

# **jWB - Deploying a Project From One jWB Server To Another**

## **INTRODUCTION**

It is common practice to develop on a completely separate environment from the one presented to your end users. In this instance, a mechanism is needed to move your completed project from one machine to another.

JWB provides a comprehensive toolkit for bundling your application into a package which may be installed on another server.

This document describes the process of building an export from one server and importing it into another server. An example application is used as the basis for this document, though the principle is the same for any jWB application.

The finished export definition, which comprises the Used Car Demonstration is zipped together with this document for reference.

NOTE – It is only possible to export and import between jWB servers which have the same major and point release numbers. It is possible to transfer packages from a 3.1.1 server to a 3.1.2 server, but not between a 3.1.1 server to a 3.2.1 server. This may be changed in a future release of jWB.

## ANATOMY OF A jWB APPLICATION

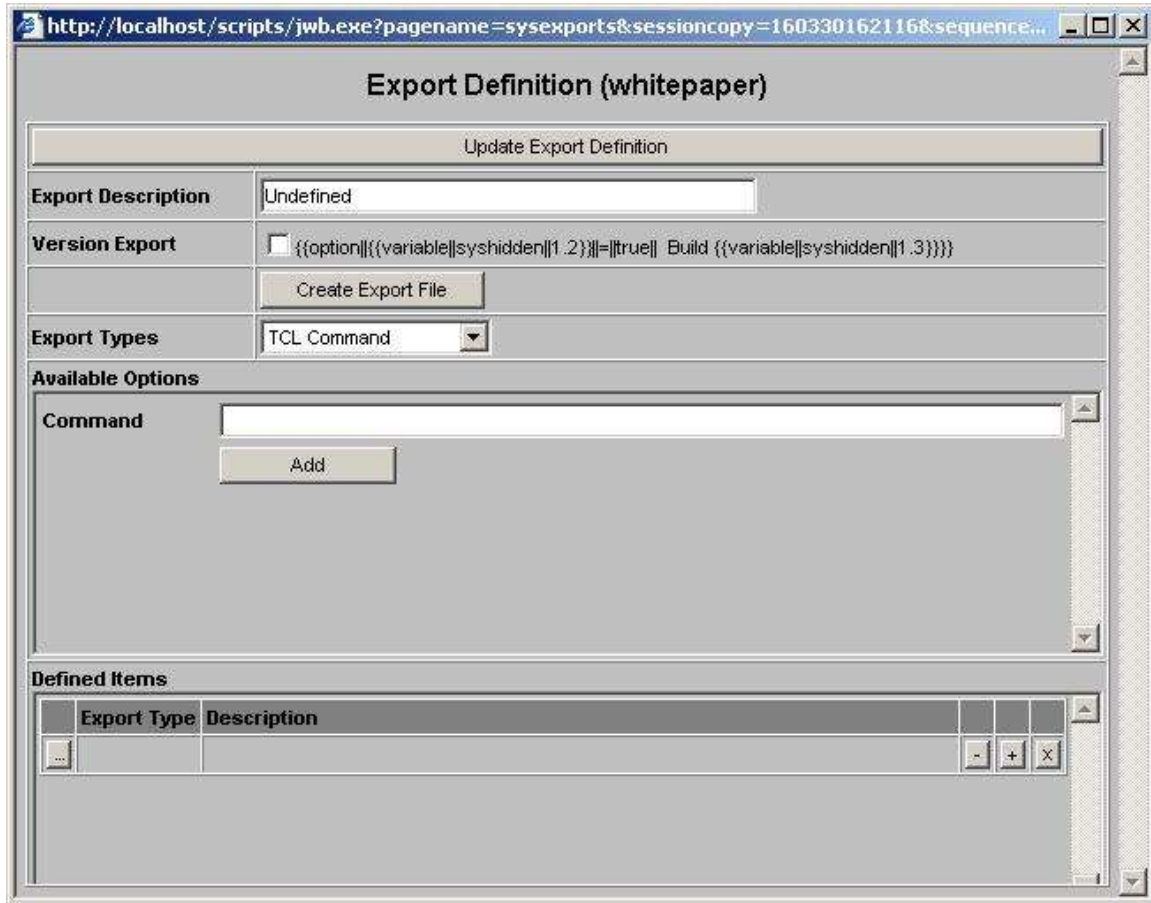
The used car sample application represents a typical small web builder project. This application is made up of the following objects;

Objects (All in same module)	Subroutines	Files
(Pages) pagcarddelete pagcardelres pagcarimage pagcarconfirmdel pageditcar pagfront paginsertcar pagupdatecar  (Tables) tabcarimage tabeditcar tabfront tabfrontbody tabupdatecar  (Scripts) alternatecolour header footer	(All in webBP) prepagconfirmdel prepageditcar prepagfront prepaginsert prepagupdate	(Data files) CAR CAR]D MARQUE MARQUE]D RESERVATION RESERVATION]D  (Picture files – stored in webroot/images) car.jpg car1.jpg car2.jpg car3.jpg car4.jpg car5.jpg car6.jpg

As can be seen from the table, the structure of a fairly simple application, can build up into a large list of components. In order to move this application to another server, each of these components will need to be packaged into the export file.

## APPLICATION EXPORTS

The application exports screen can be activated by choosing application exports from the main web builder menu. This brings up a list of all exports which have been defined on the server in the right hand pane. In order to add a new export, click on the large button at the top of the screen. Give the export a name, and click OK. The export definition screen should appear as below;



The screenshot shows a web browser window with the URL `http://localhost/scripts/jwb.exe?pagename=sysexports&sessioncopy=160330162116&sequence...`. The page title is "Export Definition (whitepaper)". The form contains the following elements:

- An "Update Export Definition" button at the top.
- An "Export Description" text input field containing "Undefined".
- A "Version Export" section with a checkbox and the text `{{option|{{variable|syshidden|1.2}}|=|true| Build {{variable|syshidden|1.3}}}}`.
- A "Create Export File" button.
- An "Export Types" dropdown menu set to "TCL Command".
- An "Available Options" section with a "Command" text input field and an "Add" button.
- A "Defined Items" table with columns "Export Type" and "Description". The table is currently empty.

An export description may be entered. This can be used to store more information about the export that is being created.

In order to make use of the build comparison functionality within web builder, the version export box should be ticked.

The export is made up of a list of linear instructions. As these instructions are added to the export, they will appear at the base of the screen in the defined items table.

The instructions themselves may be added by selecting the type of instruction from the export types list, entering or choosing any parameter information, and then clicking on the add button.

The different types and appropriate uses of instructions are listed in the following table;

TCL Command	Any command that could be entered at a jShell prompt may be added to the export.  For example, <b>BASIC webBP *</b>
Entire File Contents	This will copy the entire contents of an operating system file. This could be used to copy across hash files.
Program File Contents	This will copy across the entire contents of a file, and after copying, will execute a BASIC and CATALOG of each item in the file.
Application Module	This will copy across all web builder objects from a particular application module.
Specific Object	If one object in particular needs to be copied across, this option will do it for you.
Specific Record	This is used where you only want to copy only a single record from a hash file.
Specific Configuration Parameters (post v3.2.1)	This allows miscellaneous items to be transferred including the following:-  <ul style="list-style-type: none"> <li>Start Page</li> <li>Post Logon option</li> <li>Session Purge</li> <li>Connection Options</li> <li>Program Files</li> <li>Application Files</li> <li>Help Files</li> <li>Default Guest User</li> <li>Process Executes</li> <li>DIRT Tags</li> <li>Named Equivalentents</li> <li>Language Definitions</li> <li>Editor Defaults</li> <li>Default Preview Size</li> <li>Browser Navigation Blocking</li> <li>Cache Blocking</li> </ul>

## THE USED CAR EXPORT

To create the Used Car Export, the package needs to include three separate components.

The first task is to package up all the objects in the project. Because all of the objects are in the same application module, this can be accomplished by choosing the Application Module option from the Export Type list. The screen will refresh to show a list of all Application Modules set up on the server. Selecting the Used Car Demonstration module from the list, and clicking the Add button will add all the objects to the export.

The next task is to add all the BASIC code that accompanies the project. All of the code is contained within one file, webBP, on the server, so if it could be guaranteed that no other program files existed in that file, the Program File Contents option from the Export Type list could be used. In this particular example, the WebBP file might contain other programs, so each program needs to be treated individually.

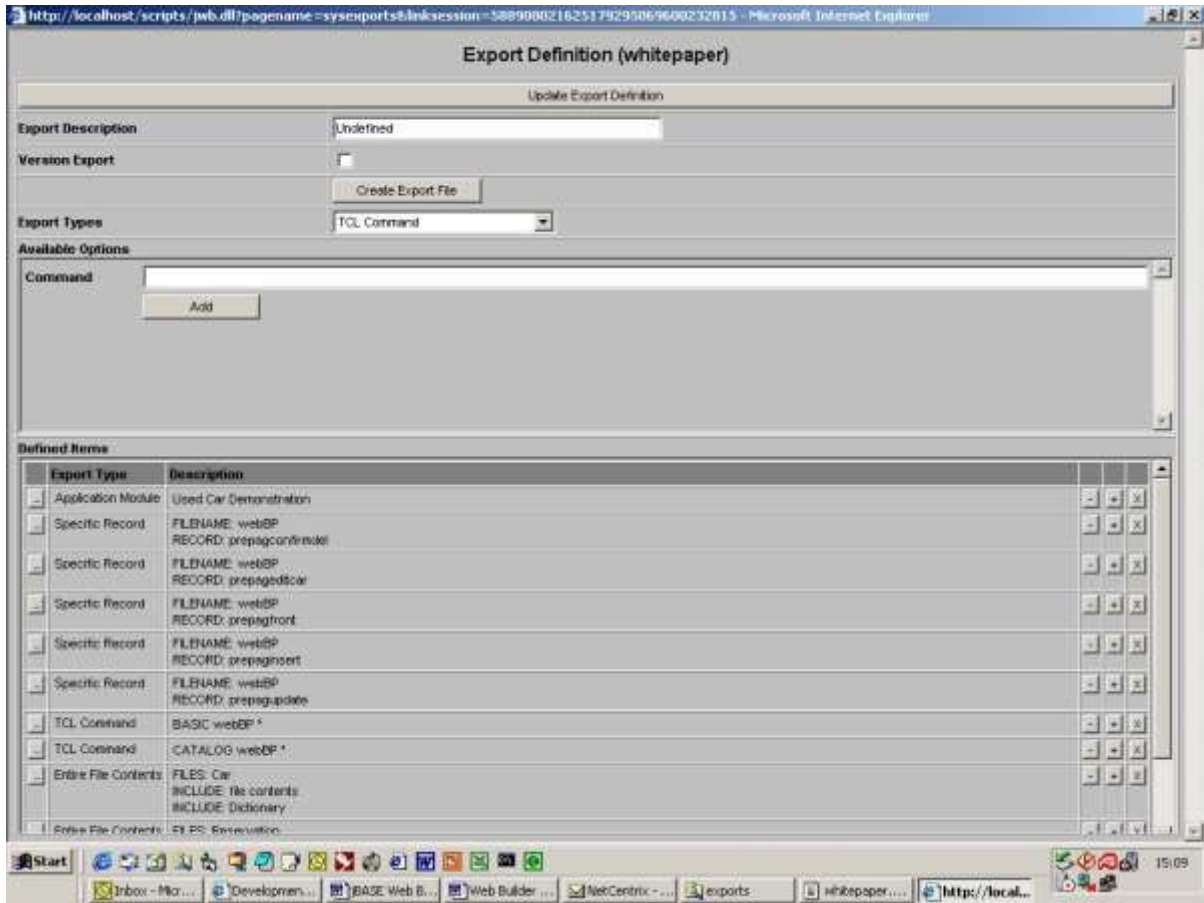
To add individual programs to the export, the Specific Record option from the Export Type list can be used. Each program file can be specified in turn by giving the filename webBP, and the record of the individual program name.

Once the individual programs have been copied across to the target server, they will need to be compiled and catalogued before they can be recognised by jWB. The TCL Command option from the Export Type list enables us to do this. By adding two entries, one with a command of **BASIC webBP \*** and the other with a command of **CATALOG webBP \***, all the programs in webBP on the target server will be compiled and catalogued.

If programs needed to be compiled and catalogued individually, a separate BASIC and CATALOG entry would be needed for each program eg. **BASIC webBP prepagconfirmdel** and **CATALOG webBP prepagconfirmdel**.

The data files may be copied across using the Entire File Contents option from the Export Type list. An entry is needed for each of the three data files. Ticking the file contents box will package the records contained in the file. Ticking the Dictionary box will bring the Dictionary file across.

Once all the instructions have been added to the export package, the screen should appear as follows;



## PICTURE FILES

This export file does not cater for the picture files that are part of the Used Car Demonstration. If the platform on the target machine is identical to the source machine, and the web server has exactly the same root directory, these files may be added to the export using the Specific Record option from the Export Type list.

Because of this restriction, it is recommended that separate files required by the application be kept apart from the export. This means that the export remains platform independent.

## **PRODUCING THE EXPORT**

Once the list of instructions is complete, the export can be built by clicking on the Create Export File button. Web Builder will prompt for a path to write the export file out to.

This path must exist on the server, and the filename must not already be there. So if **c:\temp\temp.exp** is specified, the directory **c:\temp** must exist, but the file **c:\temp\temp.exp** must not.

## **THE IMPORT PROCESS**

Once the export has been produced, it needs to be copied to somewhere accessible from the target machine (on to the local hard drive for example).

The file can then be imported by selecting the Application Imports option from the main web builder menu. A list of all previous imports will be displayed.

The file may be imported by clicking on the large Import File button at the top of the screen. Web builder will prompt for the path to the file. Once the valid path has been entered, web builder will import the file.

After the import, any separate files need to be copied to the correct directories on the target server, and then the application should be ready for testing.