



**ATTIX<sup>5</sup>**  
**BACKUP PROFESSIONAL**

WEBACCESS INSTALLATION GUIDE

USER MANUAL v5.x

## Table of Contents

---

<b>1. Introduction</b> .....	<b>2</b>
Prerequisites .....	2
Components .....	2
Unicode Support .....	3
<b>2. Installation</b> .....	<b>4</b>
The Java Software Development Kit (JSDK) 1.5 .....	4
The Tomcat Servlet Engine v5.5 .....	4
Installing WebAccess .....	4
<b>3. Configuration</b> .....	<b>5</b>
Tomcat Documentation .....	5
WebAccess as the only Tomcat application .....	5
<b>4. Customising WebAccess</b> .....	<b>6</b>
/Web-INF/web.xml .....	6
/Web-INF/defaults.properties .....	6
/Web-INF/classes/log4j.xml .....	7
Briefcase upload limit .....	7
<b>5. Running WebAccess</b> .....	<b>8</b>
<b>6. Administration Interface</b> .....	<b>9</b>
Modifying the WebAccess Admin username/password .....	9
<b>7. Addendum A – Tomcat Customisation</b> .....	<b>10</b>
Configuring Tomcat for SSL .....	10
Configure Tomcat for WebAccess .....	11

### 1. Introduction

Backup Professional protects your critical documents and data files so that you can recover them quickly and easily in the event of either their loss, or the loss of the computer that you keep them on.

WebAccess provides secure, remote access from any web browser to your backup data set. This document describes how to install WebAccess.

#### PREREQUISITES

You must be familiar with installing software on the Windows platform. The installation must be done from an account with administrator privileges.

The WebAccess application requires both Java and a Java Servlet engine. This document describes how to install these and prior experience of these is not required.

*Note: While numerous servlet engines are available, Tomcat is a Jakarta initiative that is available free for commercial use. WebAccess adheres strictly to the JSP/Servlet deployment specification and as a result there should not be any problem deploying WebAccess on a different Servlet 2.3/JSP 1.2 compliant servlet engine. Installation and deployment instructions for these servlet engines are, however, not documented here.*

#### COMPONENTS

The following components are required:

Name	Available From	Usage
Java SDK 1.5	<a href="http://java.sun.com/j2se">java.sun.com/j2se</a>	Supports the Java Server Pages used by WebAccess
Tomcat 5.5	<a href="http://jakarta.apache.org/tomcat">jakarta.apache.org/tomcat</a>	Runs the Java Server Pages used by WebAccess and supports secure communication with a web browser via SSL
webaccess.zip	Attix5	Contains the components for the WebAccess application.
<b>WebAccess Components:</b>		
webaccess.war	The WebAccess application as a Web Archive file for Tomcat.	
server443.xml and server8444.xml	Tomcat configuration file for WebAccess. One of these may be used for a new Tomcat installation.	
sslkeystore	File containing a SSL certificate. May be used when configuring Tomcat to support SSL.	

### UNICODE SUPPORT

As of Backup Professional v3.2.0, WebAccess supports Unicode. However, one configuration operation is required for this to work correctly.

Tomcat uses the default encoding type for your JVM when it performs any string/data encoding operations. For Unicode support, the “file-encoding” must be explicitly set to UTF-8. In addition, a MIME encoding type for the JVM must also be set to UTF-8.

Tomcat 5.5: To set these encoding types for Tomcat, you will need to set the JAVA\_OPTS environment variable to include the following options:

Example (Windows): SET JAVA\_OPTS=' -Dfile.encoding=utf-8 -Dmail.mime.charset=utf-8'

Example (Unix): export JAVA\_OPTS=' -Dfile.encoding=utf-8 -Dmail.mime.charset=utf-8'

These encoding types can also be configured using the Tomcat system tray icon, if Tomcat is installed on a Windows machine.

Be sure not to overwrite any unrelated existing JAVA\_OPTS settings. You are also not advised to modify the existing Tomcat deployment scripts, such as catalina.bat or catalina.sh.

## 2. Installation

This chapter describes how to install the different components needed to run WebAccess. Each section will discuss in more detail how to configure the components.

### THE JAVA SOFTWARE DEVELOPMENT KIT (JSDK) 1.5

If Java SDK V1.5 is already installed then you may skip this step. If you were not provided with the installer, you can download the current JDK from [java.sun.com/j2se](http://java.sun.com/j2se) (look for a quick download link on the right-hand side) and install it according to the installer instructions.

### THE TOMCAT SERVLET ENGINE V5.5

If you already run Tomcat V5.5, you may skip this step but please read through the notes.

If you were not provided with the installer, you can download Tomcat 5.5 from [jakarta.apache.org/tomcat](http://jakarta.apache.org/tomcat). Look for the Download heading on the left-hand side and click on the Binaries link. Install this according to the installer instructions. The installer will prompt you to accept the JSDK that it locates. Ensure that this is the one installed in the previous step. If installing on Windows, during the installation process, **enable the checkbox to install the NT service**. This allows you to start and stop Tomcat from the Services window in the Control Panel. Do not deselect any boxes.

When the installation is complete, you may test it from a web browser by using the URL: <http://localhost:8080/>. This will display the top-level documentation for Tomcat.

### INSTALLING WEBACCESS

The application is provided as a WAR (Web ARchive) file named `webaccess.war`. Copy this file into the tomcat `webapps` directory. After a short duration, Tomcat should expand this WAR file automatically and create a `webaccess` application directory. If this does not happen, then open the WAR file using WinZip (or any other zip utility), and extract the contents into a `webaccess` sub-directory within the Tomcat `webapps` directory. Be sure to “extract using folder names”.

You will see a directory structure in the `/webaccess` folder containing at least the following:

- `/css`
- `/html`
- `/images`
- `/js`
- `/WEB-INF`

## 3. Configuration

This chapter describes how to configure WebAccess on Tomcat. If you have installed Tomcat only to support WebAccess follow the instructions in the chapter. If you support other applications on Tomcat as well, refer to **Addendum A: Tomcat Customisation** before continuing with the next chapter.

### TOMCAT DOCUMENTATION

A more detailed explanation of how to configure Tomcat for SSL may be found at <http://jakarta.apache.org/tomcat/tomcat-5.5-doc/ssl-howto.html>. Note that since you are running JDK1.5.x, you do **not** need to install the additional JSSE libraries referred to in the Tomcat SSL configuration guide. This is only required if you are running an earlier JSDK.

Also, the keystore file that is generated in the instructions is not necessary unless you wish to provide an authenticated certificate for your company. For your convenience, Attix5 provides a sslkeystore file that you can use for your initial evaluation/demonstration.

### WEBACCESS AS THE ONLY TOMCAT APPLICATION

If you are only using this installation of Tomcat to support WebAccess, and the Tomcat configuration file, server.xml, has not been edited after the Tomcat installation, then you may use either the server443.xml or server8444.xml file, both supplied by Attix5.

The default port number for SSL is 443. If you are **NOT** running the Attix5 Backup Professional Storage Platform (using ports 443/8443) on the same machine, **AND** you are not running IIS (which uses ports 80 and 443), then you can use the file server443.xml.

If you **are** running the Attix5 Backup Professional Storage Platform on the same machine using ports 443/8443, **or** you are running IIS using port 443, then you will need to configure WebAccess to communicate on a different port. The convention is to use port 8444 and you should then use the file server8444.xml.

Note that if you can use port 443 for WebAccess then it will make it easier for your users to access their backup sets. Using port 443, a user can type the URL <https://hostname/webaccess>. If you use port 8444, then the user must type <https://hostname:8444/webaccess>.

#### To configure Tomcat for WebAccess:

1. Locate the Tomcat "conf" directory and rename [server.xml](#) to [server.xml.tomcat](#).
2. Copy [server443.xml](#) or [server8444.xml](#) to this directory and rename it [server.xml](#).
3. Copy the Attix5 file [sslkeystore](#) to the same directory.
4. To bind to a specific port add 'address="x.x.x.x"' in the Connector port= section.

Finally, restart Tomcat (on Windows use the NT Services window). You can now test this by using the URL: <https://hostname> or <https://hostname:8444>. This will display the Tomcat home page. If you do not want your users to see the Tomcat homepage if they type in this URL by mistake, create a file named index.html and place it in the Tomcat webapps/ROOT folder. This could be an empty page or it could contain information for your users such as a link to the WebAccess page: <https://hostname/webaccess> or <https://hostname:8444/webaccess>.

## 4. Customising WebAccess

The WAR file that you have received from Attix5 needs to be configured for your environment. The following sections describe the various files that will need to be (or can be) customised. The files are indicated relative to your `$TOMCAT_HOME/webapps/webaccess` directory. These changes can be set by using the Administration Interface – see the [Administration Interface](#) chapter for more information.

### /WEB-INF/WEB.XML

This is the main web application descriptor file. It indicates all components of the web application and their associated properties including URL mappings, session timeouts etc.

- **session timeout** : this is specified in minutes and can be modified according to your internal policy for session duration

```
<session-config>
  <session-timeout>2</session-timeout>
</session-config>
```

### /WEB-INF/DEFAULTS.PROPERTIES

This file needs to be configured for your Attix5 Storage Platform environment.

- **SSL\_SERVER** : this indicates the location of your Attix5 Secure Storage Platform name server.

```
SSL_SERVER=bpserver:443
```

- **GROUP\_NAME** : this is the default group name to use in WebAccess

```
GROUP_NAME= acmegroup
```

- **sitename** : the label/name to use when referring to this site in the WebAccess pages.

```
sitename=WebAccess
```

- **default.faxnum** : the default fax number to display when referencing your company's fax information

```
default.faxnum=2721000000
```

- **default.first.page** : the first available view to use once the user has logged in. Currently, only 'backupset' is supported. Other options will be made available as they are added to the WebAccess application.

```
default.first.page=backupset
```

- **show.group** : indicates whether or not to show the default group setting on the login page.

```
show.default.group=true
```

- **fixed.group** : this indicates whether the group setting for this instance of WebAccess is fixed or can be modified by the user. This setting works in combination with the previous setting, show.default.group.

```
fixed.group=false
```

# ATTIX<sup>5</sup> BACKUP PROFESSIONAL

## WEBACCESS V5.x INSTALLATION GUIDE

- **mail.host** : this is the DNS name or IP address of the SMTP host that WebAccess will use to send emails and fax emails.

```
mail.host=mail.acme.com
```

- **mail.from** : this is the default 'From:' address that will be used in all email sent from WebAccess.

```
mail.from=webaccess@acme.com
```

- **max.attachment.size** : this is the maximum total attachment size, in bytes, that can be added to a WebAccess email or fax email.

```
max.attachment.size=50000
```

- **fax.gateway.domain** : this is the fax gateway used to send faxes. WebAccess sends an email to this gateway and the gateway, in turn, converts the email to a fax and sends it to the intended destination.

```
fax.gateway.domain=vax.co.za
```

- **default.fax.from** : this is the default email address used as the 'From' address when sending a fax (to the fax email gateway). The fax gateway uses this email address for authentication.

```
default.fax.from=faxes@acme.com
```

**Note:** If you do not plan to setup faxing functionality, just comment out these lines with '#', or leave their values blank and WebAccess will omit the fax toolbar buttons.

### /WEB-INF/CLASSES/LOG4J.XML

This file is used to control logging output for the WebAccess application. You can use this to enable debugging information as well as to limit the specific output within a given logging priority.

This file can be modified at run-time and WebAccess will detect the modification and amend its logging accordingly.

For more information on log4j configuration, go to: <http://jakarta.apache.org/log4j>

### BRIEFCASE UPLOAD LIMIT

Open the file `\webaccess\WEB-INF\classes\webwork.properties` and modify the entry `webwork.multipart.maxSize=10485760` with the applicable value in Bytes. Save and close the file.

**Note:** you must restart the Tomcat service to apply this change.

## 5. Running WebAccess

Start the Apache Tomcat, if it is not already started. For Windows users you can do this from the **Services** Window that you will find in the **Control Panel**. Login in to [https://your\\_server\\_name\[:<port\\_number>\]/webaccess](https://your_server_name[:<port_number>]/webaccess) from any web browser. Remember to include the port number in the URL if you are not using port 433 for SSL. You should see a page similar to this:

Type in a valid group name (if applicable), username, password and key, and log in. If all goes well, you should see a page similar to the following:

File Name	Size	Date
No Files		

Congratulations on successfully installing the WebAccess web application from Attix5! If you need more information about the WebAccess application, please refer to the [Attix5 Backup Professional WebAccess User Manual](#).

## 6. Administration Interface

The administrator interface allows you to configure the WebAccess properties via your web browser, as opposed to modifying properties manually. Ensure that Apache Tomcat is started.

- From your web browser, go to `https://your_server_name[:<port-number>]/webaccess/admin.jspx`
- If WebAccess is the only Tomcat application, or if the default Tomcat installation has not been modified, then the default username and password for administration are “admin” and “admin”.
- If WebAccess is the only Tomcat application, then you should modify this password by editing the file `$TOMCAT_HOME/conf/tomcat-users.xml` to set a new password. This is described below.
- If the admin username or password has been changed, then you must edit the WebAccess configuration file, `web.xml`, as described below.
- Once you’ve entered the correct username and password, you will then be forwarded to the administration interface. This allows you to modify various configuration parameters as displayed on the admin page.

### MODIFYING THE WEBACCESS ADMIN USERNAME/PASSWORD

Modifying the WebAccess admin username and password is accomplished by modifying the username and password in the `tomcat-user.xml` file:

```
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users>
  <role rolename="admin"/>
  .
  <user username="admin" password="admin" roles="admin"/>
</tomcat-users>
```

Ensure however that you do not change the role value without also updating the value reflected in `/WEB-INF/web.xml`.

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>Admin</web-resource-name>
    <url-pattern>/admin.jspx</url-pattern>
    <url-pattern>/configure.jspx</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <role-name>admin</role-name>
  </auth-constraint>
</security-constraint>
<login-config>
  <auth-method>FORM</auth-method>
  <form-login-config>
    <form-login-page>/show-admin-login.jspx</form-login-page>
    <form-error-page>/show-admin-login.jspx</form-error-page>
  </form-login-config>
</login-config>
<security-role>
  <role-name>admin</role-name>
</security-role>
```

### 7. Addendum A – Tomcat Customisation

This section describes in detail what needs to be changed if you are using Tomcat for other applications as well and cannot use the supplied xml files. *Note: If you are only using Tomcat for WebAccess, you do not have to modify any of these settings.*

#### CONFIGURING TOMCAT FOR SSL

If you cannot use the server.xml files supplied by Attix5, then edit the existing server.xml file as follows:

- Look for the following lines:

```
<!-- Define a SSL Coyote HTTP/1.1 Connector on port 8443 -->
<!--
<Connector className="org.apache.coyote.tomcat4.CoyoteConnector"
    port="8443" minProcessors="5" maxProcessors="75"
    enableLookups="true" acceptCount="10" debug="0"
    scheme="https" secure="true"
    useURIVValidationHack="false">
    <Factory className="org.apache.coyote.tomcat4.CoyoteServerSocketFactory"
        clientAuth="false" protocol="TLS" />
</Connector>
-->
```

- Delete the HTML comment lines **marked** above (this will make this connector available for SSL requests) and change all “8443” references in the `server.xml` file to “443”.
- Next we indicate to Tomcat where to find the certificate keystore file. The default behaviour for Tomcat with SSL configuration is to look for a “.keystore” file in the home directory of the id under which Tomcat is running. Additionally, this “.keystore” file must have the password ‘changeit’. If this suits your environment, copy the `sslkeystore` file that you unzipped from `webaccess.zip` to the home directory of this id, as indicated previously.
- If this is not suitable for your environment, and you would perhaps prefer to use your own company-specific keystore file, look for the following lines in the connector block you just uncommented above:

```
<Factory className="org.apache.coyote.tomcat4.CoyoteServerSocketFactory"
    clientAuth="false" protocol="TLS" />
```

- Modify the lines to include the following two attributes:

```
keystoreFile="<location of keystore file>"
keystorePass="<your preferred password>"
```

- The result should look something like this:

```
<Factory className="org.apache.coyote.tomcat4.CoyoteServerSocketFactory"
    clientAuth="false" protocol="TLS"
    keystoreFile="C:\foo\bar\acmekeystore"
    keystorePass="f00zb@11" />
```

Tomcat is now configured to use SSL.

### CONFIGURE TOMCAT FOR WEBACCESS

- Edit the file `$TOMCAT_HOME/conf/tomcat-users.xml`

This is to enable you to login to the administration interface of WebAccess. This requires a role called 'admin' and a user called 'admin'. These are provided by default in Tomcat version 4.1 and the WebAccess configuration file is set up to use them automatically. If your tomcat-users.xml file does not have these defined, then add them as shown below. Note that if you use any user and role names other than "admin" then you must edit the WebAccess configuration file web.xml as explained below.

Create a role called '**admin**' and a user called '**admin**' if there is not one already and specify a password for the admin login:

```
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users>
  <role rolename="admin"/>
  .
  .
  <user username="admin" password="admin" roles="admin"/>
</tomcat-users>
```

**Note:** choose a reasonable password

This is all you need to do to configure the administration authentication for WebAccess. Note that web.xml has already been configured for these roles and will need to be changed if you choose other role and user names.

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>Admin</web-resource-name>
    <url-pattern>/admin.jsp</url-pattern>
    <url-pattern>/configure.jsp</url-pattern>
  </web-resource-collection>

  <auth-constraint>
    <role-name>admin</role-name>
  </auth-constraint>
</security-constraint>
<login-config>
  <auth-method>FORM</auth-method>
  <form-login-config>
    <form-login-page>/show-admin-login.jsp</form-login-page>
    <form-error-page>/show-admin-login.jsp</form-error-page>
  </form-login-config>
</login-config>
<security-role>
  <role-name>admin</role-name>
</security-role>
```

After you have modified all the necessary changes, please refer to the previous chapter, [Customising WebAccess](#), to set the default connection settings for WebAccess.