Background Information

Application Designer runs as a web application in any kind of servlet container supporting the servlet API. Information on the required version of the Servlet specification is provided in the section *Hardware and Software Requirements*.

There are multiple servlet containers available on the market. Tomcat is the most commonly used servlet container of the open source world, but there are also others that in most cases are part of Java EE application server environments: Websphere from IBM, BEA Weblogic, Sun IPlanet, SAP Web Application Server, JBoss and many others.

This chapter provides background information that is helpful when transferring Application Designer to a servlet container of your choice. This chapter is not designed to be useful when installing Application Designer for the first time - use the standard Tomcat servlet container for doing your first steps.

This chapter covers the following topics:

- Application Designer Web Application
- Creating a Second Application Designer Web Application inside Your Tomcat Installation
- Adding Application Designer to an Existing Web Application
- Building a Web Application Archive

Application Designer Web Application

Have a look at your installation's <*installdir*>*/tomcat/webapps* directory. You see the subdirectory *cis*. This is the name of the web application which is used by default.

The *cis* directory contains the following subdirectories:

```
/cis/                   General information.
    /config/            Configuration files.
    /licensekey/        License key file.
    /styles/            Standard location of style sheet files.
    /temp/              Temporary files.
/cisdemos/              Application Designer demo project.
/cisdocumentation/      Documentation.
/cispinedit/            Directory for pinEdit (word processor for HTML pages).
/EclipsePlugin/         Directory for the Eclipse plug-in.
/HTMLBasedGUI/          Application Designer base project.
/META-INF/              Web application standard directory.
/SWTBasedGUI/           Directory for the SWT client.
/WEB-INF/               Web application standard directory.
    /web.xml            Web application's configuration file.
    /classes/           Web application's classes.
    /lib/               Web application's libraries.
```

Configuration of *web.xml*

As with any other web application, the *web.xml* file contains configuration information which you have to take care of when deploying. The file contains information about the servlets which are part of the Application Designer web application. There is one servlet `Connector` that contains important configuration parameters: `cis.home` and `cis.log`. See the description of the *web.xml* file for further information.

If only working with Application Designer GUIs, you do not have to pay further attention to the *web.xml* file. If working in more complex scenarios in which you might also define other servlets or when you access Enterprise Java Beans, you have to adapt the *web.xml* file to your needs (for example, you have to add bean reference information).

Multiple Deployments

The name "cis" for the web application is just the default installation name for the web application. You can easily name it in a different way. As with any other correct web application, you can also deploy it multiple times to the same servlet engine.

This is important for you in case you add the Application Designer web application to an existing web application on your side. You add Application Designer to your web application as you add normal libraries, i.e. Application Designer is now running under the control of your own web application.

The following sections provide more information on this aspect.

Creating a Second Application Designer Web Application inside Your Tomcat Installation

To demonstrate the usage of Application Designer as a standard web application, you can create a second Application Designer instance inside your Tomcat environment (and a third, fourth, etc.): simply copy the whole *cis* web application directory and paste it with a new name in Tomcat's *webapps* directory.

Step by step: let us assume that the name of the new application is *secondcis*:

- Copy the directory *<installdir>/tomcat/webapps/cis*.
- Paste it in the *webapps* directory and rename the new directory to *secondcis*. As a result, you have the following directories:

  ```
  <installdir>/tomcat/webapps/cis
                              /secondcis
  ```

Each web application instance now contains an independent Application Designer. It is no problem to run different versions of Application Designer inside one servlet container.

If you want to access the demo workplace of the first instance, you reference the following:

http://localhost:51000/cis/HTMLBasedGUI/workplace/demo.html

If you want to access the demo workplace of the second instance you reference the following:

http://localhost:51000/secondcis/HTMLBasedGUI/workplace/demo.html

You do not have to make further configuration steps when working with Tomcat: the standard *web.xml* contains the REALPATH parameter as a pointer to the web application's directory. This means that the directory path is determined automatically. If you create a second web application in a different servlet container, you might have to adjust the *web.xml* file after copying so that the new web application points to the right directory.

Adding Application Designer to an Existing Web Application

The same way we created a second instance of Application Designer in the section before, we can now add Application Designer to your exisiting web application in which you want to use Application Designer functions. Just copy the *cis* directory inside your web application's directory and merge the *web.xml* files of both web applications.

Building a Web Application Archive

A web application archive can be simply built by zipping one whole web application directory into a corresponding *.war* file.

However, there may also be more complex scenarios in which you want your deployable runtime to be structured differently compared to your design time. See the section *Design Time Mode and Runtime Mode*.

Application Designer also provides the WAR Packager tool that takes over the building of the *.war* file. This tool is designed to cover very basic scenarios of packaging a web application archive file. It should only be used if you are not familiar with ANT build processes. See *WAR Packager* in the *Development Workplace* documentation for more information.