

## Beginners Guide to uPortal

Lee Chamberlain, Margaret Nah Juahkah, and Robert Sherratt

Version 1.0  
4 July 2003

PORTAL Workpackage **10**

the PORTAL project is funded under  
the FAIR Programme of the Joint Information Systems Committee

## Acknowledgements

The PORTAL project is funded under the FAIR Programme of the Joint Information Systems Committee (JISC).

The project is undertaken as a partnership between Academic Services Interactive Media at the University of Hull and UKOLN, based at the University of Bath.

For further information on the JISC and JISC Programmes, visit their Web site at <http://www.jisc.ac.uk/>.

For further information on Academic Services Interactive Media, visit their Web site at <http://www.acsweb.hull.ac.uk/imedia/>.

For further information on UKOLN, visit their Web site at <http://www.ukoln.ac.uk/>.

For further information on the PORTAL project itself, visit the project Web site at <http://www.fair-portal.hull.ac.uk/>.

## Contributors

Much of the material contained in this guide has been taken from documentation on the JA-SIG web site, some of the main contributors of this material being:

<b>John Ferreira</b>	Cornell University, JA-SIG documentation writer
<b>Zed A. Shaw</b>	University of British Columbia

The following people kindly offered their installation notes which have been incorporated into this guide:

<b>Elliot Smith</b>	Birmingham University
<b>Paul Browning</b>	University of Bristol
<b>Mat Hunter</b>	Virginia Tech.
<b>Geralyn M. Hollerman</b>	Louisiana University

## Table of Contents

<b>1. INTRODUCTION</b> .....	<b>4</b>
1.1    INITIAL REQUIREMENTS .....	5
<b>2. QUICK START DISTRIBUTION</b> .....	<b>6</b>
2.1    STARTING QUICK START ON WINDOWS.....	7
2.2    STARTING QUICK START ON UNIX.....	8
<b>3. uPORTAL ONLY DISTRIBUTION</b> .....	<b>11</b>
3.1    CONFIGURING uPORTAL.....	12
<i>The Database Server</i> .....	13
<i>Other Database examples</i> .....	15
<i>Database Connection Pooling</i> .....	15
<i>The Servlet Container</i> .....	15
3.2    INSTALLING ON WINDOWS.....	16
3.3    INSTALLING ON UNIX .....	20
<b>4. EXPLORING uPORTAL</b> .....	<b>26</b>
4.1    DOWNLOADING THE uPORTAL SOURCE CODE .....	29
<b>APPENDIX A — SETTING JAVA_HOME</b> .....	<b>30</b>
<b>APPENDIX B — CHECKING PORTS</b> .....	<b>31</b>
<b>APPENDIX C — ANT</b> .....	<b>32</b>
<b>APPENDIX D — WEB APPLICATION ENVIRONMENTS</b> .....	<b>34</b>

# 1. Introduction

This guide is aimed primarily at those new to installing and configuring the uPortal product. It also serves as an introduction to the main features of uPortal. Readers are assumed to have a basic understanding of the operating system upon which uPortal is to be installed.

The current release of this guide has been tested against version 2.1.2 of uPortal, although it is likely that other 2.1.x versions will not differ greatly from this.

A second version of this guide will be released towards the end of the PORTAL project, early in 2004. Version 2 will address updates to the uPortal software, as well as enhancements to the documentation highlighted through its use.

uPortal is available for download in two different forms; a 'Quick Start' package and a 'uPortal only' package. Both are covered in this guide.

The Quick Start package enables easy installation of the portal and includes all of the software required to investigate and evaluate uPortal. The uPortal only package requires more detailed technical knowledge for deployment, integrating the portal software with local systems, and would be used when looking to implement a production version of uPortal.

For the uPortal only distribution, or where a degree of customisation is required, it is useful to have a working knowledge of:

- Jakarta Ant – a Java based build tool
- A servlet container such as *Apache Tomcat*
- Handling SQL database queries via JDBC

For those users who lack this knowledge, Appendices C and D at the end of this guide provide assistance in gaining access to resources and tutorials in these areas.

uPortal users will find extensive documentation, which is frequently being updated, on the JA-SIG uPortal site at:

<http://mis105.mis.udel.edu/ja-sig/uportal/docs.html>

You will find information covering a variety of topics under the following headings:

- **uPortal Roles**
- **Getting Started**
- **Implementation Documentation**
- **Developer**

- **Background/Architecture**
- **Release information**

In addition, the archives of the JA-SIG PORTAL mailing list may be searched at:

<http://list.unm.edu/cgi-bin/wa?S1=jasig-portal/>

The archives contain a wide variety of discussion threads concerning problems and solutions raised regarding installation and configuration issues. If you continue to experience problems with an installation of uPortal it is worth seeking assistance by outlining the problem and posting the matter to this group. It is generally useful to include the details such as:

- The platform and operating system
- The uPortal version number
- The Ant version number
- The web application environment and version number
- The database and version number
- Changes made to properties files where appropriate
- Error logs or listings

## 1.1 Initial Requirements

Before proceeding with an installation of uPortal some initial preparation of your system must be undertaken. uPortal requires the Java Development Kit (JDK) 1.1.8 or higher; uPortal has been tested on the Sun Virtual Machine, but not on the Microsoft Virtual Machine. You can obtain the JDK by going to the download section at:

<http://java.sun.com/>. A second requirement is that the `JAVA_HOME` variable is set to point to the place where the JDK is installed. Details on how to do this can be found in Appendix A.

## 2. Quick Start Distribution

The Quick Start distribution provides an ideal test bed for users who wish to make an evaluation of uPortal and its features. You can obtain the Quick Start distribution from the JA-SIG web site at:

<http://mis105.mis.udel.edu/ja-sig/uportal/download.html>

It downloads as a zip file. Installation simply requires you to extract the downloaded file. This creates a single directory, e.g. `uPortal_2_1_2-quick-start`. In the documentation this directory will be referred to as `<UPortal_HOME>`. If you look inside the directory you will discover that the Quick Start distribution comes complete with the following preconfigured components which appear as directories:

- **uPortal 2.x.x** - containing source code, libraries and property files: the contents of the uPortal only distribution
- **Ant 1-x** – the Jakarta build tool
- **Tomcat 4.x.x** – the Apache servlet container
- **HypersonicSQL 1.x.x** – the (HSQL) database which is preloaded with the data and tables required by uPortal

You will also find several files in the directory:

- **build.xml** – the build file used by Ant
- **fix-unix-modes.sh** - a unix shell script for setting file permissions
- **ant.sh, ant, ant.bat** - scripts and a Windows batch file used to invoke the Ant built tool

You can now start the Quick Start distribution. Two components need to be started before uPortal will work; the database HypersonicSQL and the servlet container Tomcat. These are initiated by running the build file - `build.xml` - with the Ant build tool.

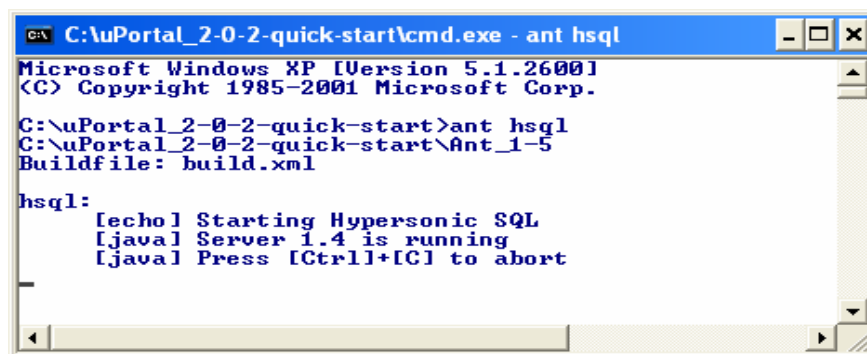
The `build.xml` file allows you to start the HSQL database, and stop or start the Tomcat servlet container. Build files can prove very useful and for those readers new to Ant it is worthwhile reading Appendix C, which contains a section on how to gain further information about the Ant build tool.

## 2.1 Starting Quick Start on Windows

Open a command window and go to <UPortal\_HOME>, e.g. c:\uPortal\_2-1-2-quick-start. In order to access uPortal you will need to start the HypersonicSQL (HSQL) database server and then start the Tomcat servlet container. You do this by using Ant to execute the build file.

To start HSQL type: `ant hsql`

If you have started the database server correctly you will see a response similar to the one shown in the window below:



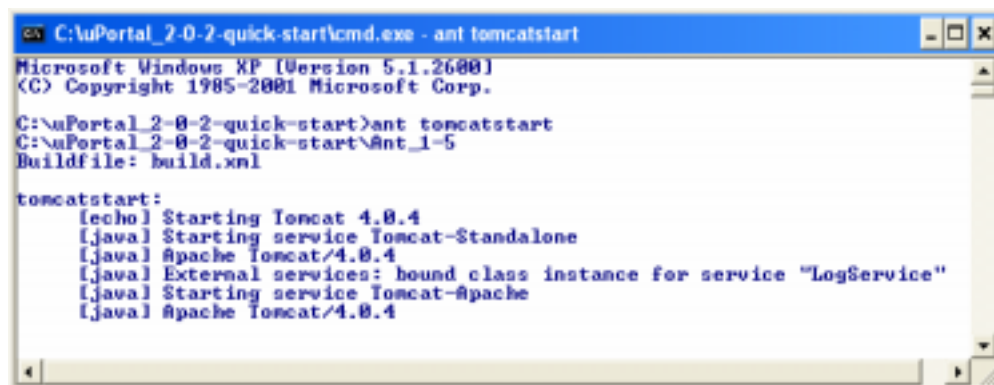
```

C:\uPortal_2-0-2-quick-start\cmd.exe - ant hsql
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\uPortal_2-0-2-quick-start>ant hsql
C:\uPortal_2-0-2-quick-start\Ant_1-5
Buildfile: build.xml

hsql:
[echo] Starting Hypersonic SQL
[java] Server 1.4 is running
[java] Press [Ctrl]+[C] to abort
  
```

Open another command window and go to <UPortal\_HOME>. To start Tomcat type: `ant tomcatstart`. The following window provides an example of the response you will get if you have managed to start Tomcat.



```

C:\uPortal_2-0-2-quick-start\cmd.exe - ant tomcatstart
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\uPortal_2-0-2-quick-start>ant tomcatstart
C:\uPortal_2-0-2-quick-start\Ant_1-5
Buildfile: build.xml

tomcatstart:
[echo] Starting Tomcat 4.0.4
[java] Starting service Tomcat-Standalone
[java] Apache Tomcat/4.0.4
[java] External services: bound class instance for service "LogService"
[java] Starting service Tomcat-Apache
[java] Apache Tomcat/4.0.4
  
```

If you experience difficulties starting HSQL or Tomcat this may be due to a conflict with other software running on your PC. See Appendix B for details of how to check what software is running on which ports on your PC. In addition, it is advisable to open a browser pointed to the host and port on which you have started Tomcat, for example:

`http://localhost:8080/`, you should see a welcome screen similar to the one below.



You are now ready to start exploring uPortal – this is discussed in section 4 of this guide.

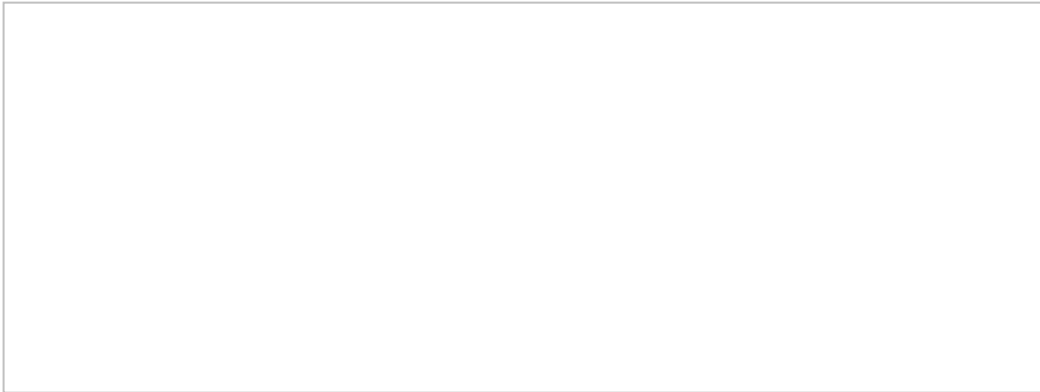
When you have finished your session with Quick Start uPortal you should shut down Tomcat by opening a command window, going to `<Uportal_HOME>` and entering the command: `ant tomcatstop`. This will shut Tomcat down in an elegant manner and retain any changes you make to the HSQL database. Close down the HSQL database by pressing `Ctrl-C` in the command window that you used to start HSQL.

## 2.2 Starting Quick Start on Unix

Open a command window and go to `<Uportal_HOME>`, e.g. `/home/uPortal_2-1-2-quick-start`. In order to access uPortal you will need to start the HypersonicSQL (HSQL) database server and then start the Tomcat servlet container. You do this by using Ant to execute the build file. To start HSQL type: `./ant .sh hsql`. If you have managed to start the database correctly you will see a response similar to the one below:

```
/uPortalquick>./ant.sh hsql
Buildfile: build.xml
hsql:
    [echo] Starting Hypersonic SQL
    [java] Server 1.4 is running
    [java] Press [Ctrl]+[C] to abort
```

Open another terminal window and go to <UPortal\_HOME>. To start Tomcat type:  
`ant tomcatstart`. The listing below shows an example of the response you will get if you have started Tomcat correctly.



If you experience difficulties starting tomcat this may be due to a conflict with other software running on your machine. See Appendix B for details of how to check what software is running on which ports. In addition, it is advisable to open a browser pointed to the host and port on which you have started Tomcat, for example: `http://localhost:8080/`, you should see a welcome screen similar to the one below.



You are now ready to start exploring uPortal – this is discussed in section 4 of this guide.

When you have finished your session with Quick Start uPortal you should shut Tomcat down by opening a command window, going to <UPortal\_HOME> and entering the

command: `ant tomcatstop`. This will shut down Tomcat in an elegant manner and retain any changes you make to the HSQL database. Close down the HSQL database by pressing `Ctrl-C` in the command window that you used to start HSQL.

### 3. uPortal Only Distribution

The uPortal only distribution is suitable for use by those intending to run and implement a production version of uPortal, or who plan to run other web applications on the same server.

In addition to the uPortal only distribution you will also need:

- **Ant** the build tool, version 1.4 or later, which may be downloaded from:  
<http://jakarta.apache.org/ant/>
- **A Java Servlet Engine/Container.** This must be compatible with the Servlet 2.2 and JSP 1.1 API. uPortal is developed under Tomcat making it a good platform to adopt - the build files for uPortal have been set up such that it can be easily deployed into Tomcat. For further information see:  
<http://jakarta.apache.org/tomcat/>  
Resin has also been used successfully by many institutions, and more information may be found at: <http://www.caucho.com/resin/>
- **A JDBC compliant database.** uPortal is developed using the HypersonicSQL database. It is an open source application and may be downloaded from:  
<http://sourceforge.net/projects/hsqldb/>  
The following databases should work with some modification to the CREATE scripts and a number of SQL statements: *HypersonicSQL, MySQL, Oracle, MS SQL Server, PostgreSQL, DB2, Sybase SQL Server* and *Ingres*.  
Detailed advice on database configuration and compatibility may be found on the JA-SIG site on: <http://mis105.mis.udel.edu/ja-sig/uportal/administrators/database.html>

The uPortal only distribution downloads as a zip file that is extracted into a single directory, e.g. `uPortal_rel-2-1-2`. In the documentation this directory will be referred to as `<UPortal_HOME>`. If you look inside the directory you will discover it contains several directories and files:

- **docs** – a directory containing documentation about uPortal
- **source** – a directory hierarchy containing the source code for uPortal
- **lib** – a directory containing jar files for the Java API's used by uPortal
- **webpages** – a directory containing a hierarchy stylesheets, image files, java servlet pages, and other files for uPortal
- **properties** – a directory which contains properties and configuration files
- **build.xml** – a build file for use by Ant

- **build.properties** – a properties file for use by Ant

The installation of the uPortal only distribution is initiated by running the build file - `build.xml` - with the Ant build tool. The build file contains a set of “targets” which enable operations to be performed when an Ant command is given in the following forms,

**Windows:** `ant <target_name>`

**Unix:** `./ant.sh <target_name>`

The `build.xml` file is different from the one used with the Quick Start distribution.

The build file provides several targets:

- Clean** - delete old build and dist directories
- Deploy** - deploys uPortal into a servlet container
- Db** - prepares a relational database with uPortal schema and data
- Dist** - creates JavaDoc, uPortal jar and WAR files

Build files are widely used with uPortal and for those readers new to Ant it is worthwhile reading Appendix C, which contains a section on how to gain further information about the Ant build tool.

The next sections detail the sequence of actions necessary to install the uPortal only distribution:

- Configure the database
- Configure the servlet container
- Run the database server on the designated port
- Populate the database
- Deploy the uPortal files into the servlet container

### 3.1 Configuring uPortal

You will need to configure uPortal to work in conjunction with the database server and the web environment of your choice before installation. Before you attempt to configure uPortal it may be useful to familiarise yourself with a number of documents which detail several aspects of uPortal configuration. Two useful reference documents may be found in `<UPORTAL_HOME>/docs`:

- **uPortal\_properties\_overview.txt** - gives a brief description of the contents of several xml, properties and conf files
- **uPortal\_tools\_overview.txt** - uPortal has a handy set of tools to assist developers with routine tasks. This document explains how these tools may be

used to add users, register channels types and stylesheets; and also explains the workings of DbLoader/Unloader

The following two sub-sections explain how to configure uPortal to run in conjunction with a database server and a servlet container respectively.

## The Database Server

By way of illustration this section explains how to configure uPortal for use with a database. For illustration examples are given using the HypersonicSQL v1.6.1 server, an open source Java database engine with a standard SQL and JDBC interface. It is good practice to make a copy of any files that need to be edited before you begin to modify them.

### Step 1.

Copy your JDBC driver jar file to `<UPortal_HOME>/lib`.

Go to the `<HYPERSONICSQL_HOME>/lib` and copy the jar file, `hsqldb.jar`. Copy the jar file into the `<UPortal_HOME>/lib` directory – where you will notice there is already a file called `hsqldb.jar` which you may rename if it causes confusion.

### Step 2.

Edit the `jdbcdriver.jar` property in `<UPortal_HOME>/build.properties` to point to the jar file copied in Step 1.

Go to the `<UPortal_HOME>` directory and open the file `build.properties`. Search for the line of text: `jdbcdriver.jar=./lib/hsqldb.jar`, and replace it with the following: `jdbcdriver.jar=./lib/hsqldb.jar` to reflect the change in name of the new database server jar file. Save the file.

### Step 3.

Edit the `<UPortal_HOME>/properties/rdbm.properties` file adding the correct details for your database driver, URL, user name and password.

Go to the `<UPortal_HOME>/properties` directory and open the `rdbm.properties` file.

Look under the heading `#### Hypersonic SQL - Server mode` for the following two lines:

```
jdbcdriver=org.hsqldb.jdbcDriver
jdbcUrl=jdbc:HypersonicSQL:hsqldb://localhost:8887
```

change them to: `jdbcdriver=org.hsqldb.jdbcDriver`

```
jdbcUrl=jdbc:hsqldb:hsqldb://localhost:8887
```

It assumed that port 8887 is free and that this is the number of the port on which you will run the database server. You can find out whether another process has already laid claim to this port by opening a command or terminal window and typing `netstat -a`.

#### Step 4.

Modify `<UPORTAL_HOME>/properties/dbloader.xml` to include the correct details for your database and driver including any local data type mappings.

In the `<UPORTAL_HOME>/properties` directory you will find a file named `dbloader.xml`, open it and search for the following block of text:

```
<db-type-mapping>
  <db-name>HypersonicSQL</db-name>
  <db-version>1.4</db-version>
  <driver-name>HypersonicSQL Driver</driver-name>
  <driver-version>1.4</driver-version>
  <type><generic>INTEGER</generic><local>INTEGER</local></type>
  <type><generic>VARCHAR</generic><local>VARCHAR</local></type>
  <type><generic>LONGVARCHAR</generic><local>LONGVARCHAR</local></type>
  <!-- map more types here -->
</db-type-mapping>
```

Change the two lines which refer to the db (database) version and driver version so that they reflect the current version, e.g. 1.6.1.:

```
<db-version>1.6.1</db-version>
<driver-version>1.6.1</driver-version>
```

Strictly speaking this will not affect the installation, but failure to note the version change may lead to confusion should you lose track of which versions you are currently using.

#### Step 5.

Modify `<UPORTAL_HOME>/properties/PersonDirs.xml` file to point to the database in which you store user information.

You may need to modify the `PersonDirs.xml` file; this file is used if the database is intended to provide user directory information. Search for the following block of text and modify as required.

```

<!-- JDBC Properties -->
  <driver>org.hsql.jdbcDriver</driver>
  <url>jdbc:HypersonicSQL:hsql://localhost:8887</url>
  <logonid>sa</logonid>
  <logonpassword></logonpassword>
  <uidquery>SELECT FIRST_NAME || ' ' || LAST_NAME AS FIRST_LAST,
            FIRST_NAME, LAST_NAME, EMAIL FROM UP_PERSON_DIR
            WHERE USER_NAME=?</uidquery>

```

For *HypersonicSQL* locate the following two lines:

```

<driver>org.hsql.jdbcDriver</driver>
  <url>jdbc:HypersonicSQL:hsql://localhost:8887</url>

```

and replace the text with these lines:

```

<driver>org.hsqldb.jdbcDriver</driver>
  <url>jdbc:hsqldb:hsql://localhost:8887</url>

```

This completes the steps required to configure uPortal for use with a database.

## Other Database examples

Details about which databases are supported by uPortal and information on how to connect uPortal with your database platform can be found at:

<http://mis105.mis.udel.edu/ja-sig/uportal/administrators/database.html>

## Database Connection Pooling

uPortal also supports database connection pooling using *Poolman*. The JA-SIG uPortal web site has a Poolman page which can be found via

<http://mis105.mis.udel.edu/ja-sig/uportal/administrators/database.html>

It is currently being developed, but will contain information on several topics:

- Setting up Poolman
- Obtaining the Driver
- Properties Configuration
- Loading the Database
- Issues and Known Bugs

## The Servlet Container

If you have not already installed a web environment that includes a servlet container you will need to do so. This section explains how to configure the Apache Tomcat servlet container to provide a suitable environment for uPortal. You will now have to change the location of the place into which the uPortal files are deployed. Go to `<UPORTAL_HOME>`, open the `build.properties` file and search for the line containing the value for 'deploy home':

```
deploy.home=C:/JavaClasses/jakarta-tomcat-4.0.4/webapps/uPortal2
```

Change that line so that the value points to the Tomcat directory on your system, for example,

### Windows

```
deploy.home=C:/Tomcat_4-0-4/Tomcat_4-04/webapps/uPortal2
```

### Unix

```
deploy.home=/local/Tomcat_4-0-4/Tomcat_4-0-4/webapps/uPortal2
```

Sections 3.2 and 3.3 continue with the installation of uPortal on the Windows and Unix platforms respectively.

## 3.2 Installing on Windows

This section demonstrates how uPortal may be installed on a Windows platform; it continues the example using HypersonicSQL and Tomcat as the respective database server and servlet container and assumes that you have already installed and configured a database server and a servlet container. The screen shot below shows a typical directory structure containing four directories which hold all the systems required to run the uPortal only distribution.

```

C:\uPortal_rel-2-0-2>dir
Volume in drive C is UAIQ
Volume Serial Number is 341B-3E06

Directory of C:\uPortal_rel-2-0-2

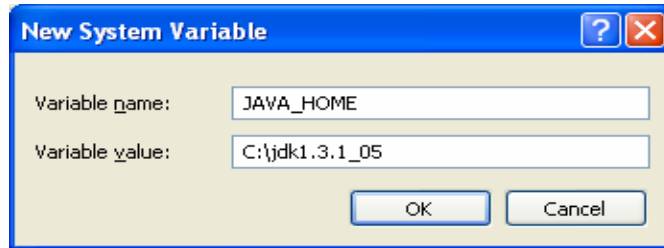
11/11/2002  16:23    <DIR>          .
11/11/2002  16:23    <DIR>          ..
11/11/2002  16:31    <DIR>          ant
11/11/2002  09:53    <DIR>          hsqldb_v.1.61
11/11/2002  09:38    <DIR>          Tomcat_4-0-4
11/11/2002  10:16    <DIR>          uPortal_rel-2-0-2
               0 File(s)      0 bytes
               6 Dir(s)  5,862,465,536 bytes free

```

The build tool Ant should be installed and the following environment variables set, for example if Ant is installed in `c:\ant`:

- set `ANT_HOME=c:\ant`
- set `JAVA_HOME=c:\jdk1.2.2`
- add `ANT_HOME%\bin` to the `PATH` variable

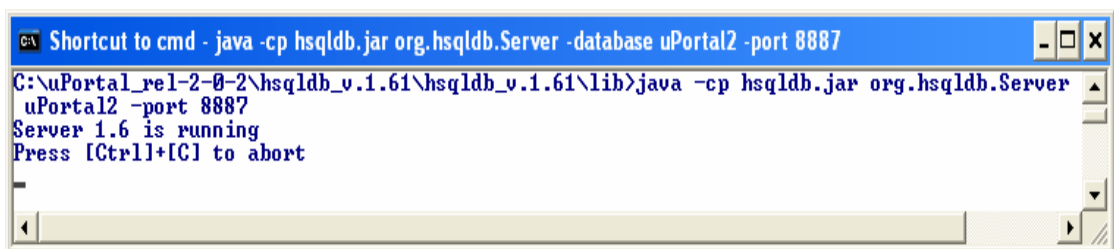
This may be done by using the *System Properties* tool from the Control Panel. Elect to add a new *Environment Variable* and enter the complete path of the directory holding the JDK, an example is shown below.



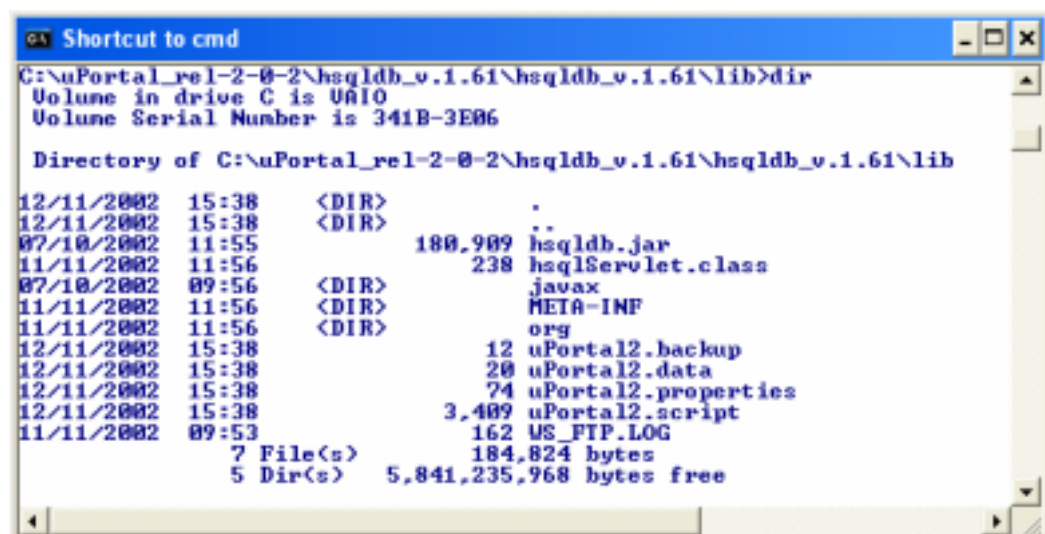
Open a command window and go to the <HYPERSONICSQL\_HOME>/lib directory. Type the following command that should run the java file within the `hsqldb.jar` file and initiate the start of the server on the desired port:

```
java -cp hsqldb.jar org.hsqldb.Server -database uPortal2 -port 8887
```

An example of the correct response to this command is shown in the screen shot below.



You will notice that the directory now contains script and database files for uPortal2, for example:



The next step is to create the tables and populate the database. Open another command window and go to the <UPORTAL\_HOME> directory, which contains the build file `build.xml`. Enter the Ant command to build the database: `ant db`. It may take several seconds for this action to be completed, the following window shows an example of a successful build.

```

C:\uPortal_rel-2-0-2>ant db
Buildfile: build.xml

prepare:
[mkdir] Created dir: C:\uPortal_rel-2-0-2\Portal_rel-2-0-2\build
[copy] Copying 554 files to C:\uPortal_rel-2-0-2\Portal_rel-2-0-2\build

compile:
[mkdir] Created dir: C:\uPortal_rel-2-0-2\Portal_rel-2-0-2\build\WEB-INF\classes
[javac] Compiling 268 source files to C:\uPortal_rel-2-0-2\Portal_rel-2-0-2\build\WEB-INF\classes
[javac] Note: Some input files use an obsolete deprecated API.
[javac] Note: Recompile with -deprecation for details.
[copy] Copying 17 files to C:\uPortal_rel-2-0-2\Portal_rel-2-0-2\build\WEB-INF\classes\properties
[copy] Copying 2 files to C:\uPortal_rel-2-0-2\Portal_rel-2-0-2\build\WEB-INF\classes\org\jasig\portal\serialize
[copy] Copying 81 files to C:\uPortal_rel-2-0-2\Portal_rel-2-0-2\build\WEB-INF\classes
[copy] Copying 75 files to C:\uPortal_rel-2-0-2\Portal_rel-2-0-2\build\WEB-INF\classes

db:
[echo] Loading DBLoader
[java] Starting DBLoader...
[java] Database name: 'HSQL Database Engine'
[java] Database version: '1.6'
[java] Driver name: 'HSQL Database Engine Driver'
[java] Driver version: '1.6'
[java] Driver class: 'org.hsqldb.jdbcDriver'
[java] Connection URL: 'jdbc:hsqldb:hsq://localhost:8889?'
[java] User: 'sa'
[java] Parsing file:C:\uPortal_rel-2-0-2\Portal_rel-2-0-2\build\WEB-INF\classes\properties\dbloader.xml...
[java] Dropping tables.....
[java] Creating tables.....
[java] Populating tables.....
.....
[java] Done!
[java] Elapsed time: 5.067 seconds

BUILD SUCCESSFUL

```

You are now ready to deploy the uPortal files into the servlet container. Enter the Ant command: `ant deploy`. This build may take several minutes to complete depending on the speed of the machine onto which you are installing uPortal. A successful build will result in a response similar to the one shown in the screen shot below.

```

C:\uPortal_rel-2-0-2>ant deploy
Buildfile: build.xml

prepare:
[mkdir] Created dir: C:\uPortal_rel-2-0-2\Portal_rel-2-0-2\build
[copy] Copying 462 files to C:\uPortal_rel-2-0-2\Portal_rel-2-0-2\build

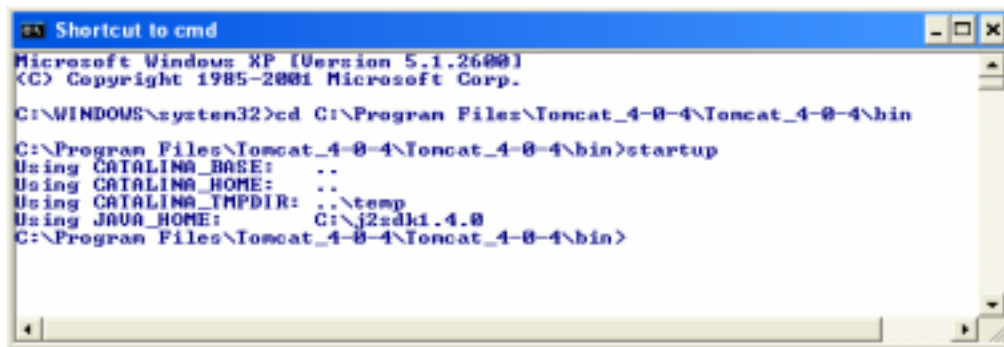
compile:
[mkdir] Created dir: C:\uPortal_rel-2-0-2\Portal_rel-2-0-2\build\WEB-INF\classes
[javac] Compiling 268 source files to C:\uPortal_rel-2-0-2\Portal_rel-2-0-2\build\WEB-INF\classes
[javac] Note: Some input files use an obsolete deprecated API.
[javac] Note: Recompile with -deprecation for details.
[copy] Copying 16 files to C:\uPortal_rel-2-0-2\Portal_rel-2-0-2\build\WEB-INF\classes\properties
[copy] Copying 2 files to C:\uPortal_rel-2-0-2\Portal_rel-2-0-2\build\WEB-INF\classes\org\jasig\portal\serialize
[copy] Copying 81 files to C:\uPortal_rel-2-0-2\Portal_rel-2-0-2\build\WEB-INF\classes
[copy] Copying 18 files to C:\uPortal_rel-2-0-2\Portal_rel-2-0-2\build\WEB-INF\classes
[copy] Copied 36 empty directories to C:\uPortal_rel-2-0-2\Portal_rel-2-0-2\build\WEB-INF\classes

deploy:
[copy] Copying 519 files to C:\Program Files\Tomcat_4-0-4\Tomcat_4-0-4\webapps\portal2

BUILD SUCCESSFUL
Total time: 40 seconds
C:\uPortal_rel-2-0-2>

```

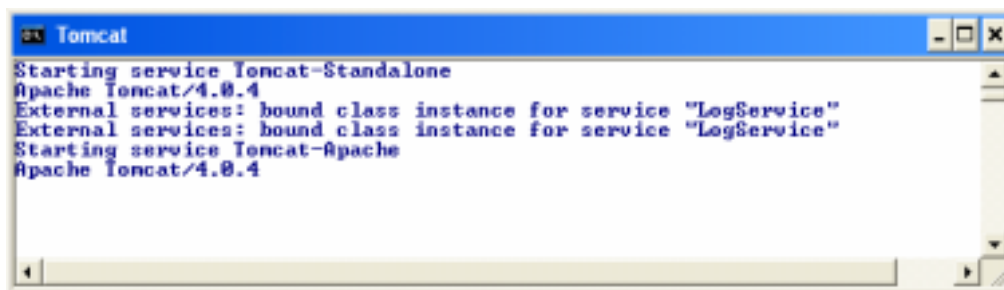
In order to start Tomcat go to the <TOMCAT\_HOME>/bin directory. Type the following command: `startup`. This should result in initiating a response such as those shown below in the next two windows.



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>cd C:\Program Files\Tomcat_4-0-4\Tomcat_4-0-4\bin

C:\Program Files\Tomcat_4-0-4\Tomcat_4-0-4\bin>startup
Using CATALINA_BASE:   ..
Using CATALINA_HOME:   ..
Using CATALINA_TMPDIR: ..\temp
Using JAVA_HOME:       C:\j2sdk1.4.0
C:\Program Files\Tomcat_4-0-4\Tomcat_4-0-4\bin>
```



```
Starting service Tomcat-Standalone
Apache Tomcat/4.0.4
External services: bound class instance for service "LogService"
External services: bound class instance for service "LogService"
Starting service Tomcat-Apache
Apache Tomcat/4.0.4
```

If you experience difficulties starting tomcat this may be due to a conflict with other software running on your PC. See Appendix B for details of how to check what software is running on which ports on your PC. Double check that Tomcat is actually running by opening a browser and pointing it towards the URL which contains the name of your server and the port on which the servlet container is running, for example `http://localhost:8080/`. The screen shot below shows the welcome screen of Tomcat.

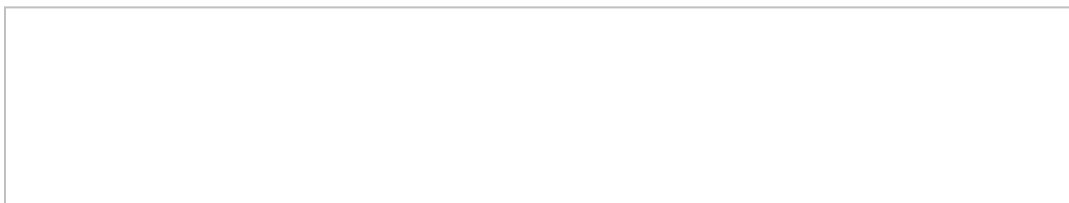


You are now ready to start exploring the portal – this is discussed in section 4 of this guide.

When you have finished your session with uPortal you should shut down Tomcat by opening a command window, going to `<UPortal_HOME>` and entering the command: `ant tomcatstop`. This will shut down Tomcat in an elegant manner and retain any changes you make to the HypersonicSQL database. Close down the HSQL database by pressing `Ctrl-C` in the command window that you used to start HSQL.

### 3.3 Installing on Unix

This section demonstrates how uPortal may be installed on a Unix platform; it continues the example using HypersonicSQL and Tomcat as the respective database server and servlet container and assumes that you have already installed and configured a database server and a servlet container. The text box below shows a typical directory structure containing 4 directories which hold all the systems required to run the uPortal only distribution.



The build tool Ant should be installed and the following environment variables set, for example if Ant has been installed in `/usr/local/ant`:

SHELL	FILE	TEXT TO APPEND TO FILE	ENTER TEXT AT PROMPT
<b>BOURNE</b> or <b>KORN</b>	~/.profile	<pre>ANT_HOME=/usr/local/ant JAVA_HOME=/usr/local/jdk-1.2.2 PATH=\${PATH}:\${ANT_HOME}/bin</pre>	<pre>&gt;export ANT_HOME=/usr/local/ant &gt; export   JAVA_HOME=/usr/local/jdk-1.2.2 &gt; export   PATH=\${PATH}:\${ANT_HOME}/bin</pre>
<b>C</b>	~/.login	<pre>setenv ANT_HOME /usr/local/ant setenv JAVA_HOME   /usr/local/jdk-1.2.2 set path=( \$path   \$ANT_HOME/bin )</pre>	<pre>&gt; setenv ANT_HOME   /usr/local/ant &gt; setenv JAVA_HOME   /usr/local/jdk-1.2.2 &gt; set path=( \$path   \$ANT_HOME/bin )</pre>

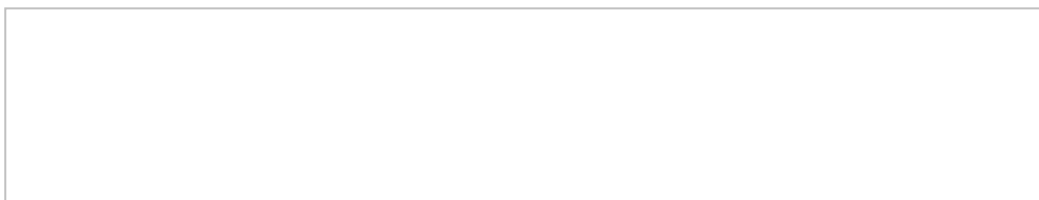
TABLE 1. Setting the Environment Variables.

You may set the environment variables by editing one of the following files and adding the appropriate text as shown in table 1. It is good practice to make a backup of the file before modifying it. Alternatively, you may set the environment variables from the command line as shown in table 1; this effects a temporary change and the variables will not be saved when you log out. In order to prompt the system to acknowledge the changes made in your modified initialisation file use the `source` command, e.g. `source .login`.

Open a terminal window and go to the `<HYPERSONICSQL_HOME>/lib` directory. Type the following command that should run the java file within the `hsqldb.jar` file and initiate the start of the server on the desired port:

```
java -cp hsqldb.jar org.hsqldb.Server -database uPortal2 -port 8887
```

An example of the correct response to this command is shown in the listing below.



You will notice that the directory now contains script and database files for uPortal2, for example:

```
zurg:/local/tempo/hsqldb_v.1.61/hsqldb_v.1.61/lib$ ls -l
total 616
-rw-r--r--  1 uportal  250      180909 Nov 11 11:03 hsqldb.jar
drwxr-xr-x  3 uportal  250      512 Nov 11 11:02 javax
-rw-r--r--  1 uportal  250       12 Nov 11 20:28 uPortal2.backup
-rw-r--r--  1 uportal  250       20 Nov 11 20:28 uPortal2.data
-rw-r--r--  1 uportal  250       70 Nov 11 20:28 uPortal2.properties
-rw-r--r--  1 uportal  250     99874 Nov 11 21:02 uPortal2.script
```

The next step is to create the tables and populate the database. Open another terminal window and go to the `<Uportal_HOME>` directory, which contains the build file `build.xml`. Enter the Ant command to build the database: `ant db`. It may take time for this action to be completed, the listing below shows an example of a successfully completed build.



```

zurg:/local/tempo/uPortal_rel-2-0-2$ ant deploy
Buildfile: build.xml

prepare:
  [mkdir] Created dir: /local/tempo/uPortal_rel-2-0-2/build
  [copy] Copying 462 files to /local/tempo/uPortal_rel-2-0-2/build

compile:
  [mkdir] Created dir: /local/tempo/uPortal_rel-2-0-2/build/WEB-INF/classes
  [javac] Compiling 260 source files to /local/tempo/uPortal_rel-2-0-2/build/WEB-INF/classes
  [javac] Note: Some input files use or override a deprecated API.
  [javac] Note: Recompile with -deprecation for details.
  [copy] Copying 16 files to /local/tempo/uPortal_rel-2-0-2/build/WEB-INF/classes/properties
  [copy] Copying 2 files to /local/tempo/uPortal_rel-2-0-2/build/WEB-INF/classes/org/jasig/portal/serialize
  [copy] Copying 81 files to /local/tempo/uPortal_rel-2-0-2/build/WEB-INF/classes
  [copy] Copying 18 files to /local/tempo/uPortal_rel-2-0-2/build/WEB-INF/classes
  [copy] Copied 37 empty directories to /local/tempo/uPortal_rel-2-0-2/build/WEB-INF/classes

deploy:
  [copy] Copying 919 files to /local/tempo/Tomcat_4-0-4/Tomcat_4-0-4/webapps/uPortal2

BUILD SUCCESSFUL
Total time: 3 minutes 49 seconds
zurg:/local/tempo/uPortal_rel-2-0-2$

```

In order to start Tomcat go to the <TOMCAT\_HOME>/bin directory. Type the following command: `./startup.sh`. This should result in initiating a response such as those shown below in the following listing.

```

uPortal/Tomcat_4-0-4/Tomcat_4-0-4> startup.sh
Using CATALINA_BASE:   /local/tempo/Tomcat_4-0-4/Tomcat_4-0-4
Using CATALINA_HOME:   /local/tempo/Tomcat_4-0-4/Tomcat_4-0-4
Using CATALINA_TMPDIR: /local/tempo/Tomcat_4-0-4/Tomcat_4-0-4/temp
Using JAVA_HOME:       /local/j2sdk1_3_1

```

If you experience difficulties starting tomcat this may be due to a conflict with other software running on your PC. See Appendix B for details of how to check what software is running on which ports on your PC. Double check that Tomcat is actually running by opening a browser and pointing it towards the URL which contains the name of your server and the port on which the servlet container is running, for example `http://localhost:8080/`.



You are now ready to start exploring the portal – this is discussed in section 4 of this guide.

The screen shot above shows the welcome screen of Tomcat. When you have finished your session with uPortal you should shut down Tomcat by opening a command window, going to <UPortal\_HOME> and entering the command: `ant tomcatstop`. This will shut down Tomcat in an elegant manner and retain any changes you make to the HSQL database.

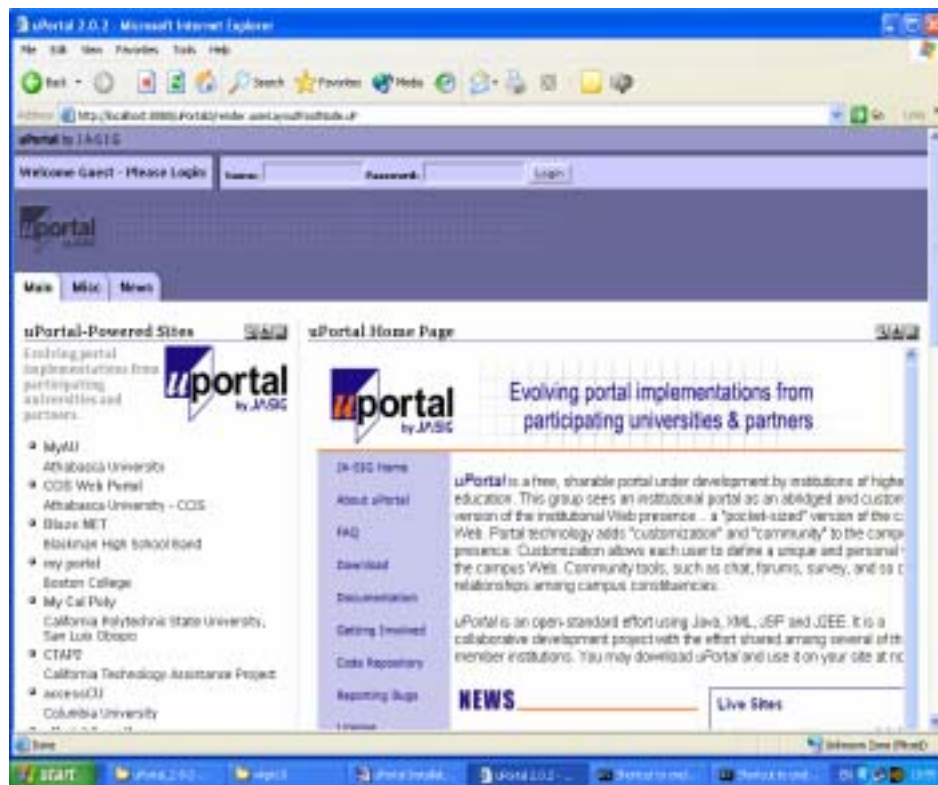
## 4. Exploring uPortal

In order to access uPortal you need a web browser - all of these, and later versions, are acceptable:

- Netscape Navigator 4.x
- Internet Explorer 4.x
- Mozilla 5.x
- Internet Explorer 5.x

Open uPortal by entering the URL including the server, port and portal name, for example: `http://localhost:8080/uPortal/`

If you have correctly installed uPortal you will see the opening page of uPortal which will appear similar to the following screen shot, the initial screen for the Guest user.



The uPortal framework ships with a simple structure that illustrates the typical features that a university or college may wish to implement. These can be adapted and customised as required. The initial portal screen illustrates the basic layout of a user's area which is subdivided into tabs, in this case *Main*, *Misc*, *News*. The Guest welcome screen reflects the

structure of the default presentation of “tab” and “column” elements. The unit sources of information presented by the portal are termed “channels”, for example on the Main tab, *uPortal-Powered Sites* and *uPortal Home Page* are both channels. At the top right hand corner of these two channels you will notice a group of three icon-buttons. These buttons, and others you will encounter, allow you to manipulate the channel in a variety of ways. For example, the three buttons at the top of the uPortal Home Page enable you to:

- **Focus** – display only the channel information
- **Minimise** – collapse the area displaying the information leaving just the channel title
- **Detach** - opens a separate window containing the only the channel

The Guest user does not require a password to login to the portal, nor is the user allowed to customise the basic layout. uPortal requires a user to login to rearrange the basic layout and retain the changes. You may login to the portal by assuming any one of several authenticated identities:

Username	Password
demo	demo
student	student
faculty	faculty
staff	staff
developer	developer

Once you login you are presented with a layout that has been modified to the needs of your group. The layout for the staff user is shown below.



You will notice in the screen shot above, showing the layout for the staff group, that there are three options not available to the Guest user – *Home*, *Preferences*, *Logout*.

The Preferences option allows the user to modify their layout by managing profiles, choosing skins and adding and removing channels and tabs.

The demo and development users have an additional function at their disposal – the option named Channel Admin which allows developers to modify existing channels or publish new ones. A good way to become familiar with this facility is to add some channels of your own using the default uPortal channel types. Later you may want to attempt writing your own channels. There are two tutorials on the JA-SIG web site that explain how to write and publish simple channels. You will find these on: <http://mis105.mis.udel.edu/ja-sig/uportal/developers/> .



## 4.1 Downloading the uPortal Source Code

In order to update to the latest version of uPortal you will occasionally need to download the latest version and incorporate it into your existing system.

Users who would like access to the source code for uPortal can do so through the web front end to the uPortal CVS repository; please note that this access is **read-only**. The URL for the web front end to CVS is:

```
http://mis105.mis.udel.edu/cgi-bin/viewcvs.cgi
```

To download the source code, which is available in zip and tar.gz formats, go to:

```
http://www.yale.edu/tp/portal-dist/
```

To access CVS on Windows you will need to install a CVS client, if you wish to write into CVS you also need an SSH client/tunnel. Details of how to obtain and install these are given on the JA-SIG website at:

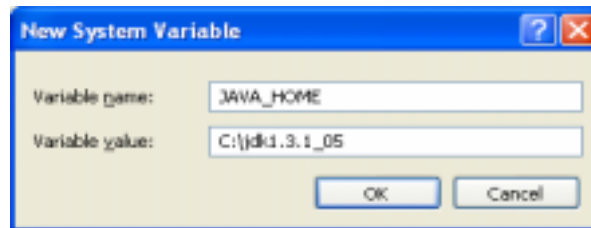
```
http://mis105.mis.udel.edu/ja-sig/uportal/cvs.html
```

If you wish to gain write access to the CVS uPortal source code then follow the guidance given at the JA-SIG website at:

```
http://mis105.mis.udel.edu/ja-sig/uportal/cvsaccess.html
```

## Appendix A — Setting JAVA\_HOME

On Windows this may be done by using the *System Properties* tool from the Control Panel. Elect to add a new *Environment Variable* and enter the complete path of the directory holding the JDK, an example is shown below.



On Unix you will need to set a new environment variable if `JAVA_HOME` has not already been set, you can check your current environment variables by entering the command: `env`. Depending on the shell you are using, `JAVA_HOME` may be set in various ways. If you are not sure which shell you are using use the `ps` command to find out; it will show you if you are running **csh** – the C shell, **sh** – the Bourne shell, or **ksh** – the Korn shell. You may set the `JAVA_HOME` variable by editing one of the following files and adding the appropriate text as shown in the table 2. You may use a text editing program such as **vi**, **emacs** or **pico** to open, edit and save the information to the file. It is wise to make a backup of the file before modifying it. Alternatively, you may set the `JAVA_HOME` variable from the command line as shown in table 2; this effects a temporary change and the variable will not be saved when you log out. To prompt the system to acknowledge the changes made in your modified initialisation file use the **source** command, e.g. `source .login`.

SHELL	FILE	TEXT TO APPEND TO FILE	ENTER TEXT AT COMMAND LINE
<b>BOURNE</b> or <b>KORN</b>	<code>~/.profile</code>	<code>JAVA_HOME=&lt;path to JDK&gt;</code> e.g. the path to the JDK might be: <code>/local/j2sdk1_3_1</code>	<code>&gt; JAVA_HOME=&lt;path to JDK&gt;</code> <code>&gt; export JAVA_HOME</code>
<b>C</b>	<code>~/.login</code>	<code>setenv JAVA_HOME=&lt;path to JDK&gt;</code>	<code>&gt;set JAVA_HOME=&lt;path to JDK&gt;</code> <code>&gt; export JAVA_HOME</code>

TABLE 2. Setting the `JAVA_HOME` variable.

## Appendix B — Checking ports

To ensure that nothing is running on ports 8080 and 8087 find out the state of the ports on your machine by entering the following command: `netstat -a`. If something is running on port 8080 then it is probable that another process has laid claim to that port. By default Tomcat attempts to bind to port 8080 at startup. If you need to assign Tomcat to another port open the file: `<TOMCAT_HOME>/conf/server.xml`

`<TOMCAT_HOME>` is the Tomcat directory inside your uPortal directory. Search for "8080" in the text and change it to a port that is not in use, and is greater than 1024; ports less than or equal to 1024 require superuser access in order to bind to them. Remember to replace the "8080" in the URL you are using to access Tomcat. For example, if you change the port to 1977, then you should request the URL: `http://localhost:1977/uPortal2` when accessing uPortal through a web browser.

## Appendix C — Ant

Apache Ant is an open source build tool that uses XML-based files to configure target directories. It is written in Java and uses classes for the compilation and execution of a variety of build related commands. Ant was designed to overcome some of the problems, or undesirable features, of other build tools such as *Make*. To obtain the latest version of Ant and further information about it, go to the Apache site at:

<http://jakarta.apache.org/ant/>

A primer on *Installing Software with Jakarta Ant* can be found at:

<http://www.onjava.com/pub/a/onjava/2002/06/05/ant.html>

You will at least need to know about *Projects*, *Targets*, and *Properties*, and also what a build file looks like – an introduction to these topics may be found at:

<http://jakarta.apache.org/ant/manual/using.html#projects>

Each XML-based Ant build file contains one project; a project is defined by 3 attributes – *name*, *default* and *basedir*. A minimum requirement is that you specify a default directory that will be used in cases where no target directory is provided; the attributes of name and basedir are optional. A copy of the Ant build file which is included with the uPortal Quick Start distribution is shown below, the project definition is found at near the start of the file. A project may specify one or more targets, a *target* has 5 attributes – *name*, *depends*, *if*, *unless*, and *description*. Only the name attribute is mandatory. Each target defines one or more *tasks* that can be executed. Tasks are defined within the target tags and give direction for a piece of code to be executed. For example, in the build file shown below, for the target `hsq1`, the first task is a piece of code - an OS command - will display the line "Starting Hypersonic SQL":

```
<echo message="Starting Hypersonic SQL"/>
```

A complete list of Ant tasks is available at:

<http://jakarta.apache.org/ant/manual/using.html#projects>

In the sections dealing with the installation of uPortal, the Ant scripts were invoked from the command line. To see a summary of how to use the Ant command type: `ant -help`. By default, Ant will look for a build file in the current directory. If no default file is found and the `-file` option is turned on, Ant will continue searching up the directory tree until it reaches the root of your system. Several open source projects provide the ability to incorporate Ant inside of an IDE, further information on this topic can be found at: <http://jakarta.apache.org/ant/manual/>.

```

<!-- This build.xml contains targets for starting Hypersonic SQL
      (which listens on port 8887)
      and starting and stopping Tomcat 4.0.4.-->

<project name="uPortal_Quick_Start" default="tomcatstart" basedir=".">

<property environment="env"/>

<!-- The "hsq1" target starts HypersonicSQL -->

  <target name="hsq1">
    <echo message="Starting Hypersonic SQL"/>
    <java fork="true" dir="{basedir}" classname="org.hsq1.Server">
      <classpath>
        <path element path="{basedir}/HypersonicSQL_1-4-3/demo/hsq1.jar"/>
      </classpath>
      <arg value="-database"/>
      <arg value="{basedir}/HypersonicSQL_1-4-3/demo/uPortal2"/>
      <arg value="-port"/>
      <arg value="8887"/>
    </java>
  </target>

<!-- The "tomcatstart" target starts Tomcat -->

  <target name="tomcatstart">
    <echo message="Starting Tomcat 4.0.4"/>
    <java fork="true"
          dir="{basedir}" classname="org.apache.catalina.startup.Bootstrap">
      <classpath>
        <path element path="{basedir}/Tomcat_4-0-4/bin/bootstrap.jar"/>
        <path element path="{env.JAVA_HOME}/lib/tools.jar"/>
      </classpath>
      <sysproperty key="catalina.home" value="{basedir}/Tomcat_4-0-4"/>
      <!-- jvmarg value="-Dcatalina.home={basedir}/Tomcat_4-0-4"/-->
      <arg value="start"/>
    </java>
  </target>

```

## Appendix D — Web Application Environments

Among the community of universities who have implemented uPortal, a number of web environments have been successfully utilised. The following products are among the most popular choices:

- **Caucho Resin** - available from <http://www.caucho.com/resin/>. Resin is a fast servlet and JSP engine supporting load balancing for increased reliability. Resin encourages separation of content from style with its XSL support. Servlets can generate simple XML and use an XSL filter to format results for each client's capability, from PDAs to Mozilla.
- **Apache HTTP Server** – available from <http://httpd.apache.org/>. The Apache HTTP Server is a collaborative software development effort aimed at creating a robust, commercial-grade, featureful, and freely-available source code implementation of an HTTP (Web) server. It is well established and is often embedded within commercial enterprise application servers
- **Apache Tomcat** – Tomcat is the servlet container that is used in the official Reference Implementation for the Java Servlet and JavaServer Pages technologies. It is available from <http://jakarta.apache.org/tomcat/>
- **IBM Websphere** – Websphere is a collection of enterprise software tools aimed at companies who require complex e-business solutions. It includes transaction servers and tools, business integration systems and a portal. It is available from <http://www-3.ibm.com/software/inf01/websphere/>
- **BEA WebLogic** – The WebLogic platform includes several sophisticated components such as the *WebLogic Server*, *WebLogic Integration*, *WebLogic Portal*, and *WebLogic Workshop*. It is geared towards providing a single, unified, easy-to-use infrastructure platform for application development, deployment, and management. WebLogic in all its forms is available from <http://www.bea.com/products/>
- **JBoss** - available from <http://www.jboss.org/>. JBoss started as an EJB container but has now developed into a fully functional application server facilitating the full J2EE stack.