberServer.Admin.Class=amber.admin.Admin

#
# The secure connection RSA certificate file.
# This is a file generated by the class amber.net.RSAKeyPairGenerator
# This file contains the key used when handshaking with remote secure
# Amber connections.
#
# Default value: config/amberserver/cert.rsa
#
AmberServer.Certificate.Filename=config/amberserver/cert.rsa
9
#
# The password used to decrypt the certificate.
# Typically the Amber server secure connection certificate is encrypted
# to ensure some level of security. This parameter specifies the
# password to use when decrypting the certificate file.
#
# Default value:
#
AmberServer.Certificate.Password=

#
# This flag when set tells the server to log all information not only
# to the log file but also to the console error stream.
#
# Default value: false
#
AmberServer.LogToConsole=false

#
# This setting tells the server how many login threads to use to handle incoming
# connections. By using more threads more incoming connections can be handled
# at once however there is a memory penalty.
#
# Default value: 2
#
AmberServer.LoginThreadNumber=2

#
# How many pending connections the socket listener will allow before rejecting connections.
#
# Default value: 5
#
AmberServer.PendingConnectionNumber=5
```

```
#
# The class which is used to handle and manage Applications.
#
# Default value: amber.server.manager.application.ApplicationManager
#
AmberServer.Manager.ApplicationManager=amber.server.manager.application.ApplicationM
anager

#
# The class which is used to handle and manage Devices.
#
# Default value: amber.server.manager.device.DeviceManager
#
AmberServer.Manager.DeviceManager=amber.server.manager.device.DeviceManager

#
# Additional manager section.
# This section allows the server to be configured to support additional
# types of incoming connections. There are always 3 possible, they are:
# - Browser based connections.
# - Application based connections.
# - Device based connections.
# To add a new manager to the Amber server core use the following format.
# The managers must use monotonically increasing numbers starting at 0.
#
# AmberServer.Manager.0=Full class name of the manager class to add

#
# Extension module section.
# This section allows the server to be configured to support additional
# functionality modules. These modules add some form of functionality to
# the Amber server core.
# To add a new module to the Amber server core use the following format.
# The modules must use monotonically increasing numbers starting at 0.
#
# AmberServer.Extension.0.Name=Name by which the module is identified. Must be unique.
# AmberServer.Extension.0.Module=Full class name of the module class to add
# AmberServer.Extension.0.Configuration=String configuration to be used when starting the
module. This is module specific.
```

## DatabaseManager.properties

This file is located in the amberserver/config/database directory and contains the primary configuration options for the Amber Server Database Manager Interface. Available options are:

```
#
# Configuration file for the Amber Database Manager
#

#
# Global Database Manager properties
#

#
# The base directory of the Amber log file.
#
# Default value:
#
DatabaseManager.LoggingDirectory=

#
# The base filename to log Database Informational and Error messages to
#
# Default value: DatabaseManager.log
#
DatabaseManager.LoggingFilename=DatabaseManager.log

#
# The level of messages logged by the server.
# Levels are:
# LoggingOff    = -1 ;
# LoggingLow    = 0 ;
# LoggingMedium = 1 ;
# LoggingHigh   = 2 ;
# LoggingDebug  = 3 ;
# LoggingVerbose = 4 ;
#
# Default value: LoggingLow
#
DatabaseManager.LoggingLevel=0

#
# The basic structure of the file is as a series of Connection Pool entities
# The entities are numbered from 0 up sequentially.
# Parameters for the Connection pools are:
#
```

```
#
# The name by which the connection pool will be accessed. This value MUST be
# unique or the connection pools cannot be accessed.
#
# Default Value:
# DatabaseManager.ConnectionPool0.PoolName

#
# The database JDBC URL used to connect to the database.
#
# Default value:
#
# DatabaseManager.ConnectionPool0.DatabaseName

#
# The database JDBC driver to use to connect to the database.
#
# Default value: sun.jdbc.odbc.JdbcOdbcDriver
# DatabaseManager.ConnectionPool0.DatabaseDriver

#
# The database username to use when logging into the database.
#
# Default Value:
# DatabaseManager.ConnectionPool0.DatabaseUsername

#
# The database password to use when logging into the database.
#
# Default Value:
# DatabaseManager.ConnectionPool0.DatabasePassword

#
# The minimum number of connections to create when the connection pool is created.
#
# Default Value: 5
# DatabaseManager.ConnectionPool0.MinConnectionNumber

#
# The maximum number of connections to ever create for the connection pool.
#
# Default Value: 10
# DatabaseManager.ConnectionPool0.MaxConnectionNumber

#
# The connection pool refresh period to use when ensuring the database connections
# remain live. The value is specified in milliseconds
```

#  
# Default Value: 60000  
# DatabaseManager.ConnectionPool0.ThreadInactivityDuration

# Directory Structure

The various components of the Amber server are located in specific subdirectories located off the main installation directory (typically called amber). The directory structure of the Amber installation is as follows, again this may differ depending on the installation:

```
amber
  amberserver
    bin
    classes
    config
      amberserver
      database
      elektron
    component
  database
  src
  lib
  com
  live
  htdocs
  amber
    admin
      images
    doc
    elektron
      images
    images
    lib
  usr
    classes
    src
  webServer
```

The contents of these directories are:

- bin. The binary files which when run start/stop the Amber components.
- classes. Classes which relate to Amber. These typically relate to the client component classes. While there are a variety of client jars available for use in the lib directory. These classes are included separately to allow the creation of custom jars containing only required Amber client files.
- config. This directory contains configuration files required by the various components of the Amber Server.
- database. This contains the files relating to the HypersonicSQL database.
- lib. Various Java library files required by the various Amber components. This is the location of the .jar files. It also contains the class files for the Soma Web server.

- live. The base directory for the Soma web server. The HTML root directory for the web server is the htdocs directory. This directory contains the web pages used by Amber.
  - amber/admin. Amber administration HTML files.
  - amber/doc. Amber related document HTML files.
  - amber/elektron. Elektron development environment HTML files.
  - amber/images. Images required by general Amber HTML files.
  - amber/lib. The location of the Amber client jar files.
- run. Contains the process ID files for the various servers when run under UNIX systems.
- usr. Contains the user files created by the developer. This is the base directory for the Elektron program.
  - classes. This is the default location for the user created class files.
  - src. This is the default location for the user created source files.
- webServer. A copy of the Soma web server.