WAR Packager

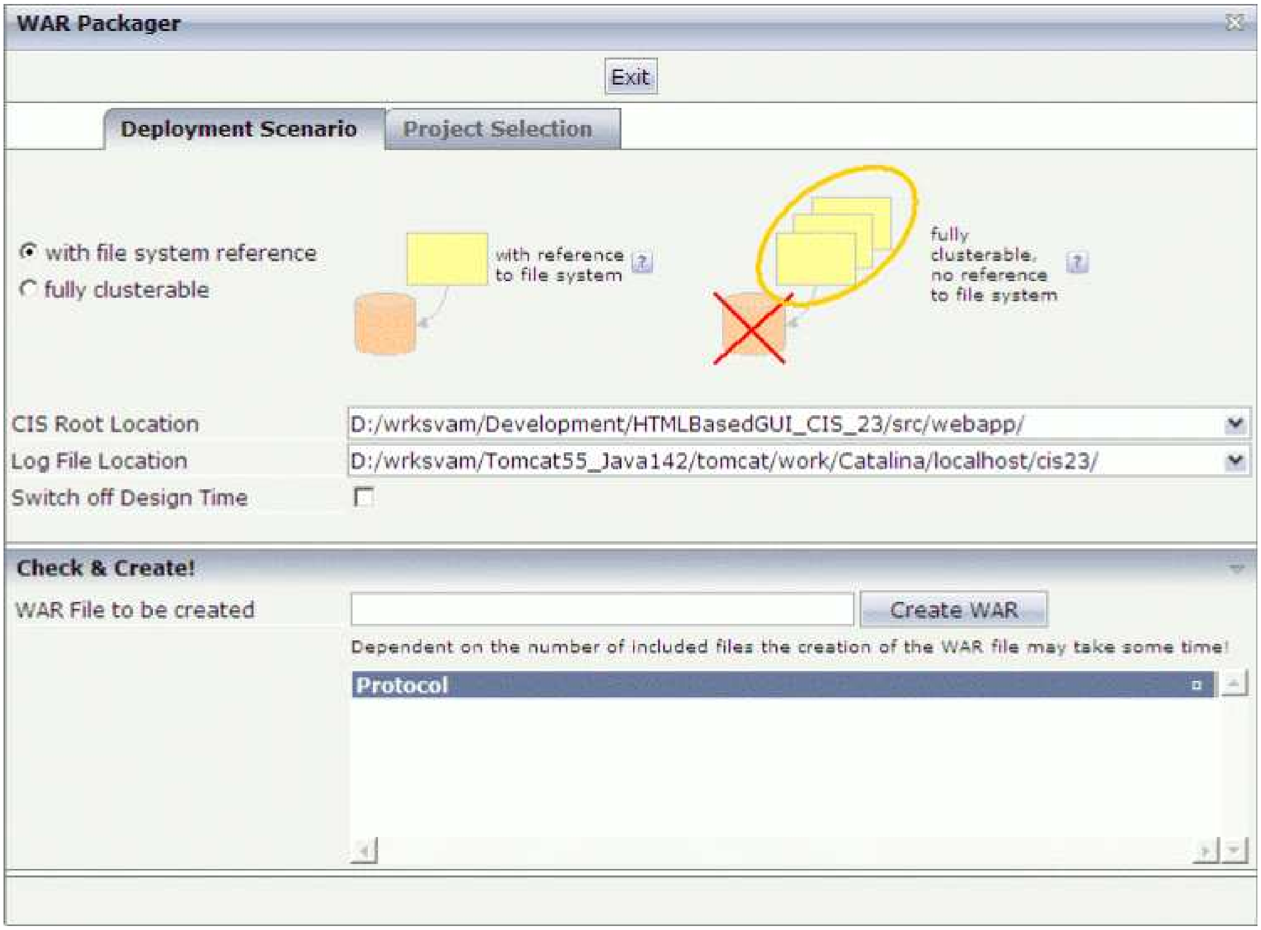
In order to deliver the web applications that you have built with Application Designer, you need to create a web archive (WAR file). The WAR Packager provides an easy-to-use graphical user interface that you can use to create the web archive.

This chapter covers the following topics:

- Invoking the WAR Packager
- Types of Generation
- Creating a Web Archive
- Java API for WAR File Generation

Invoking the WAR Packager

When you invoke the WAR Packager, the following dialog appears.



- ● with file system reference
- ○ fully clusterable

CIS Root Location	D:/wrksvam/Development/HTMLBasedGUI_CIS_23/src/webapp/
Log File Location	D:/wrksvam/Tomcat55_Java142/tomcat/work/Catalina/localhost/cis23/
Switch off Design Time	□
WAR File to be created	

Dependent on the number of included files the creation of the WAR file may take some time!

To invoke the WAR Packager

- In the **Development Tools** node of the navigation frame (which is visible when the **Tools & Documentation** button has previously been chosen), choose **WAR Packager**.

Types of Generation

When you create a web archive, you have to define the type of generation:

- Web Archive with Reference to the File System
- Web Archive without Reference to the File System

Web Archive with Reference to the File System

This is the standard mode in which Application Designer is delivered. The name of the corresponding option in the WAR Packager is **with file system reference**.

In the web archive's *web.xml* file there are references to the web application's root in the file system and to the location where log files are written. The development functions of Application Designer (such as the Layout Painter) require access to the file system and as a consequence know where to read and write files. In addition, the Application Designer class loader, which simplifies development, accesses application classes through the file system (*appclasses* directory of a project).

This mode is ideal for development but has some disadvantages in clustered runtime scenarios:

- With some applications servers, you do not know the location of the deployed files (or: you should not know the location at all).
- In clustered scenarios, you would require to have one directory that is the same for all cluster nodes.

Web Archive without Reference to the File System

This is the preferred deployment mode for applications. The name of the corresponding option in the WAR Packager is **fully clusterable**.

The resources are not read from the file system but are read via the servlet engine's web resource reader. The log is written into the log of the servlet engine as well. Classes are not read via the Application Designer class loader but are read from the directory *WEB-INF/classes* or *WEB-INF/lib*.

There is nothing to be aware of from the Application Designer side when running the application in an application server cluster.

When creating the web archive in this mode, the following is implicitly done:

- The file *cisconfig.xml* is adapted so that it does not use the Application Designer class loader and does not to start a monitoring thread of its own.
- All class files in the directory *<project>/appclasses/classes* are copied to *WEB-INF/classes*.
- All files in the directory *<project>/appclasses/lib* are copied to *WEB-INF/lib*.

Creating a Web Archive

The WAR file is written to the *bin* directory of your servlet container. If you are running a standard installation of Application Designer, the file is written to *<installdir>/tomcat/bin*.

See also: *Java API for WAR File Generation*.

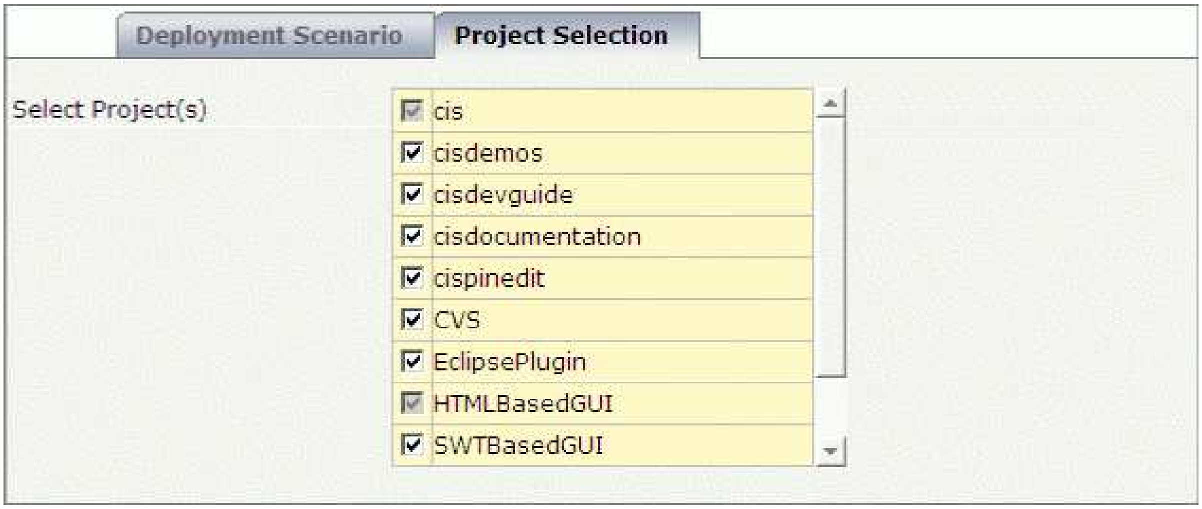
To create a web archive

1. Define the generation type by selecting one of the following option buttons: **with file system reference** or **fully clusterable**.
2. When you have selected the option button **with file system reference**, you have to specify the following:

Option	Description
CIS Root Location	The directory which contains your web applications. You can also select `REALPATH` from the drop-down list box. This variable contains the servlet engine's path to the web applications. It is filled during runtime. **Caution:** `REALPATH` is not supported by all servlet engines.
Log File Location	The directory which contains your log files. You can also select one of the following from the drop-down list box: ● **REALPATH/xxx** The log files are written to a subdirectory of the `REALPATH` (see the above description), where *xxx* stands for the name of the subdirectory. ● **TEMP** This variable contains the servlet engine's path to the temporary directory. It is filled during runtime. The `TEMP` variable is supported by all servlet engines.
Switch off Design Time	Optional. Select this check box if you do not want to include the development tools and Application Designer's documentation in your web archive.

Note:
The above options are disabled if you have selected the option button **fully clusterable**.

3. Select the **Project Selection** tab.



Select Project(s)

☑ cis
☑ cisdemos
☑ cisdevguide
☑ cisdocumentation
☑ cispinedit
☑ CVS
☑ EclipsePlugin
☑ HTMLBasedGUI
☑ SWTBasedGUI

4. Select the application projects that you want to include in your web archive.

The project "HTMLBasedGUI", which contains the Application Designer resources, cannot be deselected. The projects starting with "CIS" contain demos only; if you want, you can deselect these projects.

5. In the text box **WAR File to be created**, enter the name and extension for your web archive file.
6. Choose the **Create WAR** button.

   The web archive is created.

   After the creation, a protocol is available in the lower part of the WAR Packager dialog.

Java API for WAR File Generation

Application Designer provides a Java API for generating a WAR file. Call the method `main` of class `WARGenerator` (package `com.softwareag.cis.editor`) with the following argument list:

Argument	Description
1. Application Designer home directory	Mandatory. The directory which contains your web applications.
2. Temporary directory	Mandatory. A temporary directory.
3. Result file name	Mandatory. The name of the generated WAR file.
4. Clusterable indicator	Mandatory. A flag (`true` or `false`) which indicates whether the generated WAR file can be used within a server cluster.
5. Switch off design time	Mandatory. A flag (`true` or `false`) which indicates whether the generated WAR file contains the Application Designer design time resources (tools).
6. Project name	Optional. The name of the application project that is to be part of the WAR file. If you want to add multiple projects, add further names subsequently.

For more details, see the Java API documentation (JavaDoc).