

Oracle® Forms Developer

WebUtil User's Guide

Release 1.0.6

October 2005

Copyright © 2005

Primary Author: Duncan Mills, Grant Ronald, Orlando Cordero

Contributing Author: James Amalraj, Ganesan Prasanna, Ananth Satyanarayana, Opendro Singh

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Contents

1 Introduction to WebUtil

1.1	What Is WebUtil?	1-1
1.2	The Principles of the WebUtil Design.....	1-1
1.3	The Technology Behind WebUtil	1-2

2 Configuring WebUtil

2.1	Introduction	2-1
2.2	WebUtil Prerequisites	2-1
2.3	The WebUtil Components and Directory Structure	2-2
2.3.1	Design Time Components	2-2
2.3.2	Runtime Components	2-2
2.3.3	WebUtil Directory Structure	2-3
2.4	Configuring Your Database for WebUtil.....	2-3
2.5	Configuring Oracle HTTP Server for WebUtil	2-3
2.6	WebUtil Entries in the formsweb.cfg File.....	2-4
2.7	WebUtil Entries in the Environment File	2-5
2.8	The webutil.cfg File	2-6
2.9	The webutil.properties File.....	2-6
2.10	Additional WebUtil Installation Steps.....	2-6
2.11	Securing WebUtil	2-7
2.11.1	WebUtil Security Overview	2-7
2.11.2	Securing WebUtil within the Internet.....	2-7
2.11.3	Securing WebUtil within the Intranet.....	2-8

3 Configuring WebUtil

3.1	The formsweb.cfg File	3-1
3.1.1	Required Configuration in the formsweb.cfg File	3-1
3.1.2	Optional WebUtil Parameters in formsweb.cfg	3-1
3.2	The webutil.cfg File	3-3
3.2.1	Logging Options	3-3
3.2.2	Installation Options	3-3
3.2.3	File Upload and Download Options.....	3-5
3.2.3.1	Enabling Transfer	3-5
3.2.3.2	Access Control.....	3-5
3.2.3.3	Defining Read and Write Directives	3-5

3.2.3.4	The Work Area.....	3-6
---------	--------------------	-----

4 Functions in WebUtil

4.1	Client Server Parity APIs	4-1
4.1.1	Client_Text_IO Package.....	4-2
4.1.2	Running HOST Commands	4-2
4.2	Ported D2KWUTIL Functions	4-2
4.3	WebUtil Public Functions	4-3
4.3.1	WebUtil_ClientInfo	4-4
4.3.2	WebUtil_C_API.....	4-4
4.3.3	WebUtil_File	4-5
4.3.4	The File Transfer Package.....	4-6
4.3.5	WebUtil_Host.....	4-8
4.3.6	WebUtil_Session	4-9
4.3.7	WebUtil_Browser.....	4-10
4.3.8	WebUtil_SeparateFrame	4-10
4.3.9	WebUtil_Core.....	4-11
4.4	Utility Functions.....	4-11
4.4.1	DelimStr	4-12
4.4.2	Show_WebUtil_Information	4-12
4.4.3	WebUtil_Util.....	4-12
4.5	Internal APIs.....	4-12

5 Using WebUtil in Your Applications

5.1	Adding The Required Objects.....	5-1
5.2	Seeing the PJC at Design Time	5-2
5.3	Adding WebUtil Code.....	5-2
5.4	Using OLE Commands	5-2

6 Performance Tuning Considerations

6.1	Introduction	6-1
6.2	Optimizing Your Use of WebUtil	6-1

7 Tracing and Diagnostics

7.1	Connection Logging	7-1
7.2	Error Handling	7-1
7.3	Diagnostic Tracing	7-2

A Runtime Setup Checklist

B WebUtil Error Reference

Introduction to WebUtil

This manual documents the principals and usage of the WebUtil utility with Oracle Forms Applications. You should read and understand the sections on installing and configuring WebUtil before using the utility.

This chapter contains the following sections:

- [Section 1.1, "What Is WebUtil?"](#)
- [Section 1.2, "The Principles of the WebUtil Design"](#)
- [Section 1.3, "The Technology Behind WebUtil"](#)

1.1 What Is WebUtil?

WebUtil is a pre-packaged set of components that can be used to add a great deal of extra functionality to Web-deployed Forms applications. WebUtil addresses common challenges faced by Oracle Forms developers who wish to build applications which integrate tightly with the client browser - the computer at which the end user is actually located.

Traditionally, Oracle Forms has provided the means of integration with the computer that the Forms executable is running on. However, in the Web-deployed scenario, the delivery of the Forms application can be remote from end users and may even be on a different operating system from the systems that are being used to host the application's user interface. WebUtil allows Oracle Forms developers the means to interact with the client browser computers by using PL/SQL, and without having to learn any new technology. As such, WebUtil is designed for programmers who are migrating client server applications from Microsoft Windows desktops to the Web, but still need some integration between their Oracle Forms applications and external packages such as the Microsoft Office Suite running on the client browsers' computers.

1.2 The Principles of the WebUtil Design

The aim of WebUtil is to provide a utility that any Forms developer can use to carry out complex tasks on the client browser computer by simply coding in PL/SQL. Although WebUtil itself uses Java extensively, there is no need for the developer to have any understanding of Java. Everything the developer needs to do is exposed through a standard PL/SQL library webutil.pll.

All of the components that are needed to support WebUtil are likewise exposed through a simple object group.

For an Oracle Forms application to take advantage of WebUtil, you only need to attach the WebUtil library and subclass the Object Group. No setup or configuration in code is required as WebUtil is self-configuring and where necessary, self-installing.

Some features of WebUtil, such as client side OLE integration, require code to be installed onto each client computer. One of the key features of WebUtil is its ability to "self-install" in these cases. There is no need for an administrator to configure each and every computer because WebUtil detects which client-side pieces are missing and automatically installs them.

Another aim of WebUtil is to simplify porting of client server integration code to work on the client browser computer. To achieve this, WebUtil contains PL/SQL APIs mimic the behavior of the equivalent native Forms functions.

An example of this is the Forms TEXT_IO package. This package provides the developer with facilities to read and write text files on the application server. WebUtil provides an alternative implementation, CLIENT_TEXT_IO, which is identical to TEXT_IO except that calls are implemented on the client browser computer rather than on the middle tier. Migrating existing code to operate on the client can be achieved using a simple search and replace operation in the PL/SQL code to changes references to point at the alternative "CLIENT" implementations provided by WebUtil.

The final goal of WebUtil design is to add value to Oracle Forms as a product. WebUtil adds capabilities to Forms which have simply not existed before in either web or client server deployments.

1.3 The Technology Behind WebUtil

WebUtil uses the extensibility features of Oracle Forms, both in the Java client using the Pluggable Java Component (PJC) mechanism, and on the application server using the Forms interfaces to Java. Nothing in WebUtil is secret; it uses published APIs and documented techniques to provide all of its functionality. Of course one of the key benefits of the utility is that it then wraps up those complex activities inside a series of simple PL/SQL APIs.

WebUtil is designed to be extended, to add further functionality, as it is required. Much basic infrastructure, such as error diagnosis and self-installation, is provided as part of the WebUtil kernel, and customers can use these APIs to expand WebUtil with additional functionality as requirements dictate.

Configuring WebUtil

This chapter contains the following sections:

- [Section 2.1, "Introduction"](#)
- [Section 2.2, "WebUtil Prerequisites"](#)
- [Section 2.3, "The WebUtil Components and Directory Structure"](#)
- [Section 2.4, "Configuring Your Database for WebUtil"](#)
- [Section 2.5, "Configuring Oracle HTTP Server for WebUtil"](#)
- [Section 2.6, "WebUtil Entries in the formsweb.cfg File"](#)
- [Section 2.7, "WebUtil Entries in the Environment File"](#)
- [Section 2.8, "The webutil.cfg File"](#)

2.1 Introduction

If you've selected the **Complete** installation option for installing Oracle Developer Suite 10g (10.1.2), WebUtil is installed and configured for you.

If you've downloaded WebUtil to install into a previous version of Oracle Forms, or to upgrade an earlier version of WebUtil, read the *Webutil Release Notes* document that is part of this distribution. You can also read the *Webutil Release Notes* at <http://www.oracle.com/technology/products/forms/>.

Note: If you've downloaded WebUtil from Oracle Technology Network, Webutil 1.0.6 can be configured with OracleAS 10g (10.1.2) as well as OracleAS 10g (9.0.4). You'll need to extract the Zip file to ORACLE_HOME/forms for Oracle Application Server 10g (10.1.2) or ORACLE_HOME/forms90 for Oracle Application Server 10g (9.0.4).

[Appendix A, "Runtime Setup Checklist"](#) features a checklist that you can use to guide you through the installation and setup process.

2.2 WebUtil Prerequisites

In order to use WebUtil there are several version prerequisites:

1. Oracle Forms Version: WebUtil is designed to run with Oracle Forms 10g (9.0.4) and Oracle Forms 10g (10.1.2). No testing has been done with previous versions.

2. Client Java Virtual Machine (JVM): WebUtil requires either JInitiator 1.3.1.13 and higher, or the Sun Java Plug-in Version 1.4.2 or higher. WebUtil uses certain Java 1.3 features that are not available in earlier versions of the JVM.
3. UNIX and Linux directory permissions: WebUtil downloads install.syslib libraries mentioned in webutil.cfg into the bin directory of the JRE or JVM. The client needs write permission in that particular directory.

For more information about supported plug-ins, see the latest certification matrix, *Oracle Forms 10g: Client Platform Support Statement of Direction* at <http://www.oracle.com/technology/products/forms/>.

4. Oracle Application Server Java Virtual Machine – For its Java operations on the middle tier, WebUtil needs a JVM of version 1.2.2 or higher. The JVMs supplied with Oracle Developer Suite 10g (10.1.2), Oracle Developer Suite 10g (9.0.4), Oracle Application Server 10g (10.1.2), and Oracle Application Server 10g (9.0.4) are ideal.
5. Oracle Database Version – The file transfer facilities within WebUtil that post and get files from the database need to be installed into a 9.0.1 database or higher. Using WebUtil against an older database will mean that this type of file transfer is not available to you.

2.3 The WebUtil Components and Directory Structure

The following sections contain information about:

- [Section 2.3.1, "Design Time Components"](#)
- [Section 2.3.2, "Runtime Components"](#)
- [Section 2.3.3, "WebUtil Directory Structure"](#)

2.3.1 Design Time Components

For design time use WebUtil consists of just two modules: the webutil.pll PL/SQL library and the webutil.olb object library. The PLL contains all of the PL/SQL APIs required to call WebUtil and the OLB contains the definitions of the Bean Areas that the WebUtil subcomponents use. Both of these files exist in the `forms` subdirectory.

Also for design time use you may optionally include the `frmwebutil.jar` file in your `FORMS_BUILDER_CLASSPATH` environment variable/Registry entry. Adding this file to the `CLASSPATH` will prevent a warning message from appearing if you open the `WEBUTIL_CANVAS` in the layout editor and it will also allow you to view the version numbers of each of the WebUtil subcomponents. The JACOB libraries also need to be added to `FORMS_BUILDER_CLASSPATH`. See [Section 2.3.2, "Runtime Components"](#) for more information on obtaining the JACOB libraries.

2.3.2 Runtime Components

At runtime the primary components used for WebUtil are the `frmwebutil.jar` file and `webutil.pll`. Some features of WebUtil such as OLE integration also require extra operating system libraries, which will be downloaded to the client on demand such as the `jacob.dll` and `jacob.jar` files for OLE integration. These operating system-specific files are then stored in the `forms/webutil` and `forms/java` directories, respectively.

JACOB source code and licensing details can be obtained at <http://prdownloads.sourceforge.net/jacob-project/>.

Also, see the *WebUtil Release Notes* document that is part of the WebUtil distribution. You can also read the *WebUtil Release Notes* at <http://www.oracle.com/technology/products/forms/>. The *Webutil Release Notes* contains information about downloading and signing JACOB libraries.

2.3.3 WebUtil Directory Structure

Whether you are using WebUtil that is included with Oracle Developer Suite 10g (10.1.2) or you downloaded it from Oracle Technology Network and extracted it, the WebUtil directory structure has these folders:

- doc
- java
- server
- webutil

Webutil.pll, Webutil.olb and the create_webutil_db.sql exist in the Forms directory. When you extract the WebUtil Zip file, its contents are extracted into the ORACLE_HOME\forms or ORACLE_HOME\forms90 folder. All files will be copied to the respective directories in the ORACLE_HOME.

2.4 Configuring Your Database for WebUtil

Some of the functions that WebUtil provides for file transfer in the WEBUTIL_FILE_TRANSFER package require the database package WEBUTIL_DB. The script called create_webutil_db.sql is supplied in the forms directory to create this package. This script uses the following database packages, which must be available to the schema that is installing the WEBUTIL_DB package:

- DBMS_LOB
- UTL_ENCODE
- UTL_RAW

Oracle recommends that you create a special user "webutil" on your database as the owner of this package.

Once the WEBUTIL_DB package has been created and compiled on the database you will need to make it available to the relevant users of your application. Additionally, you will have to create a synonym of WEBUTIL_DB for this package if you are running from a schema other than the one that you installed it with. The Forms PL/SQL code refers to the package as WEBUTIL_DB without a schema prefix.

The WEBUTIL_DB package is executed with invokers' rights so that the package can refer to schema objects that the currently connected user can see, and the owner of the WEBUTIL_DB package itself does not have to be granted access to every possible table that contains a BLOB column to upload to.

2.5 Configuring Oracle HTTP Server for WebUtil

WebUtil needs a single virtual directory to be defined in order to download files at runtime as they are needed. You will need to create a virtual directory called j2ee/DevSuite/application-deployments/forms/formswb/webutil which maps onto the ORACLE_HOME/forms/webutil directory in the Oracle Developer Suite and Oracle Application Server ORACLE_HOME directories.

For Oracle Developer Suite, add the following line to the file
j2ee/DevSuite/application-deployments/forms/formsweb/
orion-web.xml under the Oracle Developer Suite Oracle Home:

```
<virtual-directory virtual-path="/webutil" real-path="$ORACLE_
HOME/forms/webutil" />
```

In Oracle Application Server, define the same virtual directory in the
forms/server/forms.conf file:

```
AliasMatch ^/forms/webutil/(.*) $ORACLE_HOME/forms/webutil/$1"
```

2.6 WebUtil Entries in the formsweb.cfg File

To use your applications with WebUtil you will need to create some specific settings in your formsweb.cfg file. These may be applied to a specific profile within the formsweb.cfg or may be defined for all profiles. A full list of valid WebUtil parameters is supplied in [Chapter 3, "Configuring WebUtil"](#).

1. Define the name and location of the of the WebUtil Jar file.

At runtime the Forms server must have access to the frmwebutil.jar. This should be present in the ORACLE_HOME/forms/java directory. You define the name and location of this jar file using the configuration parameter webUtilArchive that passes the name of the frmwebutil.jar to the Forms server:

```
webUtilArchive=frmwebutil.jar
```

2. Define the HTML templates to use with JInitiator and the Java Plug-in.

WebUtil uses its own HTML template files, which contain all of the required parameters that can be sent to WebUtil. There are three such templates, supplied in the forms/server directory. Any formsweb.cfg profile that uses WebUtil will need to use these templates rather than the supplied base.htm, basejini.htm and basejpi.htm.

Note: If you currently use a customized template to launch your applications, you can copy the WebUtil templates and customize them to your requirements.

Set the following parameters in your configuration to direct the servlet to use the WebUtil specific templates (amend the physical location as required, or copy the templates into your forms/server directory, in which case you will need no path, just the template name):

- baseHTMLjinitiator=webutiljini.htm
- baseHTMLjpi - Should point to the baseHTML file for the Java Plug-in e.g.
baseHTMLjpi=webutiljpi.htm.
- baseHTML - Should point to the baseHTML for WebUtil, e.g.
baseHTML=webutilbase.htm.

Note: For Oracle Application Server, on all platforms, you must regenerate webutil.pll before using it; otherwise you'll encounter error ORA-06508 when running a form with the attached library. To recompile, use the following command:

```
frmcmp module=ORACLE_HOME\forms\webutil.pll
userid=<webutil/webutil@dbconnect> module_type=library compile_
all=yes
```

The user ID can be any user with the required privileges as described in [Section 2.4, "Configuring Your Database for WebUtil"](#)

2.7 WebUtil Entries in the Environment File

In order to work, WebUtil requires several changes to your environment file (usually default.env stored in the /forms directory). You can also create a separate .env file especially for WebUtil applications.

1. The FORMS_PATH must include the directory where a copy of webutil.pll exists. You can either explicitly add the webutil/forms directory to this path, or you can of course copy the webutil.pll file to an existing directory that is mapped into the FORMS_PATH or the ORACLE_PATH.

Note: Ensure that the total length of your FORMS_PATH string does not exceed 255 characters. This can cause problems on Windows based installs. If necessary, use ORACLE_PATH as an additional variable to define extra long search paths.

2. The second requirement within the .env file is for an environment variable WEBUTIL_CONFIG. This variable must point at the name and location of your webutil.cfg file. By default this is in the ORACLE_HOME/forms/server directory e.g.

```
WEBUTIL_CONFIG=ORACLE_HOME\forms\server\webutil.cfg
```

3. The frmwebutil.jar (including its physical location) file must be included in the CLASSPATH, along with the Java runtime Jar rt.jar. e.g.

```
CLASSPATH=\ORACLE_
HOME\forms\java\frmwebutil.jar;c:\ids9i\jdk\jre\lib\rt.jar
```

Note: Some of the file transfer functions of WebUtil use Java integration on the application server. The steps concerning Java set-up in the .env file are not required if this functionality will not be used.

4. The PATH and library load path (on UNIX platforms) must be defined either in the general environment or specifically in the .env file to allow Forms to call Java. On a Win32 platform this will involve adding the ORACLE_HOME\jdk\jre\bin\client directory of a 1.3 or 1.4 Java install to the PATH. This enables Forms to find the jvm.dll file. A suitable Java home will already exist under the Developer Suite or Oracle Application Server ORACLE_HOME directory structure.

On UNIX platforms, Forms needs to locate the libjava.so file so the LD_LIBRARY_PATH on most UNIX platforms including Linux, will need to contain

the directory that holds this file. For instance on Linux this would be `$JAVA_HOME/jre/lib/i386/native_threads`.

Again the Java runtime install that is part of Oracle Application Server or Oracle Developer Suite is ideal

2.8 The webutil.cfg File

The webutil.cfg file is the master configuration file for WebUtil. It allows the administrator to control features such as logging and file transfer permissions. The basic webutil.cfg supplied is sufficient for initial needs but if you want to enable file transfer functionality or logging then you will have to edit this file. The settings within this file are discussed in detail in the next chapter.

Note: As a security precaution, the default settings in the webutil.cfg disable the WebUtil file transfer features. To use file transfer to or from the database or the application server you must explicitly enable these features and define the access control parameters as required.

2.9 The webutil.properties File

When libraries are downloaded, WebUtil creates the webutil.properties file which is located in the Java home directory. If this file already exists, an entry is made for each downloaded library, and WebUtil will not download the same library again.

2.10 Additional WebUtil Installation Steps

Use this list along with the installation checklist in [Appendix A, "Runtime Setup Checklist"](#) of the *WebUtil User's Guide* or Online Help to sign the WebUtil and JACOB Jar files:

1. Download http://prdownloads.sourceforge.net/jacob-project/jacob_18.zip. This archive supplies the core OLE functionality provided by WebUtil.
2. From the JACOB Zip file, extract both `jacob.dll` and `jacob.jar` into the `ORACLE_HOME\forms\WebUtil` and `ORACLE_HOME\forms\java` directories, respectively
3. You need to sign both `frmwebutil.jar` and `jacob.jar` with the same digital certificate. This is a one-time operation which allows your end-users to trust that the JACOB routines can access client side resources. If you do not have an existing signing certificate, or if you are not sure how to sign Jar files, a script is in the `forms\WebUtil` directory to help you. This script is called `sign_webutil.sh` for UNIX and `sign_webutil.bat` for Windows.

To sign the Jar files:

4. Open a Command window and change to the `ORACLE_HOME\forms\webutil` directory.
5. Check that `ORACLE_HOME/jdk/bin` is in the path. If it is not, add it.
6. In Windows, Issue `sign_webutil.bat ORACLE_HOME\forms\java\frmwebutil.jar` (or the path to where you installed WebUtil). On UNIX, issue `sign_webutil.sh ORACLE_`

HOME/forms/java/frmwebutil.jar (or the path to where you installed WebUtil).

7. In Windows, Issue `sign_webutil.bat ORACLE_HOME\forms\webutil\jacob.jar`. In Unix, issue `sign_webutil.sh $ORACLE_HOME>\forms\java\jacob.jar`.

Note: You only need to do this once to create the certificate that will be installed in the JInitiator Keystore.

8. Because in this release the JACOB code is in an external Jar file and not incorporated into `frmwebutil.jar`, it needs to be downloaded. To do this, change the `WebUtilArchive` setting to read:

```
WebUtilArchive=frmwebutil.jar,jacob.jar
```

2.11 Securing WebUtil

This section contains the following:

- [Section 2.11.1, "WebUtil Security Overview"](#)
- [Section 2.11.2, "Securing WebUtil within the Internet"](#)
- [Section 2.11.3, "Securing WebUtil within the Intranet"](#)

2.11.1 WebUtil Security Overview

Securing WebUtil is just as important as you would secure any other JavaBean. In addition, the client integration features that comprise WebUtil are all features that allow an Oracle Forms application to communicate with the user's client computer. `frmwebutil.jar`, like many Jar files that require specific permissions, need to be signed to work properly. Oracle Corporation encourages the use of certificates issued by a trust authority that is internal or external to your company, and not to use any type of self-signed certificates.

One potential security issue is when a user uses a stolen and signed `frmwebutil.jar` file to access user client computers without their knowledge.

2.11.2 Securing WebUtil within the Internet

When the `frmwebutil.jar` is signed and used, the signed version is downloaded to the Jar cache folder on the client computer. Previously, this signed version could also be used by unauthorized domain servers. As a solution, you can add a list of server domains to the `frmwebutil.jar` file before signing it.

`frmwebutil.jar` has a new text file named `TrustedDomains.txt` and is located in `ORACLE_HOME/forms/webutil/common`. This file contains a list of trusted server domains where each domain is separated by a new line.

WebUtil checks whether or not `frmwebutil.jar` was downloaded from one of the listed allowed servers.

The following is a sample `TrustedDomains.txt` file:

```
/ TrustedDomains.txt
/
```

```
/ Copyright (c) 2004, Oracle. All Rights Reserved.
```

```
/
/  NAME
/    TrustedDomains.txt - List of Trusted Domains
/
/  DESCRIPTION
/    This text file contains the list of domains that can be trusted
/    for downloading Forms WebUtil. Each line can have only one domain
/    and/or each domain string can have * wild-character to match multiple
/    domains. IP addresses can also be used, but not preferred since we
/    may not be able to get the right IP through proxy. Localhost is
/    always trusted.
/
/  NOTES
/    Please refer documentation for more information about securing WebUtil
/    jar. Also read about WebUtilTrustInternal applet parameter.
/
*.oracle.com
foo.net
foo.com
*.foo.org
```

The first and last lines allow any sub-domain within oracle.com and foo.org to sign and download frmwebutil.jar. The user can type the IP address in a browser URL while the domain filters are host names or vice versa as long as the host name and IP address are resolvable.

A mismatch between the listed server domain in `TrustedDomains.txt` and the one that `frmwebutil.jar` was downloaded from throws an exception. Oracle Forms will terminate immediately so that it is not possible to continue with the application because of a security violation.

If no `TrustedDomains.txt` file is found, for backwards compatibility reasons, WebUtil assumes that the user decided to not protect `frmwebutil.jar`. Oracle Corporation recommends all users to run `frmwebutil.jar` with security enabled.

2.11.3 Securing WebUtil within the Intranet

Within the intranet (LAN/WAN), it is possible to use host names without the domain suffixes, e.g. `http://forms-pc:8888/`. In this case, a domain filter like `*.oracle.com` will not match. To trust computers within an intranet, there is an applet parameter called `WebUtilTrustInternal`. This parameter, when set to `TRUE` (default), indicates that all the intranet computers will be trusted. When it is set to `FALSE`, users will always need to type the domain suffixes to match the domain filter in `TrustedDomains.txt`.

For information about setting `WebUtilTrustInternal`, see [Section 3.1.2, "Optional WebUtil Parameters in formsweb.cfg"](#).

Configuring WebUtil

This chapter contains the following:

- [Section 3.1, "The formsweb.cfg File"](#)
- [Section 3.2, "The webutil.cfg File"](#)
- [Section 3.2.3, "File Upload and Download Options"](#)
- [Section 3.2.3.4, "The Work Area"](#)

3.1 The formsweb.cfg File

You're already using the formsweb.cfg file to configure your Oracle Forms applications. WebUtil has some requirements for existing settings within the configuration that you are defining, and some optional new parameters, which can be added to the file, to customize certain aspects of WebUtil behavior.

3.1.1 Required Configuration in the formsweb.cfg File

- webUtilArchive - Location and name of the frmwebutil.jar file. If you have followed the installation instructions this parameter should look like:
`webUtilArchive=frmwebutil.jar,jacob.jar`
- EnvFile - If you use a named environment file or the default.env ensure that the FORMS_PATH that you have set includes the ORACLE_HOME\forms directory and that you have set the WEBUTIL_CONFIG variable to define the physical location of your webutil.cfg file, ORACLE_HOME\forms\server\webutil.cfg.
- baseHTMLJinitiator - Should point at the WebUtil JInitiator template file, for example, `baseHTMLJinitiator=webutiljini.htm`.
- baseHTMLjpi - Should point to the baseHTML file for the Java Plug-in, for example `baseHTMLjpi=webutiljpi`. baseHTMLjpi is used when JInitiator cannot be used, such as with non-Windows platforms.
- baseHTML - Should point to the baseHTML for WebUtil, e.g.
`baseHTML=webutilbase.htm`. baseHTML is for running WebUtil using Internet Explorer's native JVM.

3.1.2 Optional WebUtil Parameters in formsweb.cfg

The following parameters can be set to configure WebUtil behavior but are not required. They can all be passed on the browser URL as well to change the behavior without changing the configuration entry.

- **WebUtilLogging** - This parameter can be used to switch on logging in WebUtil. It can take one of the following values (case is not important), as shown in [Table 3–1, "WebUtilLogging Parameter Values and Descriptions"](#):

Table 3–1 WebUtilLogging Parameter Values and Descriptions

Value	Description
Off	WebUtil does no logging. This is the default value for this parameter
Console	Log messages are written to the Java console on the browser's computer
Server	Log messages are written to the WebUtil log file on the server
All	Log Messages are written to both the Java Console and the Server log file

- **WebUtilLoggingDetail** - Defines the level of logging used if logging has been enabled. There are only two levels of detail: **Normal** and **Detailed**. Normal is the default setting.
- **WebUtilErrorMode** - Defines how errors are displayed. Errors will always be displayed on the Java console, additionally you can specify one of the following for the WebUtilErrorMode parameter (case is not important), as shown in [Table 3–2, "WebUtilErrorMode Parameter Values and Descriptions"](#):

Table 3–2 WebUtilErrorMode Parameter Values and Descriptions

Value	Description
Console	Error messages are written to the Java console on the browser's computer. This is the default value for this parameter (there is no way to completely switch errors off)
Server	Error messages are written to the WebUtil log file on the server
Alert	Error messages pop up in a dialog as well as the console - This is a good mode to use during testing
All	Error messages appear in an alert and are written to the console and the Server log file

- **WebUtilDispatchMonitorInterval** - Used with the `WebUtil_Session` package to control how often the monitor thread checks to see if the Forms session is still alive. The value is measured in seconds and the default value is 5 seconds.
- **WebUtilMaxTransferSize** - When transferring files using WebUtil the file is transferred in segments, the size of which defaults to 16k. You can change the size of the transfer segments using this parameter to define the number of bytes required. Note that 16k is the maximum value accepted.
- **WebUtilTrustInternal** - This parameter allows clients within an Intranet to download a signed `frmwebutil.jar`, e.g. `http://forms-pc:8888/`. When this parameter is set to TRUE (default), the `TrustedDomains.txt` file is not used, so a domain filter like `*.oracle.com` will not match.

3.2 The webutil.cfg File

webutil.cfg is the key configuration file that controls how WebUtil works. This file is structured as a conventional Java properties file and is divided into three sections:

- [Section 3.2.1, "Logging Options"](#)
- [Section 3.2.2, "Installation Options"](#)
- [Section 3.2.3, "File Upload and Download Options"](#)

3.2.1 Logging Options

Because logging from WebUtil can be requested in the URL, webutil.cfg provides a way to ultimately control logging through the utility. This prevents "Denial of Service" type attacks attempting to disrupt the server by swamping the logging mechanism. By default all logging is disabled. [Table 3–3, "Logging Parameter Values and Descriptions"](#) shows the parameters that exist in webutil.cfg to control logging:

Table 3–3 Logging Parameter Values and Descriptions

Parameter	Description
<code>logging.file</code>	Defines the name and location of the log file to use. The file should be writable. WebUtil does no automatic log file maintenance and the log will continue to grow until manually deleted or truncated.
<code>logging.enabled</code>	Master switch that tells WebUtil to allow logging. Can be set to TRUE or FALSE
<code>logging.errorsonly</code>	Tells WebUtil to only log error messages not diagnostic messages. Can be set to TRUE or FALSE (Note: Errors will only be logged if the WebUtilError mode in the formsweb.cfg is set to SERVER or ALL to send the errors to the server in the first place)
<code>logging.connections</code>	This switch is separate from the main <code>logging.enabled</code> switch, and instructs WebUtil to keep track of all Forms sessions that use WebUtil. This log record includes the connecting computer IP address, the operating system information of the client and the running module.

3.2.2 Installation Options

One of the key features of WebUtil is its ability to automatically install any client-side libraries that it requires. This feature is used by WebUtil internally for its own requirements, but can also be used by the WebUtil administrator to install application-specific libraries as well.

The key setting in this section is `install.syslib.location`, which points to the virtual directory that contains the various WebUtil libraries which you've defined as `forms/webutil`. This path needs to either be an absolute URL of the computer or the virtual directory mapped to the `forms/webutil` directory (e.g. `http://apps-1/forms/webutil`), or a virtual directory relative to the documentbase `<server>/forms/` (e.g. `/webutil`).

Libraries to be downloaded are then listed in the entries with the Format `syslib.<os>.<package>.<n>` where `<n>` is a code as shown in [Table 3–4, "Operating System Type Installation Codes"](#):

Table 3–4 Operating System Type Installation Codes

Code	Operating System
0	Windows 32 bit (XP, Win2000 etc.)
1	Linux
2	Solaris
3	HP
4	AIX
5	Mac

- `<os>` is a number representing the client operating system family that this library is destined for.
- `<package>` is an identifier representing the WebUtil component that uses this library. To add your own library to the download list you should use "user" for this value.
- `<n>` a sequence number starting at 1 and incrementing by 1 – there must be no gap in this sequence for a particular platform and package combination.

The library definition string itself consists of four segments delimited by a vertical bar. The format is `name|size|version|showDownloadDialog`:

- **Name** – the name of the library file including its extension e.g. `jacob.dll`, `cfunc.so` etc.
- **Size** – the size in bytes of the library. This is used to checksum the download and must be correct
- **Version** – a version string, which WebUtil checks to see if it has already downloaded this library. If this version number is changed the new library will be automatically downloaded. (This is simply based on a string comparison of the version number not a numerical comparison of the version level)
- **ShowDownloadDialog** – While auto-installing, WebUtil can be asked to display a progress bar `TRUE` switches this on, `FALSE` suppresses it.

As an example for an application that uses the default WebUtil settings and has an additional two custom Windows DLLs to download this section might look like:

```
install.syslib.location=/webutil
## Change 2nd value to correct file size to avoid raising an error
install.syslib.0.7.1=jacob.dll|94208|1.0|true
install.syslib.0.9.1=JNIsharedstubs.dll|65582|1.0|true
install.syslib.0.9.2=d2kwut60.dll|192512|1.0|true
#Application custom libraries
install.syslib.0.user.1=scanner.dll|23056|1.0|true
install.syslib.0.user.2=mortgagecalc.dll|100230|1.0|true
```

3.2.3 File Upload and Download Options

This section contains the following:

- [Section 3.2.3.1, "Enabling Transfer"](#)
- [Section 3.2.3.2, "Access Control"](#)
- [Section 3.2.3.3, "Defining Read and Write Directives"](#)
- [Section 3.2.3.4, "The Work Area"](#)

For logging you may want to control access to the file transfer functions offered by WebUtil. WebUtil provides switches to enable file transfer to and from the database and to and from the application server disk. The administrator can define lists of directory roots on the application server that are readable and writable. Applications can only read and write from these directories and an error will be raised at runtime if access elsewhere is requested.

3.2.3.1 Enabling Transfer

Transfer is controlled by two master switches:

- `transfer.database.enabled=TRUE | FALSE`
- `transfer.appsrv.enabled=TRUE | FALSE`

These control access to the database and application server respectively. Additionally transfer to the database will only work if you are running against an Oracle 9.0.1 Database Server or higher.

3.2.3.2 Access Control

This section contains the following:

- [Section 3.2.3.3, "Defining Read and Write Directives"](#)
- [Section 3.2.3.4, "The Work Area"](#)

3.2.3.3 Defining Read and Write Directives

In the case of file transfer to and from the database, the database package that is used (WEBUTIL_DB) runs with Invokers rights. This means that the code can only upload to database records when the running Form can see that database schema.

For access to the application server disk, there is an additional switch `transfer.appsrv.accessControl` to define if the application can read from anywhere on the application server or from a controlled set of directories. When set to TRUE, WebUtil uses the directories listed in the `transfer.appsrv.read.<n>` entries and the `transfer.appsrv.write.<n>` entries. In both cases the sequence of values must start with 1 (e.g. `transfer.appsrv.read.1=d:emp`) and be contiguous.

Access control is based on simple string matching, so you should make your specifications as precise as possible, e.g. an entry of `"transfer.appsrv.read.1=d:\"` would allow WebUtil to read any file on the D disk in any subdirectory. An entry of `d:\temp` will restrict it to any file in the `d:\temp` directory or any subdirectory.

You can define up to 50 read and write locations.

For example, the directives required to allow WebUtil applications to read any directory on the server D drive but only write to the `d:\work\upload` and `d:\work\log` directories would be:

- `transfer.appsrv.enabled=true`
- `transfer.appsrv.accessControl=true`
- `transfer.appsrv.read.1=d:\`
- `transfer.appsrv.write.1=d:\work\upload`
- `transfer.appsrv.write.2=d:\work\log`

3.2.3.4 The Work Area

Administrators can also define a work area root using the parameter `transfer.appsrv.workAreaRoot`. The work area is a temporary directory on the application server where it is safe for applications to read and write. The work area must be set to a directory where the Oracle Forms Runtime Process has write permission. If it is unset, WebUtil assumes the work area to be the home/temp directory of the user who started the Oracle Forms server.

WebUtil provides an API (`WebUtil_File_Transfer.GetWorkArea()`) which will return a private directory for each connected user under this root work area. This private directory allows multiple users to upload files with the same name to the application server without overwriting files belonging to other users. Each user has a private directory under the work area which WebUtil automatically creates when required.

The `Client_Image` package used for client side `Read_Image_File` and `Write_Image_File` need the work area root to be defined.

The work area is always assumed to be readable and writable and need not be explicitly listed in the `transfer.appsrv.read` and `write` lists.

Functions in WebUtil

This chapter contains the following sections:

- [Section 4.1, "Client Server Parity APIs"](#)
- [Section 4.2, "Ported D2KWUTIL Functions"](#)
- [Section 4.3, "WebUtil Public Functions"](#)
- [Section 4.4, "Utility Functions"](#)
- [Section 4.5, "Internal APIs"](#)

4.1 Client Server Parity APIs

One of the key functions of WebUtil is to provide a way of re-directing calls commonly used in client server to operate on the Client Browser computer rather than on the application server computer. These functions and packages are all prefixed with "Client_", and make for simple conversion of client server code to operate on the client browser computer with a simple search and replace operation:

- Client_Get_File_name
- Client_Host
- Client_Image (Includes Read_Image_File and Write_Image_File)

Note: The Client_Image package contains implementations of the Read and Write image file Built-ins which work between a Forms image item and the client browser computer hard drive. In order for this feature to work you must enable File transfer to the application server and set up an appropriate work area – see [Chapter 3.2.3.1, "Enabling Transfer"](#).

- Client_OLE2 (Win32 Clients only)
- Client_Text_IO
- Client_Tool_Env

Each of these packages or program units is designed to work as closely to the basic version of the code as possible. This includes raising the same exceptions and errors. For documentation purposes treat these program units just like their Built-in equivalents. Note however, that there may be certain differences in implementation and you should test your code thoroughly to ensure that they behave as expected.

Many of the client server parity features in WebUtil are implemented using the main WebUtil public functions that often will offer more power and flexibility than the "Client_" version.

4.1.1 Client_Text_IO Package

One function in the Client_Text_IO package is Client_Text_IO.fopen. This function allows users to specify character set or file encoding. The character set name can be Oracle names like UTF8, JA16SJIS, AR8MSWIN1256, etc. or Java encoding names like UTF-8, MS932, WINDOWS-1256.

If the Oracle character set name does not have an equivalent in Java or if the encoding is unknown to the JVM, then it will use the client system.file default encoding. If the user does not also specify any encoding, then the client system's default file encoding is used.

Usage for CLIENT_TEXT_IO.FOPEN is:

```
FUNCTION fopen(spec VARCHAR2, filemode VARCHAR2, charSet VARCHAR2 := null)
RETURN file_type;
```

When a file is opened in append mode, there is a limitation that read is not allowed. Append means only writing at the end of the file. It is slightly different from the TEXT_IO append operation where read is allowed.

4.1.2 Running HOST Commands

If you have a host command of the form:

```
HOST ('DEL c:\temp.txt')
```

This will work in both client/server and Web deployed environments, but in web deployed environments the command runs on the middle tier.

If you change this to:

```
CLIENT_HOST('DEL c:\temp.txt')
```

This does not work. You must use:

```
CLIENT_HOST('cmd /c DEL c:\temp.txt');
```

You must ensure that the command will run from the **Start | Run** dialog on Windows.

4.2 Ported D2KWUTIL Functions

D2KWUTIL is another add-on package used extensively by client-server applications running on Windows. WebUtil contains client side ports of a limited number of D2KWUTIL functions, specifically those involved with environment information on the client computer, such as the functions to read and write the Windows Registry.

Only the Win_API_Environment package has been ported along with some dependent packages.

The APIs remain identical to the basic D2KWUTIL APIs, but again the prefix of "Client_" needs to be used. Errors in Client D2KWUTIL code can be referenced using the Client_Win_API package, just like in the client server version of the utility.

The following Win_API_Environment functions are available when running the browser on a Win32 computer only:

- Create_Registry_Key
- Delete_Registry_Key
- Get_Computer_Name
- Get_Environment_String
- Get_Net_Connection
- Get_Temp_Directory
- Get_Windows_Directory
- Get_Windows_Username
- Get_working_Directory
- Read_INI_File
- Read_Registry
- Write_INI_file
- Write_Registry
- Write_RegistryEx

Note: These functions are a direct port of the D2KWUTIL functions and are implemented using the WebUtil_C_API package to call into the D2KWUTIL DLL. The same information or function may be available in a more efficient manner elsewhere in WebUtil, e.g. WebUtil_ClientInfo.

4.3 WebUtil Public Functions

The core of WebUtil is a set of packages prefixed with "WebUtil_". These packages provide a set of APIs that contain:

- Client server Parity APIs
- Ported D2KWUTIL functions
- WebUtil Public Functions
- Utility Functions
- Internal APIs
- Information about the client computer
- A low level API to call C functions on the Client (Win32 clients only), this can be used to interface with the Windows APIs and other 3rd party code.
- File manipulation functions for files and directories on the client
- File transfer services to the client, database and application server file system
- Ability to execute operating system commands on the client
- Recovery functionality from Forms session time-outs and other failures

4.3.1 WebUtil_ClientInfo

Table 4–1, "Available Functions in the WebUtil_ClientInfo Package" describes the WebUtil_ClientInfo package functions that are used to obtain information about the client computer:

Table 4–1 Available Functions in the WebUtil_ClientInfo Package

Function	Purpose
Get_Date_Time	Returns a PL/SQL date containing the date and time on the client computer.
Get_file_Separator	Returns the character used on the client computer as the file separator e.g. "\" on Windows.
Get_Host_Name	Gets the name of the client computer.
Get_IP_Address	Gets the IP address of the client computer as a string.
Get_Java_Version	Returns the version of the JVM that is running the Forms Applet. This may be useful if you have your own JavaBeans in the form that depend on a certain Java version.
Get_Language	Returns the language code of the client e.g. en for English, de for German.
Get_Operating_System	Returns the O/S name of the O/S that the browser is running under.
Get_Path_Separator	Returns the character used to separate directory locations on Paths e.g. ";" on Windows.
Get_System_Property	Allows you to get any Java System Property.
Get_Time_Zone	Returns the time zone that the client computer is running in.
Get_User_Name	Returns the name of the user logged into the client computer.

4.3.2 WebUtil_C_API

The WebUtil_C_API is a comprehensive API which allows you to call into C libraries on the client. The API only supports Win32 Clients.

You must register your function that you wish to call. This call requires the name of the library file (e.g. DLL or .EXE) that contains the function and the name of the function itself. Once declared, the reference to the function is held in a `WebUtil_C_API.FunctionHandle`.

If the function takes parameters then you will need to declare a parameter list (of type `WebUtil_C_API.ParameterList`) to which you add each parameter to the function in turn, providing an indicator of the data type, a value if required, a maximum size (for `char*` variables) and an indicator saying if this parameter is IN, OUT, or IN OUT.

Functions are invoked by passing the handle of the registered function and the handle of the parameter list that defines its parameters.

4.3.3 WebUtil_File

WebUtil_File contains APIs to manipulate files and directories on the client and to display file selection dialogs. The package contains a new type FILE_LIST, which is a PL/SQL table (array) that is used to return multiple file names. This array can be accessed using standard PL/SQL table techniques, e.g. use <file_list>.COUNT to return the number of elements and <file_list>(n) to retrieve a specific numbered element. [Table 4-2, " Available Functions in the WebUtil_File Package"](#) describes these functions:

Table 4-2 Available Functions in the WebUtil_File Package

Function	Purpose
Copy_File	Copies a file. Returns a Boolean to indicate success.
Delete_File	Deletes a file. Returns a Boolean to indicate success.
Rename_File	Renames a file. Returns a Boolean to indicate success.
Create_Directory	Creates the named directory if it does not exist. Any intermediate directories will also be created. Returns a Boolean to indicate success.
Directory_Root_List	Returns a FILE_LIST containing the directory roots on the client system. On a Windows PC these would correspond to the drives on the computer.
Directory_List	Returns a FILE_LIST containing all the files and subdirectories in a particular directory. There is an optional Return_Files_Only argument to restrict the returned list to just files with no directories.
Directory_Filtered_List	Like Directory_List this will return a list of files in a directory but you are able to file filter using the '*' and '?' wildcards e.g. *.FMB to restrict the list.
File_Exists	Returns a Boolean value indicating if the named file exists on the client
File_Is_Directory	Returns a Boolean value indicating of the file name supplied is actually a directory on the client.
File_Is_Hidden	Returns a Boolean value indicating of the file has it's hidden attribute set.
File_Is_Readable	Returns a Boolean value indicating of the file can be read.
File_Is_Writable	Returns a Boolean value indicating of the file can be written to.
File_Size	Returns the size of the file in bytes.
Directory_Selection_Dialog	Displays a Directory selection dialog. The initial directory and the title for the dialog can be defined. The selected directory is returned.

Table 4–2 (Cont.) Available Functions in the WebUtil_File Package

Function	Purpose
File_Selection_Dialog	Allows the definition of a File Save or Open dialog with a configurable file filter. Returns a single file selection. The file filter can use ? and * as wildcards and is in the format <label> <filterString> . Multiple filters can be specified, e.g. Gif Files *.gif JPEG Files *.jpg .
File_Multi_Selection_Dialog	As File_Selection_Dialog except that the user can select multiple files and these are returned in a FILE_LIST.
File_Open_Dialog	A convenience method that creates a File Open Dialog.
File_Save_Dialog	A convenience method that creates a File Save Dialog.
Get_file_Separator	Returns the character used on the client computer as the file separator e.g. "\" on Windows.
Get_Path_Separator	Returns the character used to separate directory locations on Paths e.g. ";" on Windows.

4.3.4 The File Transfer Package

The WebUtil_File_Transfer package provides APIs for transferring files to and from the client browser computer. All of the transfer functions provide the ability to display a progress bar as the transfer takes place, to re-assure the end user if the connection is slow or the transfer is large. These functions are described in [Table 4–3, "Available Functions in the WebUtil_File_Transfer Package"](#).

Transferring data directly to a BLOB column in the database requires that you are running Oracle Database Server 9.0.1 or above and that you have installed the WEBUTIL_DB package and granted execute privilege to the current user. You will also need a public synonym of WEBUTIL_DB for the package.

Transferring data to or from the application server disk involves using some Java on the middle tier and so your .env file should be suitably configured with the correct Classpath information and Path / Library load path information to call Java from Forms PL/SQL.

Table 4–3 Available Functions in the WebUtil_File_Transfer Package

Function	Purpose
URL_To_Client	Transfers a file from a specified URL to the client computer. The file size can be specified as a checksum to see that the transfer succeeded. If it is not supplied the check will not be carried out. The function returns true if the download succeeded.
URL_To_Client_With_Progress	Displays a progress bar as the download happens. You can supply a title, subtitle and message to display in the dialog as the download takes place.

Table 4–3 (Cont.) Available Functions in the WebUtil_File_Transfer Package

Function	Purpose
Client_To_DB	<p>Uploads a named file on the client into a BLOB column in the database. You must define the name of the target table, the name of the column, and a where clause that defines one and only one row in that table. This upload is carried out using the current database connection and schema privileges. The process can also be made asynchronous, in which case you can define the name of a trigger which WebUtil will execute as soon as the upload is complete.</p> <p>Note: Asynchronous mode offers better re-drawing of the Forms screen while the transfer takes place, but will not offer the ability to carry out significant work while the transfer is taking place. The webutil.cfg parameter transfer.database.enabled must be set to true to use this function.</p>
Client_To_DB_With_Progress	As Client_To_DB but with a progress bar, which again can have various labels defined for it.
DB_To_Client	<p>Downloads a file from a BLOB column in the database to the client computer. The table name, column name and a where clause all have to be specified to identify the binary object to download. The current forms session and schema is used for the download. The webutil.cfg parameter transfer.database.enabled must be set to true to use this function.</p>
DB_To_Client_With_Progress	As DB_To_Client but with a progress bar.
Client_To_AS	<p>Uploads a named file from the client onto the application server disk. The process can also be made asynchronous, in which case you can define the name of a trigger that WebUtil will execute as soon as the upload is complete.</p> <p>Note: Asynchronous mode offers better re-drawing of the Forms screen while the transfer takes place, but will not offer the ability to carry out significant work while the transfer is taking place. The webutil.cfg parameter transfer.appsrv.enabled must be set to true to use this function, and if the transfer.appsrv.accessControl parameter is also set to true, the upload target directory must be listed in the transfer.appsrv.write list. It is sensible to call the WebUtil_File_Transfer_IsASWritable function with the target filename before starting the upload to ensure that you will be able to write to this location on the application server.</p>
Client_To_AS_With_Progress	As Client_To_As but with a progress bar.

Table 4–3 (Cont.) Available Functions in the WebUtil_File_Transfer Package

Function	Purpose
AS_To_Client	Transfers a file from the application server to the client. The webutil.cfg parameter transfer.appsrv.enabled must be set to true to use this function, and if the transfer.appsrv.accessControl parameter is also set to true, the upload target directory must be listed in the transfer.appsrv.read. list. It is sensible to call the WebUtil_File_Transfer_IsASReadable function with the target filename before starting the download to ensure that you will be able to read from this location on the application server.
AS_To_Client_With_Progress	As AS_To_Client with a progress bar.
In_Progress	If an Asynchronous update is in progress this function will return true. This should be used before starting another transfer off as only one transfer can take place at once.
Asynchronous_Upload_Success	Returns a Boolean indicating if an Asynchronous load succeeded. This should be called from the callback trigger that is executed when an Asynchronous upload has finished.
Get_Work_Area	Returns a work area directory on the application server that is private to this user. It can take a Boolean argument (defaulting to true) that indicates that WebUtil should create the private directory if it does not already exist. The private directory is created under the directory specified in transfer.appsrv.workAreaRoot (in the webutil.cfg). The directory that is created is constructed from the IP address and username of the client computer - so will be unique between end users, but multiple applications running on the same client computer will share the directory.
Is_AS_Writable	Given a file name this function returns true if the current rules defined in the webutil.cfg file will allow the file to be written.
Is_AS_Readable	Given a file name this function returns true if the current rules defined in the webutil.cfg file will allow the file to be read.

4.3.5 WebUtil_Host

The Webutil_Host package provides a series of routines for executing commands on the browser computer.

Importantly, this package contains two types. The first is a PROCESS_ID which is used to record a reference to a client side process, which can be used to obtain the output of that process or to terminate it.

The second type is OUTPUT_ARRAY, which is a PL/SQL table of VARCHAR2 holding the output from a client side command.

Table 4–4, " Available Functions in the WebUtil_Host Package" describes these functions:

Table 4–4 Available Functions in the WebUtil_Host Package

Function	Purpose
Host	Runs the specified command on the client – optionally this can return the return code. This call is blocking.
Blocking	Runs the specified command on the client – optionally this can return a process id if you need to examine the return code or output. This call is blocking.
NonBlocking	Runs the specified command on the client – optionally this can return a process id if you need to terminate the process. This call is non-blocking.
NonBlocking_With_Callback	Runs the specified command on the client, you also supply a trigger, which WebUtil will execute once the client process has completed. This call is non-blocking.
Terminate_Process	Given a process handle of a process this function will kill the process on the client.
Get_Return_Code	Given a process handle of a completed process this function will return the return code of the process as an integer.
Get_Standard_Output	Given a process handle of a completed process this function will return an OUTPUT_ARRAY containing any output that was sent to standard output by the client process.
Get_Standard_Error	Given a process handle of a completed process this function will return an OUTPUT_ARRAY containing any output that was sent to standard error by the client process.
Release_Process	Given a process handle this frees up the resources associated with the process.
Get_Callback_Process	When a client command has been called from NonBlocking_With_Callback, the callback trigger executes, this function can be used to get the Process Id of the finished process. This can then be used to obtain the output or return code.
Id_Null	Tests if a process ID is null.
Equals	Tests if two process ids represent the same process.

4.3.6 WebUtil_Session

The WebUtil_Session package provides functionality for reacting to an interruption to the Forms session. It provides a way of defining a URL that the browser should re-direct to if the Form ends or crashes. This is commonly required when the Forms server times out and you don't want the user to be left looking at a blank applet area in the browser.

Note that the session monitor will not detect that the Forms server has timed out until the user attempts to use the Form in some way. At that point they will be directed.

[Table 4–5, "Available Functions in the WebUtil_Session Package"](#) describes these functions:

Table 4–5 Available Functions in the WebUtil_Session Package

Function	Purpose
Enable_Redirect_On_TimeOut	Enables the time-out monitor and allows you to specify a URL that the browser should re-direct to when the situation occurs.
Disable_Redirect_On_TimeOut	Switches off the monitor - if you do not call this function before issuing EXIT_FORM then the browser will be still re-directed even though it's a "normal" exit.

4.3.7 WebUtil_Browser

The WebUtil_Browser package provides some calls for use when the Form that is using WebUtil is running within a browser window (see also WebUtil_SeparateFrame). [Table 4–6, "Available Functions in the WebUtil_Browser Package"](#) describes these functions.

Table 4–6 Available Functions in the WebUtil_Browser Package

Function	Purpose
BrowserMessage	Sends a message to the browser message line (as opposed to the Forms status line)
GetAppletParameter	Provides access to Applet parameters defined in your formsweb.cfg. For instance the lookAndFeel parameter.
GetAppletSize	Returns the width and height of the applet area in the browser, in pixels, as a string in the format "width,height"
ShowMenuBar	Allows you to hide the Forms Menu all together. Note: to work the following must be true: <ul style="list-style-type: none"> ■ The menu property of the form must be null ■ Logo should be set to no in the formsweb.cfg
ShowStatusBar	Allows you to hide or display the Forms status bar - much like the setting of the Console window at design time, except that the this procedure allows you to do it dynamically at runtime. Note: The Status bar will only display if the console window was initially assigned at design time.

4.3.8 WebUtil_SeparateFrame

The WebUtil_SeparateFrame package provides some calls for use when the Form that is using WebUtil is running outside of the browser window, using the parameter separateFrame=true. [Table 4–7, "Available Functions in the WebUtil_SeparateFrame Package"](#) describes these functions:

Table 4–7 Available Functions in the WebUtil_SeparateFrame Package

Function	Purpose
IsSeparateFrame	Returns a Boolean indicating that you are running in separate frame mode or not
GetSeparateFrameSize	Returns the width and height of the frame, in pixels, as a string in the format "width,height"
ShowMenuBar	Allows you to hide the Forms Menu all together. Note: to work the following must be true: The menu property of the form must be null. Logo should be set to no in the formsweb.cfg
ShowStatusBar	Allows you to hide or display the Forms status bar - much like the setting of the Console window at design time, except that the this procedure allows you to do it dynamically at runtime. Note: The Status bar will only display if the console window was initially assigned at design time.
SetTitle	Sets the title on the MDI frame
SetIcon	Sets the icon used in the control box of the MDI frame and in the Task bar etc. The icon specified can be in GIF or JPG format and will be searched for in the same way as icons for buttons within the Form.
AllowResize	Allows you to disable (or re-enable) the resizing of the MDI frame
CenterMDI	Auto-centers the MDI frame for you on the display. Comes in two versions, one of which will allow you to nudge up or down to fit in with the task bar etc.

4.3.9 WebUtil_Core

Most functions in the WebUtil_Core package are private to WebUtil. Some functions, however, can be called directly. [Table 4–8, "Available Functions in the WebUtil_SeparateFrame Package"](#) describes these functions:

Table 4–8 Available Functions in the WebUtil_SeparateFrame Package

Function	Purpose
IsError	Checks if the last WebUtil call that you made succeeded.

4.4 Utility Functions

These functions do not perform client side integration; however, they do provide useful functionality which you can freely re-use within your own applications.

4.4.1 DelimStr

DelimStr is a package which provides a set of calls for interacting with delimited strings such as comma delimited strings. It provides APIs to treat the delimited string as an array, allowing the programmer to get and set members of the array by index number, and handling the storage of multiple data types within it.

4.4.2 Show_WebUtil_Information

This procedure causes the hidden WebUtil window to display in your Form, providing you with the versions of all the WebUtil components. The procedure can optionally be called with a Boolean argument. Passing a false to this parameter will cause the information window to hide again.

4.4.3 WebUtil_Util

This package contains a single function `BoolToStr()` for converting a Boolean value to text. By default the strings `true` or `false` will be returned depending on the value of the Boolean, however, the `true` and `false` strings can be overridden to reflect other values, for instance `Yes` and `No`.

4.5 Internal APIs

Certain packages within WebUtil are in place purely to help with the requirements of the public APIs. These packages should *not* be called directly:

- `Java_Appserv_Reader`
- `Java_Appserv_Writer`
- `Java_Exception`
- `Java_File`
- `Java_System`

Using WebUtil in Your Applications

The following sections contain information about using WebUtil in your Forms applications:

- [Section 5.1, "Adding The Required Objects"](#)
- [Section 5.2, "Seeing the PJC at Design Time"](#)
- [Section 5.3, "Adding WebUtil Code"](#)

5.1 Adding The Required Objects

WebUtil is designed to be as simple as possible to use within your applications. For each form that needs to use WebUtil functions, do the following:

1. Attach the `webutil.pll` library. This library contains all of the APIs available through WebUtil and is the only way of calling the WebUtil functions. Attempting to call the WebUtil PJC's directly will result in errors.
2. Subclass the WebUtil object group. `webutil.olb` contains a WebUtil object group, which you can simply drag into your Form. This object group contains all of the PJC's that implement the WebUtil functionality, along with a Canvas and Window to hold them and the alert used for popup errors. Normally the WebUtil window is hidden so its presence will not interfere with your user interface.

Note: The order of these operations is important. If you subclass the object group before attaching the library, the triggers within the object group will be marked as uncompiled and calls to WebUtil will fail with a WUC-15 error unless you do a Recompile All (Control-shift-K) on your module.

Once the WebUtil objects have been attached to your Forms you can start to use the APIs exposed by the PLL and are detailed in [Chapter 4, "Functions in WebUtil"](#). There is no need to carry out any initialization steps in your code, as the utility automatically initializes itself.

Note that the WebUtil block that contains the PJC is normally hidden but if your application navigates using the `NEXT_BLOCK` and `PREVIOUS_BLOCK` Built-ins, your users may inadvertently navigate to the block. You should check your navigation logic to ensure that this does not happen.

The WebUtil block itself also serves as an "About Screen" for the utility where it lists the version number of the PLL and the versions of each PJC that is used to implement the various functions. To view this information at runtime, call the `SHOW_WEBUTIL_INFORMATION` function.

5.2 Seeing the PJC at Design Time

At design time there is no need to interact with or change the properties of any of the objects in the WebUtil object group. However, you can view the WebUtil canvas to view the PJC versions that are in use. By default, opening the WebUtil canvas in the Layout Editor results in errors because Forms cannot find the `frmwebutil.jar` file to instantiate the beans. To make this Jar file available, edit the `FORMS_BUILDER_CLASSPATH` in the environment or in the registry and include the physical location of the `frmwebutil.jar` file, for example, `ORACLE_HOME\forms\java\frmwebutil.jar` and the physical location of `frmall.jar`.

Note: The total length of the `FORMS_BUILDER_CLASSPATH` cannot exceed 512 characters.

5.3 Adding WebUtil Code

Once the WebUtil library has been attached to your form you can start to add calls to the various PL/SQL APIs defined by the utility. However, there is an important restriction in the use of WebUtil functions: WebUtil can only start to communicate with the client once the Form has instantiated the WebUtil PJC. This means that you cannot call WebUtil functions before the Forms user interface is rendered. This would include triggers such as `PRE-FORM`, `WHEN-NEW-FORM-INSTANCE` and `WHEN-NEW-BLOCK-INSTANCE` for the first block in the Form.

Likewise you cannot call functions after the user interface has been destroyed for instance in `POST-FORM`.

If you do need to use WebUtil at Forms startup, it is recommended that you create a timer with a short duration (about 1ms) in your `WHEN-NEW-FORM-INSTANCE` trigger, then make the WebUtil call in a `WHEN-TIMER-EXPIRED` trigger.

5.4 Using OLE Commands

Some OLE commands work fine in a client/server environment, as well as in Web deployed applications (on the server), but when you use WebUtil to enable client side functions, they do not work.

For example, the following code will work by running OLE as client/server, in a 3-tier environment, or on the application server:

```
DECLARE
    app  OLE2.OBJ_TYPE;
    args OLE2.LIST_TYPE;
BEGIN
    -- create a new document
    app := OLE2.CREATE_OBJ('Word.Basic');
    OLE2.INVOKE(app, 'FileNew');
    -- insert data into new document from long item
    args := OLE2.CREATE_ARGLIST;
    OLE2.ADD_ARG(args, :long_item);
    OLE2.INVOKE(app, 'Insert', args);
    OLE2.DESTROY_ARGLIST(args);
    -- save document as example.tmp
    args := OLE2.CREATE_ARGLIST;
    OLE2.ADD_ARG(args, 'c:\temp\example.tmp');
    OLE2.INVOKE(app, 'FileSaveAs', args);
    OLE2.DESTROY_ARGLIST(args);
    -- load example.tmp into ole item in form
```

```

FORMS_OLE.ACTIVATE_SERVER('msw_obj');
FORMS_OLE.INITIALIZE_CONTAINER('msw_obj', 'c:\temp\example.tmp');
FORMS_OLE.CLOSE_SERVER('msw_obj');
-- clear data from example.tmp
OLE2.INVOKE(app, 'EditSelectAll');
OLE2.INVOKE(app, 'EditClear');
-- save empty example.tmp
args := OLE2.CREATE_ARGLIST;
OLE2.ADD_ARG(args, 'c:\temp\example.tmp');
OLE2.INVOKE(app, 'FileSaveAs', args);
OLE2.DESTROY_ARGLIST(args);
-- close example.tmp
args := OLE2.CREATE_ARGLIST;
OLE2.ADD_ARG(args, 2);
OLE2.INVOKE(app, 'FileClose', args);
OLE2.DESTROY_ARGLIST(args);
-- exit MSWord
OLE2.RELEASE_OBJ(app);
END;

```

However, when you change this code to use CLIENT_OLE2 (using search-and-replace) it doesn't work.

The following example is code which *will* convert to CLIENT_OLE2:

```

DECLARE
  app      CLIENT_OLE2.OBJ_TYPE;
  docs     CLIENT_OLE2.OBJ_TYPE;
  doc      CLIENT_OLE2.OBJ_TYPE;
  selection CLIENT_OLE2.OBJ_TYPE;
  args     CLIENT_OLE2.LIST_TYPE;
BEGIN
  -- create a new document
  app := CLIENT_OLE2.CREATE_OBJ('Word.Application');
  CLIENT_OLE2.SET_PROPERTY(app, 'Visible', 1);

  docs := CLIENT_OLE2.GET_OBJ_PROPERTY(app, 'Documents');
  doc  := CLIENT_OLE2.INVOKE_OBJ(docs, 'add');

  selection := CLIENT_OLE2.GET_OBJ_PROPERTY(app, 'Selection');

  -- insert data into new document from long item
  CLIENT_OLE2.SET_PROPERTY(selection, 'Text', 'this is a test message');

  -- save document as example.tmp
  args := CLIENT_OLE2.CREATE_ARGLIST;
  CLIENT_OLE2.ADD_ARG(args, 'c:\example.doc');
  CLIENT_OLE2.INVOKE(doc, 'SaveAs', args);
  CLIENT_OLE2.DESTROY_ARGLIST(args);

  -- close example.tmp
  args := CLIENT_OLE2.CREATE_ARGLIST;
  CLIENT_OLE2.ADD_ARG(args, 0);
  CLIENT_OLE2.INVOKE(doc, 'Close', args);
  CLIENT_OLE2.DESTROY_ARGLIST(args);

  CLIENT_OLE2.RELEASE_OBJ(selection);
  CLIENT_OLE2.RELEASE_OBJ(doc);
  CLIENT_OLE2.RELEASE_OBJ(docs);

```

```
-- exit MSWord
CLIENT_OLE2.INVOKE(app, 'Quit');
END;
```

If you are running client/server Forms you *may* be using code like

```
app := OLE2.CREATE_OBJ('Word.Basic');
```

or

```
app := OLE2.CREATE_OBJ('Word.Application');
```

Both of these code examples initialize an OLE interaction with Microsoft Word. The difference between the two examples is that Microsoft changed the interface between their Office95 and their Office97 applications. Webutil has been tested against the recent (and current) Office97 interface (Word.Application).

So, if you are using the Office97 interface then doing a search and replaced to change OLE2 to CLIENT_OLE2 will probably work. If you are using the old (obsoleted by Microsoft) interface then doing this search and replace may not work.

Performance Tuning Considerations

This chapter contains the following sections:

- [Section 6.1, "Introduction"](#)
- [Section 6.2, "Optimizing Your Use of WebUtil"](#)

6.1 Introduction

WebUtil as a utility is designed primarily to provide extra client side functionality. This functionality comes at a cost. The utility has to exchange messages between the application server and the client, resulting in traffic over the network. Additionally the process of marshalling and passing data across the network will take time.

Taking for instance the case of OLE integration; calling OLE on the application server or the equivalent OLE integration in client server will be much faster than making remote calls to OLE on the client.

However, it is anticipated that the kinds of applications that require WebUtil functionality will primarily be aimed at intranet use over known network conditions of latency and bandwidth.

6.2 Optimizing Your Use of WebUtil

The following suggestions should help you to reduce the network and other costs inherent in using the utility.

1. Only WebUtil Enable Forms that actually need the functionality. Each form that is WebUtil enabled will generate a certain amount of network traffic and memory usage simply to instantiate the utility, even if you don't use any WebUtil functionality.
2. Disable logging and error recording on the server except for during testing and debugging. Active logging will slow down the client considerably and generate a lot of network traffic.
3. Think carefully about object reuse. Whenever you obtain an object handle from the client, such as a `Client_OLE2.OBJ_TYPE` or a `WebUtil_C_API.FunctionHandle`, this represents one or more Java objects created and held by the Forms client. Creating these objects is expensive and they should be re-used when possible, rather than being destroyed and re-created.
4. Use functions in an efficient manner. For instance if you need to read a text file on the client computer, it will probably be more efficient to transfer the file to the users's work area using `WebUtil_File_Transfer` and then read it with normal `TEXT_IO`, rather than reading it line by line across the network

5. Once Client side objects have been finished with you should call the relevant destroy or release method, to free up the client side resources. If you do not, they will be automatically cleaned up when the form is closed, but before then they will be consuming memory resources on the client. If you persist a resource so that it can be referenced in another form you must take care to explicitly clean it up.
6. Be aware there may be multiple ways to obtain the same information or execute a function. This applies particularly to D2KWUTIL functions which are often duplicated by more efficient WebUtil functions.
7. The file transfer functions Client_To_AS and AS_To_Client use Java calls from the Forms engine on the middle tier. There is a time penalty for the very first call as the JVM is started and the extra memory overhead of running the JVM. Upload and download from the database does not use Java on the middle tier and is more efficient.
8. The most efficient file download facility is the URL_To_Client function. This should be used in preference to AS_To_Client if possible, but does require that the source directory is mapped as a virtual directory to the web server.
9. If the transfer from the client to the database is done in a forms session, a commit is required after file transfer (to the database) before exiting the Form to save the changes. Otherwise, the next time you start that Form, WebUtil raises error "WUT-111: Database LOB is un-initialized wu_test_table.BLOB - where ID=1".

Tracing and Diagnostics

In this chapter the tracing and diagnostic facilities of WebUtil are discussed. These cover three areas:

- [Section 7.1, "Connection Logging"](#)
- [Section 7.2, "Error Handling"](#)
- [Section 7.3, "Diagnostic Tracing"](#)

7.1 Connection Logging

If administrators are interested in knowing detailed information about the users and computers that are connecting to their application, WebUtil provides a connection logging mechanism which is separate from the other tracing and logging within the utility.

Connection logging is enabled by setting the flag `logging.connections` in `webutil.cfg` to `true`.

Connection Logging gets information about the connected user directly from the client computer, including the IP address and operating system user name. Note that because the IP address is obtained from the browser computer, it will be the real IP address of the client rather than the IP address of a firewall, or router, which the client is behind. The form that the user is connected to will also be recorded.

7.2 Error Handling

In some circumstances WebUtil may raise an error, this may be due to the misuse of an API or some internal problem. The errors that can be raised are detailed in [Appendix B, "WebUtil Error Reference"](#). Error logging is controlled by the `formsweb.cfg` parameter `WebUtilErrorMode`. This can be set to one of four values:

- Console (the default)
- Alert
- Server
- All

Errors are always displayed on the Java console, these switches force the errors to be reported elsewhere as well.

Setting the parameter to Alert shows the error in a pop up alert in the browser. Setting it to "server" streams the error back to the server log file. In order for this to work, the `webutil.cfg` settings `logging.file` and `logging.enabled` must be defined.

Setting the parameter to All both shows the error in an alert on the client and in the server log file.

7.3 Diagnostic Tracing

Another diagnostic feature of WebUtil is the trace facility. This can be used to see the various calls that are made to the client and detail about them.

Tracing is enabled using the WebUtilLogging parameter in the formsweb.cfg file. This can be switched to one of four values:

- Off (the default)
- Console
- Server
- All

By default logging messages are suppressed. Much like errors they can be directed elsewhere using these settings. In order for server side logging to work, the webutil.cfg logging.file must be defined, the logging.enabled parameter set to true and the logging.erroronly parameter set to false.

You can also control the detail of the logging that is carried out using the formsweb.cfg parameter WebUtilLoggingDetail. This can be set to two detail levels:

- Normal (the default)
- Detailed

Runtime Setup Checklist

Use the following checklist to help you setup WebUtil in a production environment.

Table A-1 WebUtil Setup Check List

Task	Reference	Done
1. Create or choose a database schema to own the WEBUTIL_DB package. This schema needs Create Procedure privilege and access to certain DBMS_ and UTL_ packages	Chapter 2.4, "Configuring Your Database for WebUtil" Schema: _____	
2. Run the create_webutil_db.sql script from the /forms directory using the schema created in [3]	Chapter 2.4, "Configuring Your Database for WebUtil"	
3. Create a public Synonym WEBUTIL_DB for the package	Chapter 2.4, "Configuring Your Database for WebUtil"	
4. Grant execute on the WEBUTIL_DB synonym to any schemas that need to do database upload / download from the schema that owns the WEBUTIL_DB package (see [3])	Chapter 2.4, "Configuring Your Database for WebUtil"	
5. Define the virtual directory "/webutil" pointing to the forms/webutil directory of the ORACLE_HOME directory (see [2])	Chapter 2.5, "Configuring Oracle HTTP Server for WebUtil"	

Table A–1 WebUtil Setup Check List

Task	Reference	Done
6. For each application that needs to be WebUtil enabled, in the formsweb.cfg config entry for the application add the entry: webUtilArchive= frmwebutil.jar	Chapter 2.6, "WebUtil Entries in the formsweb.cfg File"	
7. In the formsweb.cfg config entry for the application define the locations of the WebUtil template .htm files. Define entries for: <ul style="list-style-type: none">■ baseHTMLjinitiator■ BaseHTMLjpi■ baseHTML■ These should point at the relevant templates in the /server directory of your installation location (see [2])		

Table A-1 WebUtil Setup Check List

Task	Reference	Done
8. For the environment files (envFile=) used by each configuration in the formsweb.cfg that uses WebUtil <ul style="list-style-type: none">■ Ensure that the FORMS_PATH includes the /forms directory of the WebUtil install location (see [2]).■ Add a new environment value WEBUTIL_CONFIG=. This should point at the file /server/webutil.cfg in the WebUtil install location (see [2])■ Include the full path and name of /forms/java/frmwebutil.jar (see [2]) in the CLASSPATH= setting.■ Include the full path and name of the Java rt.jar in the CLASSPATH= setting. rt.jar is the Java runtime jar file and can be found in the ORACLE_HOME/jdk/jre/lib directory or in any other Java 1.3 or 1.4 install.■ (Windows Only) If not defined in your base PATH, the JRE /bin/classic directory (e.g. ORACLE_HOME/jdk/jre/bin/classic) should be added to a PATH= setting in the .env file so that WebUtil can find the jvm.dll file This path should also include directories required to actually run Forms itself.■ (UNIX only) Ensure that the LD_LIBRARY_PATH (or equivalent) includes the location of the libjava.so file	Chapter 2.7, "WebUtil Entries in the Environment File"	
9. Configure your webutil.cfg file <ul style="list-style-type: none">■ Configure your tracing / logging options■ Define custom system libraries for download if required (Win32 only)■ Configure file transfer permissions (if required)■ Define the work area	Chapter 3.2.1, "Logging Options" Chapter 3.2.2, "Installation Options" Chapter 3.2.3, "File Upload and Download Options"	
10. Add the /forms directory (see [2]) to your environment or Registry FORMS_PATH on your development computer(s)	Chapter 5, "Using WebUtil in Your Applications"	

Table A-1 WebUtil Setup Check List

Task	Reference	Done
11. (Optional) Add the path (see [2]) and name of forms/java/frmwebutil.jar to your FORMS_BUILDER_CLASSPATH in your development computer environment or Registry.	Chapter 5.2, "Seeing the PJC at Design Time"	
12. For each Form that needs WebUtil functionality <ul style="list-style-type: none">■ Attach the webutil.pll file■ Subclass the WebUtil Object group from webutil.olb■ Ensure that the navigation logic in your Form bypasses the WEBUTIL block■ When running WebUtil applications from the builder, change your Edit Preferences Runtime Settings in the Menu to ensure that the Application Server URL setting specifies a configuration in the formsweb.cfg that is WebUtil "enabled".	Chapter 5.1, "Adding The Required Objects"	

WebUtil Error Reference

This section contains a complete list of all WebUtil error messages, their meanings, and where appropriate, their resolution. Errors are prefixed with a three letter suffix which indicates which JavaBean has raised the error, for instance WUT for errors from the file transfer beans. Errors prefixed with WUC are common errors that are raised by the WebUtil JavaBean superclass.

All of the error messages can be changed by editing the appropriate message bundle (see [Chapter 8, "Customizing WebUtil"](#) for more information).

Errors fall into two categories – User errors which are generally going to be caused by misuse of the supplied APIs or some other factor out of WebUtils control such as a hard disk filling up. And Internal errors which are thrown when something serious has happened within the utility code itself. In all cases, when an error is received it is a useful technique to switch WebUtil logging on to gather as much information as possible about the circumstances leading up to the problem. See [Chapter 7, "Tracing and Diagnostics"](#) for more information.

Table B–1 **WebUtil Error Reference**

Error	Description
WUC-1	Component has no parent frame. Type: Internal Description: WebUtil cannot find the owning Frame for the Forms application. This frame is used as the parent for modal dialogs such as Alerts. Resolution: WebUtil is designed to run within the context of the Forms Java Applet. This error should never occur if run in that way.
WUC-2	Invalid logmode supplied. Type: User Description: An invalid logging mode was requested. Resolution: The following logging modes are valid: OFF ON CONSOLE SERVER ALL.
WUC-3	Invalid log detail level supplied. Type: User Description: An invalid detail level was specified for logging. Resolution: The following detail levels are valid: NORMAL DETAILED.

Table B–1 (Cont.) WebUtil Error Reference

Error	Description
WUC-4	<p>Invalid errormode supplied.</p> <p>Type: User</p> <p>Description: An invalid error mode was requested.</p> <p>Resolution: The following error modes are valid: CONSOLE ALERT SERVER ALL</p>
WUC-5	<p>No WebUtil Configuration File specified.</p> <p>Type: User</p> <p>Description: WebUtil cannot find its configuration file. This file is required to set up features like logging</p> <p>Resolution: The location of the webutil configuration file is determined in two ways either using the environment variable WEBUTIL_CONFIG on the application server or by setting the parameter WebUtilConfig in your formswweb.cfg file. Set in either way, the value should point to a valid file on the application server.</p>
WUC-6	<p>Unable to read WebUtil Configuration File.</p> <p>Type: User</p> <p>Description: The WebUtil configuration file that was specified cannot be read. This file is required to set up features like logging</p> <p>Resolution: The location of the webutil configuration file is determined in two ways either using the environment variable WEBUTIL_CONFIG on the application server or by setting the parameter WebUtilConfig in your formswweb.cfg file. Set in either way, the value should point to a valid file on the application server.</p>
WUC-7	<p>Unable to write to WebUtil Logging File.</p> <p>Type: User</p> <p>Description: The logging file that was specified in the WebUtil configuration file cannot be written to.</p> <p>Resolution: Ensure that the logging file location exists and if the file already exists it is not read-only</p>
WUC-8	<p>Server side logging was requested but no log file was specified - logging is disabled.</p> <p>Type: User</p> <p>Description: The logging.file entry in the WebUtil configuration file is missing or empty so the correct logging file cannot be identified.</p> <p>Resolution: Set the logging.file entry in your WebUtil configuration file</p>
WUC-9	<p>Unable to obtain LocalHost info from InetAddress.</p> <p>Type: Internal</p> <p>Description: WebUtil could not get information about the host computer from Java.</p> <p>Resolution: Ensure the Jar file containing WebUtil is signed.</p>

Table B–1 (Cont.) WebUtil Error Reference

Error	Description
WUC-10	Could not obtain Host IP Address. Type: Internal Description: WebUtil could not get the IP Address of the host computer from Java. Resolution: Ensure the Jar file containing WebUtil is signed.
WUC-11	WebUtil must be signed in order to work. Type: Internal Description: WebUtil could not access secure information from the applet it needs to be signed to allow it to work. Resolution: Ensure the Jar file containing WebUtil is signed.
WUC-12	Object Cache Error: Object is not the expected {0} type. Type: User Description: An object stored in the WebUtil client side Object Cache cannot be cast to the expected type. Resolution: Check your code - do no hardcode object references in your application and only use the correct type of object handle for the operation. For instance do not pass an OLE object handle to the text_io functions.
WUC-13	Object Cache Error: No object Cache present. Type: User Description: WebUtil tried to read from the Object cache but it does not exist yet. Resolution: You should only submit an object handle to WebUtil that was created by WebUtil
WUC-14	Object Cache Error: Specified object handle {0} not found in the cache. Type: User Description: WebUtil tried to read an object from the object cache but no object was found at that position. Resolution: Check that you have not already destroyed the object before this call.
WUC-15	Unexpected error, Exception: {0} Type: User Description: A last resort to any expected exception. Resolution: This is a generic error for any unexpected exception. Diagnose the exception stack in the console.
WUC-16	When setting a client side property a null value was supplied. Type: User Description: A call to set a property in webutil.properties was made but no property string was sent. Resolution: The property string should be in the format propname=value

Table B–1 (Cont.) WebUtil Error Reference

Error	Description
WUC-17	Information was missing from the supplied file download request {0}. Type: User Description: The request asking for a file download was not formed correctly Resolution: Only use the supplied PL/SQL APIs to call WebUtil functions
WUC-18	Invalid server side file specification {0}. Type: User Description: The requested server side file URL cannot be read. Resolution: Check the URL
WUC-19	Unable to write to local file {0}. Type: User Description: The attempt to make a local copy of a server side file failed. Resolution: Check that there is space on the client side disk and that the user has permissions to the specified directory
WUC-20	The stated size of the source file {0} does not match that of the downloaded file {1}. Type: User Description: WebUtil has downloaded from an URL and the file that it has created is not the same size as the Size stated in the download command for the original file. This may be because the original file was incorrectly specified. Resolution: Check the downloaded file in a text editor for more information and/or check there is space on the target device.
WUC-21	The downloaded file {0} was not created. Type: User Description: The attempt to make a local copy of a server side file failed. Resolution: Check that there is space on the client side disk and that the user has permissions to the specified directory.
WUC-22	The global object reference was incorrect ({0}). Type: User Description: An invalid object reference was passed to the client. Resolution: Only persist a valid reference.
WUC-23	Failed to download system library. {0} Type: User Description: Library required to be installed on client system could not be downloaded. Resolution: Make sure that install.syslib.location is correctly set and that the URL is valid. The actual download URL can be see from the console.

Table B–1 (Cont.) WebUtil Error Reference

Error	Description
WUC-24	Error reading URL {0} Type: User Description: Download from URL to client failed. Resolution: Make that the URL is valid and correct.
WUC-25	URL cannot be null. Type: User Description: An empty URL was passed to the client. Resolution: enter a URL in the Source field.
WUC-26	URL {0} has zero content length. Download will be cancelled. Type: User Description: The actual content of the URL is empty. For such cases, the client file will not be created. Resolution: No resolution is required for this error. No client file will be created.
WUC-27	URL {0} is not a trusted site from which WebUtil functionality can be downloaded. Type: User Description: The domain suffix of the URL used to run the Form is not in the list of TrustedDomains.txt. which contains the list of trusted domains from which WebUtil functionality can be downloaded. Resolution: Read the comments in TrustedDomains.txt on how to write trusted domain filters and ensure that the list contains this URL.
WUC-28	Error while reading the list of trusted domains from the WebUtil archive. Type: User Description: Read error while reading the TrustedDomains.txt. Resolution: TrustedDomains.txt is in oracle.forms.webutil.common package. Make sure it is correctly added to the WebUtil archive.
WUT-100	Bad file information string: {0}. Type: User Description: WebUtil was asked to upload or download a file, but incorrect information was supplied. Resolution: Supply a valid file name.
WUT-101	File to download has an invalid size: {0}. Type: User Description: The file size of a download file was defined as null, a negative value or some other non integer value. Resolution: Only use the WebUtil APIs to download files.
WUT-102	Input Buffer Empty. Type: User Description: Zero bytes where read from a file during upload. Resolution: Check the client file is not empty.

Table B–1 (Cont.) WebUtil Error Reference

Error	Description
WUT-103	File Transfer already in progress cannot upload {0}. Type: User Description: WebUtil is already in the middle of processing a file and cannot start on another simultaneous transfer. Resolution: Finish one file transfer before starting the next.
WUT-104	File Transfer already in progress cannot download {0}. Type: User Description: WebUtil is already in the middle of processing a file and cannot start on another simultaneous transfer. Resolution: Finish one file transfer before starting the next
WUT-105	Unable to open file {0} for writing. Type: User Description: WebUtil was trying to download a file but could not open the file on to client to write to. Resolution: Check that you have privileges to write to the specified file and there is enough disk space.
WUT-106	Unable to open file {0} for reading. Type: User Description: WebUtil was trying to upload a file but could not open the file on to client to read. Resolution: Check that you have privileges to read from the specified file.
WUT-107	Decode buffer is empty. Type: Internal Description: A data buffer sent from PL/SQL was empty. Resolution: Contact Oracle.
WUT-108	Error writing to file {0}. Exception: {1} Type: User Description: WebUtil encountered an error trying to write to the specified download file. Resolution: Check that there is enough disk space and that no other process has locked the file.
WUT-109	Error reading from file {0}. Exception: {1} Type: User Description: WebUtil encountered an error trying to read from the specified upload file. Resolution: Check that the file exists and can be accessed.
WUT-110	Database LOB is of zero length. Type: User Description: An attempt was made to download zero bytes from a LOB to the client. Resolution: Ensure that there is data to download.

Table B–1 (Cont.) WebUtil Error Reference

Error	Description
WUT-111	<p>Database LOB is un-initialized.</p> <p>Type: User</p> <p>Description: An attempt was made to download from a BLOB column that is null.</p> <p>Resolution: Ensure that there is data to download. A commit is required after file transfer (to the database) before exiting the Form to save the changes. A transfer succeeds only if the transfer is done in the same session.</p>
WUT-112	<p>Invalid open mode for BLOB. Value should be W or R.</p> <p>Type: User</p> <p>Description: An invalid open mode was used.</p> <p>Resolution: Use a valid mode.</p>
WUT-113	<p>Too many rows match the supplied where clause.</p> <p>Type: User</p> <p>Description: A call to openBlob() should pass a where clause sufficient to identify one and only one row.</p> <p>Resolution: Correct Where Clause.</p>
WUT-114	<p>SQL Error.</p> <p>Type: User</p> <p>Description: The combination of table name, column name and where clause passed to OpenBlob() did not generate a valid SQL statement.</p> <p>Resolution: See SQL error for details.</p>
WUT-115	<p>Checksum Error.</p> <p>Type: Internal</p> <p>Description: When closing a blob, the size of the input and output files did not match.</p> <p>Resolution: Check for database errors.</p>
WUT-116	<p>Transfer already in progress.</p> <p>Type: User</p> <p>Description: A transfer is already taking place. Only one transfer can be carried out at a time.</p> <p>Resolution: Check that one transfer is complete before starting the next.</p>
WUT-117	<p>Application Server file name cannot be null.</p> <p>Type: User</p> <p>Description: The name of a file to read or write on the application server must be supplied.</p> <p>Resolution: Supply a file name.</p>
WUT-118	<p>Application Server file does not exist or is of zero length.</p> <p>Type: User</p> <p>Description: The file on the application server must exist.</p> <p>Resolution: Supply the name of a valid file.</p>

Table B-1 (Cont.) WebUtil Error Reference

Error	Description
WUT-119	Error reading data from Application Server file. Type: User Description: The file in the application server could not be read for some reason. Resolution: Consult the exception error information
WUT-120	Zero bytes read from Application Server file. Type: User Description: The file the application server could not be read for some reason. Resolution: Check the file has not been corrupted or removed.
WUT-121	This file transfer has been forbidden by the Administrator. Type: User Description: The Administrator has forbidden file transfers in the webutil.cfg file. Resolution: If the restriction is not deliberate, update your webutil.cfg file with the correct permissions.
WUT-122	Java Functions are not available on the application server. Type: User Description: Forms services in this configuration is unable to run mid tier Java code. Resolution: Edit the .env file for this configuration to set the correct path and classpath information to run Java.
WUT-122	The frmwebutil.jar file cannot be found on the application server classpath - Some file transfer functions will not work. Type: User Description: The frmwebutil.jar file contains routines to handle file transfer to and from the application server. Without access to this Jar file these functions cannot work. Resolution: Correct your classpath in the .env file for this configuration
WUT-123	Unable to open file for writing on the application server. Type: User Description: An exception was raised while trying to open a server side file for writing during upload. Resolution: Check the webutil log for more information about the cause of the error
WUT-124	Unable write data on the application server. Type: User Description: An exception was raised while trying to write to a server side file during upload. Resolution: Check the webutil log for more information about the cause of the error.

Table B–1 (Cont.) WebUtil Error Reference

Error	Description
WUT-125	<p>Checksum error closing file on application server.</p> <p>Type: User</p> <p>Description: When an upload to the application server completed the file size was not the value expected.</p> <p>Resolution: Retry the upload operation and check for sufficient disk space / access on the middle tier</p>
WUT-126	<p>Error closing application server file.</p> <p>Type: User</p> <p>Description: An exception was raised while trying to close a server side file during upload.</p> <p>Resolution: Check the webutil log for more information about the cause of the error</p>
WUT-127	<p>Unable to create work area.</p> <p>Type: User</p> <p>Description: Webutil failed trying to create a temporary work area.</p> <p>Resolution: Check disk access permissions.</p>
WUT-128	<p>Exception creating work area.</p> <p>Type: User</p> <p>Description: An exception was raised while trying to create a temporary work area.</p> <p>Resolution: Check the WebUtil log for more information about the cause of the error.</p>
WUT-129	<p>Error in reading client file <client_file_name>.</p> <p>Type: User</p> <p>Description: Before the client image is shown in the image item holder, the image is uploaded to a temp directory in the application server. Due to a bug in client_to_AS, an empty file is created if the client file does not exist or not readable. This exception occurs if the client file is not readable.</p> <p>Resolution: Check that the file exists and is readable (has correct permissions and is not corrupt).</p>
WUT-130	<p>Client file name cannot be null.</p> <p>Type: User</p> <p>Description: Occurs when the client file name for file upload or download is null.</p> <p>Resolution: Supply the name of a valid file.</p>
WUT-131	<p>Invalid client file name. Cannot create client file.</p> <p>Type: User</p> <p>Description: occurs when client file cannot be created either due to invalid parent directory or client Java process (or its owner/ user) has no write permission to the parent directory or file.</p> <p>Resolution: Check that you have write privileges to the directory and the specified file.</p>

Table B–1 (Cont.) WebUtil Error Reference

Error	Description
WUT-132	<p>Invalid Application Server file name. Cannot create application server file.</p> <p>Type: User</p> <p>Description: Occurs when application server file cannot be created either due to invalid parent directory or application server process (or its owner/user) has no write permission to the parent directory or file.</p> <p>Resolution: Check that you have write privileges to the directory and the specified file.</p>
WUT-133	<p>Client file {0} has zero length. Upload cannot be done</p> <p>Type: User</p> <p>Description: The size of the client file shows 0 byte. Upload will not be done.</p> <p>Resolution: There is no need to do anything for this error because the file to be uploaded is of zero size and will not be uploaded.</p>
WUT-134	<p>frmwebutil.jar not in the application server CLASSPATH.</p> <p>Type: User</p> <p>Description: frmwebutil.jar is not present in CLASSPATH of the Application Server.</p> <p>Resolution: Add frmwebutil.jar path in CLASSPATH of your default.env</p>
WUT-135	<p>The application server classpath has frmwebutil.jar path, which does not exist.</p> <p>Type: User</p> <p>Description: The frmwebutil.jar path given in CLASSPATH in forms .env file does not exist or is not a valid path</p> <p>Resolution: Make sure that frmwebutil.jar path in CLASSPATH in forms .env file is valid and exists.</p>
WUF-200	<p>Unable to open file {0} for read; Exception: {1}.</p> <p>Type: User</p> <p>Description: WebUtil was unable to open the specified file to read it.</p> <p>Resolution: Check that the specified file exists and is not currently locked by another process.</p>
WUB-600	<p>Bad value for CODEBASE</p> <p>Type: User</p> <p>Description: MalformedURLException occurred while trying to form icon URL using code base and a given iconpath</p> <p>Resolution: Make sure that the icon URL path is a valid URL, either absolute or relative to the Forms code base.</p>
WUB-601	<p>Bad value for DOCUMENTBASE</p> <p>Type: User</p> <p>Description: MalformedURLException occurred while trying to form icon URL using document base and a given iconpath.</p> <p>Resolution: Make sure that the icon URL path is a valid URL, either absolute or relative to the Forms document base.</p>

Table B–1 (Cont.) WebUtil Error Reference

Error	Description
WUB-602	<p>Icon name for MDI window must be supplied.</p> <p>Type: User</p> <p>Description: Icon path is null or empty.</p> <p>Resolution: Supply a valid icon path.</p>
WUB-603	<p>Unable to load icon image {0}</p> <p>Type: User</p> <p>Description: Loading of the icon image failed.</p> <p>Resolution: Make sure that the icon URL path is a valid URL, either absolute or relative to the forms document or code base.</p>
WUF-201	<p>Unable to open file {0} for writing (mode={1}); Exception: {2}.</p> <p>Type: User</p> <p>Description: WebUtil was unable to open the specified file to write to.</p> <p>Resolution: Check that the specified file exists and is not currently locked by another process.</p>
WUF-202	<p>Unable to write to file; Exception: {0}.</p> <p>Type: User</p> <p>Description: WebUtil was unable to write to the specified file.</p> <p>Resolution: Check that the file or media is not read only and the the media has sufficient space.</p>
WUF-203	<p>Unable to read from file; Exception: {0}.</p> <p>Type: User</p> <p>Description: WebUtil was unable to read a line from the specified file.</p> <p>Resolution: Check that the file media has not been removed or the file closed before reading.</p>
WUF-204	<p>Unable to close file; Exception: {0}.</p> <p>Type: User Description: WebUtil was unable to close the specified file.</p> <p>Resolution: Check that the file media has not been removed or the file is already closed.</p>
WUF-205	<p>Invalid value {{0}} supplied for number of new lines.</p> <p>Type: User</p> <p>Description: The second argument to NEW_LINE should be a positive integer.</p> <p>Resolution: Correct the code that called the function.</p>
WUF-206	<p>Invalid value {{0}} supplied as a File Handle.</p> <p>Type: User</p> <p>Description: The supplied file handle was not a positive integer.</p> <p>Resolution: Ensure that you populate your file handles by opening a file before reading or writing.</p>

Table B–1 (Cont.) WebUtil Error Reference

Error	Description
WUF-207	<p>Null value supplied as a File Handle.</p> <p>Type: User</p> <p>Description: The supplied file handle was not a positive integer.</p> <p>Resolution: Ensure that you populate your file handles by opening a file before reading or writing.</p>
WUF-208	<p>Too many arguments have been supplied for a call to CLIENT_TEXT_IO.PUTF.</p> <p>Type: User</p> <p>Description: The PUTF function only supports up to 5 substitution variables.</p> <p>Resolution: Only call putf through the supplied PL/SQL interface.</p>
WUF-209	<p>Copy file failed.</p> <p>Type: User</p> <p>Description: After the copy operation the source and target files are not of the same size.</p> <p>Resolution: Check disk space or read permissions on the client computer.</p>
WUF-210	<p>Invalid or non-existent client directory {0}. Error in reading client directory.</p> <p>Type: User</p> <p>Description: The parent directory of the client file is not valid or does not exist.</p> <p>Resolution: Make sure that the client file path exists.</p>
WUF-211	<p>Source and target files are same. Cannot copy client file {0}</p> <p>Type: User</p> <p>Description: The source and target file path are same.</p> <p>Resolution: Specify a different target file path.</p>
WUF-212	<p>Client directory name to be listed cannot be null</p> <p>Type: User</p> <p>Description: The directory path given for listing is empty or null.</p> <p>Resolution: Make sure to pass a valid directory for listing. If you want to list the drives in windows, use WEBUTIL_FILE.DIRECTORY_ROOT_LIST</p>
WUI-303	<p>Null value supplied as the name of a system property.</p> <p>Type: User</p> <p>Description: When requesting a system property a null value was supplied as its name.</p> <p>Resolution: Request a valid Java System property such as user.name.</p>

Table B–1 (Cont.) WebUtil Error Reference

Error	Description
WUI-304	<p>An invalid Value was supplied for the name of a system property.</p> <p>Type: User</p> <p>Description: When requesting a system property an invalid property was requested.</p> <p>Resolution: Request a valid Java System property such as user.name. Ensure that the property is in lower case.</p>
WUI-305	<p>Unable to get the user . name property from Java.</p> <p>Type: Internal</p> <p>Description: Due to security restrictions the Bean was unable to get the user name for the current user.</p> <p>Resolution: Ensure the Jar file containing WebUtil is signed.</p>
WUI-306	<p>Security violation occurred while getting property.</p> <p>Type: Internal</p> <p>Description: Due to security restrictions the Bean was unable to get the requested property.</p> <p>Resolution: Ensure the Jar file containing WebUtil is signed.</p>
WUH-400	<p>Error reading console InputStream. Exception: {0}.</p> <p>Type: Internal</p> <p>Description: WebUtil was unable to capture the console output generated by a host command.</p> <p>Resolution: Internal Error - no resolution.</p>
WUH-401	<p>Invalid value ({0}) supplied as a Process ID.</p> <p>Type: User</p> <p>Description: The process ID was not a positive integer.</p> <p>Resolution: Make sure that you correctly obtain a Process ID from WebUtil by running a host command before trying to use it.</p>
WUH-402	<p>Null value supplied as a Process ID.</p> <p>Type: User</p> <p>Description: The process ID supplied was null.</p> <p>Resolution: Make sure that you correctly obtain a Process ID from WebUtil by running a host command before trying to use it.</p>
WUH-403	<p>Invalid value ({0}) supplied as an execution mode.</p> <p>Type: User</p> <p>Description: The execution mode was not an integer between 0 and 2 inclusive.</p> <p>Resolution: Only call WebUtil functions through the supplied APIs</p>

Table B–1 (Cont.) WebUtil Error Reference

Error	Description
WUH-404	<p>Null value supplied for execution mode.</p> <p>Type: User</p> <p>Description: A null value was supplied for the execution mode which should have been 0,1 or 2.</p> <p>Resolution: Only call WebUtil functions through the supplied APIs</p>
WUH-405	<p>Null command supplied.</p> <p>Type: User</p> <p>Description: You tried to execute an empty command string.</p> <p>Resolution: Only call WebUtil functions through the supplied APIs.</p>
WUH-406	<p>Error joining HostCommand Thread; Exception\n{0}.</p> <p>Type: Internal</p> <p>Description: WebUtil was unable to suspend the main PJC Thread while the host command was running.</p> <p>Resolution: Try running the command in callback mode.</p>
WUH-407	<p>OutputSink is null.</p> <p>Type: Internal</p> <p>Description: Could not get the output sink object from the HostCommand.</p> <p>Resolution: This can only happen if the HostCommand object has not yet been executed. Only use the supplied APIs to call WebUtil Functions.</p>
WUH-408	<p>The WebUtil Client code sent back a bad Process ID - Please report this error.</p> <p>Type: Internal</p> <p>Description: A Command completed and dispatched an event to the Form, however, the process handle with the event is incorrect.</p> <p>Resolution: Contact Oracle Support.</p>
WUL-919	<p>Invalid length for PARAM_IN or PARAM_INOUT string parameter.</p> <p>Type: User</p> <p>Description: Maximum length allowed for a PARAM_IN or PARAM_INOUT string parameter was null or not specified.</p> <p>Resolution: Pass the length of the string or VARCHAR2 parameter when used as PARAM_IN or PARAM_INOUT parameter.</p>
WUL-922	<p>Setting parameter properties for bind/rebind or fetch failed.</p> <p>Type: User</p> <p>Description: There is not enough information about the type, in-out specifier, and value of the parameter to be re-bound.</p> <p>Resolution: Old parameter must have correct type, in-out specifier and values required for the new parameter to be rebound.</p>

Table B–1 (Cont.) WebUtil Error Reference

Error	Description
WUL-923	<p>Missing or Invalid value for PARAM_IN or PARAM_INOUT parameter.</p> <p>Type: User</p> <p>Description: PARAM_IN or PARAM_INOUT parameter is passed with null or an invalid value.</p> <p>Resolution: Pass a valid value for PARAM_IN or PARAM_INOUT parameters.</p>
WUL-924	<p>Expected libName funcName to parse.</p> <p>Type: User</p> <p>Description: Library Name or /and Function is null or empty.</p> <p>Resolution: Make sure that library name and C function name to be called are passed correctly with a pipe separating them.</p>
WUL-925	<p>Object returned from object cache is not of type {0}.</p> <p>Type: Internal</p> <p>Description: Object returned from object cache is different from what is expected. For instance CFunc is the expected type, while some other object is returned from the cache.</p> <p>Resolution: Contact Oracle Support.</p>
WUL-926	<p>Handle [{0}] not found in object cache and could not be retrieved.</p> <p>Type: User</p> <p>Description: There is no registered handle for the given function.</p> <p>Resolution: Make sure that you register the function calling WEBUTIL_C_API.Register_Function.</p>
WUL-927	<p>Expected {0} to parse from message string.</p> <p>Type: User</p> <p>Description: Function or parameter or return type is null or passed incorrectly.</p> <p>Resolution: WEBUTIL_C_API.INVOKE_WU using function handle expects non-null function handle and correct return type. Parameter list can be null.</p>
WUL-928	<p>Library {0} not accessible, or does not contain function {1}.</p> <p>Type: User</p> <p>Description: Library could not be opened or it does not contain the required function while calling WEBUTIL_C_API.INVOKE_WU using library name and function name.</p> <p>Resolution: Make sure that library path exists and contains the given function.</p>
WUL-929	<p>Invalid action [{0}] specified in message bundle for parameter properties.</p> <p>Type: Internal</p> <p>Description: C Parameter to be added does not have a valid Bind or Rebind or Fetch specifier.</p> <p>Resolution: Internal error. Contact Oracle Support.</p>

Table B–1 (Cont.) WebUtil Error Reference

Error	Description
WUL-930	Could not select parameter [{1}] of parameter list [{0}]. Type: Internal Description: Internal error. Resolution: Contact Oracle Support.
WUL-931	Could not select parameter list [{0}]. Type: User Description: Could not extract and select the parameter for rebinding or fetching. Resolution: Old parameter must have correct type, in-out specifier and values required for the new parameter to be rebound.
WUL-932	Exception: {0} Type: User Description: ParameterList handle or ParameterHandle for rebinding or ParameterType for binding is null or empty. Resolution: Make sure that ParameterList and ParameterHandle for rebinding or ParameterType for binding are not null.
WUL-933	Cannot add NULL parameter to parameter list. Type: User Description: ParameterList handle is null and hence cannot add a parameter. Resolution: Make sure that ParameterList handle is not null.
WUL-934	Cannot retrieve value from a NULL parameter. Type: User Description: Got a null parameter handle while calling WEBUTIL_C_API.GET_PARAMETER_STRING. Resolution: Pass a correct ParameterHandle to get its value.
WUL-935	Could not select parameter from parameter list. Either the list is NULL, or the parameter handle is out of range. Type: User Description: Could not select parameter from parameter list. Either the list is NULL, or the parameter handle is out of range. Resolution: Make sure that the parameterList and ParameterHandle are passed correctly.
WUL-936	Cannot register function if libName or funcName is NULL [lib:{0},func:{1}]. Type: User Description: Either library name or function name are NULL. Resolution: Make sure that library name and function name are not NULL.
WUL-937	Invalid specifier for InOut qualifier: [{0}]. Type: User Description: InOut specifier is invalid or incorrect. Resolution: InOut specifier must be either PARAM_IN or PARAM_OUT or PARAM_INOUT.

Table B–1 (Cont.) WebUtil Error Reference

Error	Description
WUL-938	Invalid maximum length specifier: [{0}]. Type: User Description: Maximum length is not an acceptable number. Resolution: Maximum length must be a number between 0 and 32768.
WUL-939	Could not add parameter. Type: Internal Description: Internal error. Resolution: Contact Oracle Support.
WUL-940	Number of parameters do not agree between paramList and Array. Type: Internal Description: Internal Error Resolution: Contact Oracle Support.
WUL-941	Returned value is too long. Maximum length allowed for the variable is {0}, but returned length is {1}. Type: User Description: Environment variable string is longer than the maximum length allowed. Resolution: Increase the maximum buffer length to store the environment variable string.
WUS-500	Invalid URL supplied for session time out redirect {0}; Exception: {1}. Type: User Description: A URL supplied as the target for a redirect in case of time out was invalid. Resolution: Correct the URL string.
WUO-700	Unable to create the OLE Server {0}; Exception: {1}. Type: User Description: WebUtil was unable to create an instance of the specified OLE Server Application. Resolution: Check that the OLE Server Application works standalone and the name has been specified correctly.
WUO-701	Unable to release the OLE Object Handle; Exception: {0}. Type: User Description: WebUtil was unable to release the specified OLE Object Handle. Resolution: Check that the correct OLE Object Handle has been specified.
WUO-702	Unable to destroy Argument List; Exception: {0}. Type: User Description: WebUtil was unable to destroy the specified Argument List. Resolution: Check that the correct Argument List Handle has been specified.

Table B–1 (Cont.) WebUtil Error Reference

Error	Description
WUO-703	Unable to add Argument; Exception: {0}. Type: User Description: WebUtil was unable to add the specified Argument to the Argument List. Resolution: Check that the correct Object Handle has been specified for OLE Object type arguments.
WUO-704	Unable to add Argument; Exception: {0}. Type: User Description: WebUtil was unable to add the specified Argument to the Argument List. Resolution: Check that the Argument has been specified correctly.
WUO-705	Unable to invoke Method: {0}; Exception: {1}. Type: User Description: WebUtil was unable to invoke the specified Method. Resolution: Check that the Method and Arguments have been specified correctly
WUO-706	Unable to invoke Method: {0}; Exception: {1}. Type: User Description: WebUtil was unable to invoke the specified Method. Resolution: Check that the Method and Arguments have been specified correctly.
WUO-707	Unable to invoke Method: {0}; Exception: {1}. Type: User Description: WebUtil was unable to invoke the specified Method. Resolution: Check that the Method and Arguments have been specified correctly.
WUO-708	Unable to get Property: {0}; Exception: {1}. Type: User Description: WebUtil was unable to get the value of the specified Property. Resolution: Check that the Property has been specified correctly.
WUO-709	Unable to get Property: {0}; Exception: {1}. Type: User Description: WebUtil was unable to get the value of the specified Property. Resolution: Check that the Property has been specified correctly.
WUO-710	Unable to get Property: {0}; Exception: {1}. Type: User Description: WebUtil was unable to get the value of the specified Property. Resolution: Check that the Property has been specified correctly.

Table B–1 (Cont.) WebUtil Error Reference

Error	Description
WUO-711	Unable to get Property: {0}; Exception: {1}. Type: User Description: WebUtil was unable to get the value of the specified Property. Resolution: Check that the Property has been specified correctly.
WUO-712	Unable to set Property: {0}; Exception: {1}. Type: User Description: WebUtil was unable to set the specified Property to the specified value. Resolution: Check that the Property and value have been specified correctly.
WUO-713	Unable to set Property: {0}; Exception: {1}. Type: User Description: WebUtil was unable to set the specified Property to the specified value. Resolution: Check that the Property and value have been specified correctly.
WUO-714	Unable to get the last OLE Error details; Exception: {1}. Type: Internal Description: WebUtil was unable to get the details of the last OLE Error. Resolution: Contact Oracle Support.
WUO-715	Invalid value: {0} specified for OLE Object or Argument List Handle. Type: User Description: The specified OLE Object or Argument List Handle was not a positive integer. Resolution: Check that the OLE Object or Argument List Handle has been specified correctly.
WUO-716	Null value specified for OLE Object or Argument List Handle. Type: User Description: The specified OLE Object or Argument List Handle was not a positive integer. Resolution: Check that the OLE Object or Argument List Handle has been specified correctly.
WUL-901	Could not create an argument list in the Object Cache. Type: Internal Description: WebUtil was unable to create an argument list for passing to a C function. Resolution: Contact Oracle.
WUL-902	Could not remove argument list from Object Cache. Type: User Description: WebUtil was unable to delete an argument list. Resolution: Ensure that you have not already Destroyed this argument list.

Table B–1 (Cont.) WebUtil Error Reference

Error	Description
WUL-903	Argument could not be added to argument list. Type: User Description: WebUtil was unable to add an argument to an argument list. Resolution: This may be because the argument is NULL - ensure some value is passed.
WUL-904	Supplied handle was not numeric. Type: User Description: An invalid reference to a Function or argument list was passed to WebUtil. Resolution: Only use the supplied APIs to call WebUtil Functions.
WUL-905	Supplied handle was NULL Type: User Description: An invalid reference to a Function or argument list was passed to WebUtil Resolution: Make sure that you create an Argument list or register a function before using a reference to it
WUL-907	Fetching argument value from the argument list failed. Type: User Description: WebUtil was unable to get the value of the specified parameter. Resolution: Make sure that the parameter was defined as OUT or IN OUT.
WUL-908	Function could not be registered. Type: Internal Description: WebUtil was unable to create the C Function Object. Resolution: Consult the exception for more information.
WUL-909	Function could not be deregistered. Type: Internal Description: WebUtil was unable to create the C Function Object. Resolution: Consult the exception for more information.
WUL-910	Invoking the function failed. Type: Internal Description: WebUtil was unable to invoke the C Function. Resolution: Consult the exception for more information.
WUL-911	Unsupported return type {0} for C DLL function. Type: User Description: The return type requested was not valid. Resolution: Only use the WebUtil APIs to call this function.
WUL-912	c_char_ptr should have underlying Java instance of CMalloc. Type: Internal Description: Internal Error Resolution: Contact Oracle.

Table B-1 (Cont.) WebUtil Error Reference

Error	Description
WUL-913	<p>Underlying C DLL function overwrote Java CMalloc object fence post.</p> <p>Type: User</p> <p>Description: The C function that you have called has written more data to one or more of the parameters defined as OUT or IN OUT than the declared length of that variable.</p> <p>Resolution: Consult the API of the C function being called and make sure that your PL/SQL variables and max length definitions are of sufficient size.</p>
WUL-914	<p>Attempting to read unsupported IN/INOUT type parameter.</p> <p>Type: User</p> <p>Description: The parameter is not of a valid type.</p> <p>Resolution: Only use the WebUtil APIs to call this function. Ensure that you pass a valid constant for the parameter type.</p>
WUL-915	<p>Supplied value of bound variable is longer than supplied maximum length.</p> <p>Type: User</p> <p>Description: The maximum length defined for a parameter should be long enough to hold its value.</p> <p>Resolution: Correct the maximum length defined in the code.</p>
WUL-916	<p>Unsupported parameter type for C DLL function.</p> <p>Type: User</p> <p>Description: The parameter is not of a valid type.</p> <p>Resolution: Only use the WebUtil APIs to call this function. Ensure that you pass a valid constant for the parameter type.</p>
WUL-917	<p>Generic c_ptr cannot be used as OUT or INOUT parameter. Only as IN parameter or return value.</p> <p>Type: User</p> <p>Description: The C pointer type cannot be used as an argument type for OUT arguments.</p> <p>Resolution: This is a restriction</p>
WUL-918	<p>Portability error using JNI: C_PTR_LEN does not match underlying host C integral type.</p> <p>Type: User</p> <p>Description: The C_API functions in WebUtil are only supported when calling 32 bit Windows clients.</p> <p>Resolution: These functions are only available on 32 bit windows clients.</p>

Index

A

access control, 3-5
 defining read and write directives, 3-5

C

client server parity APIs, 4-1
 Client_Get_File_name, 4-1
 Client_Host, 4-1
 Client_Image, 4-1
 Client_OLE2, 4-1
 Client_Text_IO, 4-1
 Client_Tool_Env, 4-1
CLIENT_TEXT_IO
 about, 1-2
configuration entries, 3-1
configuring
 database, 2-3
 environment file, 2-5
 formsweb.cfg, 2-4
 HTTP server, 2-3
 servlet directives, 2-4
 webutil.cfg, 2-6
 webutil.properties, 2-6
create_webutil_db.sql
 database packages used by
 DBMS_LOB, 2-3
 UTL_ENCODE, 2-3
 UTL_RAW, 2-3

D

D2KWUTIL functions, 4-2
 Create_Registry_Key, 4-3
 Delete_Registry_Key, 4-3
 Get_Computer_Name, 4-3
 Get_Environment_String, 4-3
 Get_Net_Connection, 4-3
 Get_Temp_Directory, 4-3
 Get_Windows_Directory, 4-3
 Get_Windows_Username, 4-3
 Get_working_Directory, 4-3
 Read_INI_File, 4-3
 Read_Registry, 4-3
 Win_API_Environment, 4-2

 Write_INI_file, 4-3
 Write_Registry, 4-3
 Write_RegistryEx, 4-3
DelimStr
 about, 4-12
design time
 viewing PJC, 5-2
design time components
 webutil.olb, 2-2
 webutil.pll, 2-2
diagnostic tracing, 7-2
directives
 read and write, 3-5

E

error handling, 7-1
error reference
 about, B-1
 WUB-600, B-10
 WUB-601, B-10
 WUB-602, B-11
 WUB-603, B-11
 WUC-1, B-1
 WUC-10, B-3
 WUC-11, B-3
 WUC-12, B-3
 WUC-13, B-3
 WUC-14, B-3
 WUC-15, B-3
 WUC-16, B-3
 WUC-17, B-4
 WUC-18, B-4
 WUC-19, B-4
 WUC-2, B-1
 WUC-20, B-4
 WUC-21, B-4
 WUC-22, B-4
 WUC-23, B-4
 WUC-24, B-5
 WUC-25, B-5
 WUC-26, B-5
 WUC-27, B-5
 WUC-28, B-5
 WUC-3, B-1
 WUC-4, B-2

WUC-5,	B-2	WUL-938,	B-17
WUC-6,	B-2	WUL-939,	B-17
WUC-7,	B-2	WUL-940,	B-17
WUC-8,	B-2	WUL-941,	B-17
WUC-9,	B-2	WUO-700,	B-17
WUF-200,	B-10	WUO-701,	B-17
WUF-201,	B-11	WUO-702,	B-17
WUF-202,	B-11	WUO-703,	B-18
WUF-203,	B-11	WUO-704,	B-18
WUF-204,	B-11	WUO-705,	B-18
WUF-205,	B-11	WUO-706,	B-18
WUF-206,	B-11	WUO-707,	B-18
WUF-207,	B-12	WUO-708,	B-18
WUF-208,	B-12	WUO-709,	B-18
WUF-209,	B-12	WUO-710,	B-18
WUF-210,	B-12	WUO-711,	B-19
WUF-211,	B-12	WUO-712,	B-19
WUF-212,	B-12	WUO-713,	B-19
WUH-400,	B-13	WUO-714,	B-19
WUH-401,	B-13	WUO-715,	B-19
WUH-402,	B-13	WUO-716,	B-19
WUH-403,	B-13	WUS-500,	B-17
WUH-404,	B-14	WUT-100,	B-5
WUH-405,	B-14	WUT-101,	B-5
WUH-406,	B-14	WUT-102,	B-5
WUH-407,	B-14	WUT-103,	B-6
WUH-408,	B-14	WUT-104,	B-6
WUI-303,	B-12	WUT-105,	B-6
WUI-304,	B-13	WUT-106,	B-6
WUI-305,	B-13	WUT-107,	B-6
WUI-306,	B-13	WUT-108,	B-6
WUL-901,	B-19	WUT-109,	B-6
WUL-902,	B-19	WUT-110,	B-6
WUL-903,	B-20	WUT-111,	B-7
WUL-904,	B-20	WUT-113,	B-7
WUL-905,	B-20	WUT-114,	B-7
WUL-907,	B-20	WUT-115,	B-7
WUL-908,	B-20	WUT-116,	B-7
WUL-909,	B-20, B-21	WUT-117,	B-7
WUL-912,	B-20	WUT-118,	B-7
WUL-916,	B-21	WUT-119,	B-8
WUL-917,	B-21	WUT-120,	B-8
WUL-918,	B-21	WUT-121,	B-8
WUL-919,	B-14	WUT-122,	B-8
WUL-922,	B-14	WUT-123,	B-8
WUL-923,	B-15	WUT-124,	B-8
WUL-924,	B-15	WUT-125,	B-9
WUL-925,	B-15	WUT-126,	B-9
WUL-926,	B-15	WUT-127,	B-9
WUL-927,	B-15	WUT-128,	B-9
WUL-928,	B-15	WUT-129,	B-9
WUL-929,	B-15	WUT-130,	B-9
WUL-930,	B-16	WUT-131,	B-9
WUL-931,	B-16	WUT-132,	B-10
WUL-932,	B-16	WUT-133,	B-10
WUL-933,	B-16	WUT-134,	B-10
WUL-934,	B-16	WUT-135,	B-10
WUL-935,	B-16		
WUL-936,	B-16		
WUL-937,	B-16		

F

- file upload and download options, 3-5
 - enabling transfer, 3-5
 - master switches
 - transfer.appsrv.enabled, 3-5
 - transfer.database.enabled, 3-5
- formsweb.cfg
 - about, 3-1
- formsweb.cfg File
 - required configurations, 3-1
 - EnvFile, 3-1
 - WebUtilArchive, 3-1

H

- HOST commands
 - running, 4-2
- HTML templates, defining, 2-4

I

- internal APIs
 - about, 4-12

J

- JACOB
 - source code and licensing details, 2-2

L

- library definition string, 3-4
- logging
 - connection information, 7-1

M

- Microsoft Office
 - interface changes, 5-4

O

- Object Group
 - subclassing for WebUtil, 1-2
- OLE commands
 - example usage, 5-2
 - using, 5-2

P

- public functions
 - list, 4-3

R

- required objects
 - adding to applications, 5-1
 - subclassing webutil.olb, 5-1
 - webutil.pll library, 5-1
- runtime

- install checklist, A-1
- runtime components
 - frmwebutil.jar, 2-2
 - webutil.pll, 2-2

S

- security
 - Security Overview, 2-7
 - TrustedDomains.txt, 2-7
 - TrustedDomains.txt,sample, 2-7
 - WebUtil and the Internet, 2-7
 - WebUtil and the Intranet, 2-8
 - WebUtilTrustInternal, 2-8
- Show_WebUtil_Information
 - about, 4-12

T

- TEXT_IO package
 - about, 1-2

U

- utility functions
 - about, 4-11

V

- virtual directories, defining, 2-3

W

- WebUtil
 - adding code, 5-2
 - restriction, 5-2
 - components, 2-2
 - directory structure, 2-2, 2-3
 - goals, 1-1
 - installation prerequisites, 2-1
 - optimizing performance, 6-1
 - technology description, 1-2
 - viewing canvas, 5-2
- WebUtil ClientInfo package, 4-4
 - Get_Date_Time, 4-4
 - Get_file_Separator, 4-4
 - Get_Host_Name, 4-4
 - Get_IP_Address, 4-4
 - Get_Java_Version, 4-4
 - Get_Language, 4-4
 - Get_Operating_System, 4-4
 - Get_Path_Separator, 4-4
 - Get_System_Property, 4-4
 - Get_Time_Zone, 4-4
 - Get_User_Name, 4-4
- WebUtil components
 - design time, 2-2
 - runtime, 2-2
- WebUtil Installation Steps
 - signing Jar files, 2-6
- WebUtil library

- subclassing, 1-2
- WebUtil_Browser
 - about, 4-10
 - BrowserMessage, 4-10
 - GetAppletParameter, 4-10
 - GetAppletSize, 4-10
 - ShowMenuBar, 4-10
 - ShowStatusBar, 4-10
- WebUtil_C_API, about, 4-4
- WebUtil_Core
 - about, 4-11
 - IsError, 4-11
- WEBUTIL_DB, 2-3
- WebUtil_File
 - about, 4-5
 - Copy_File, 4-5
 - Create_Directory, 4-5
 - Delete_File, 4-5
 - Directory_Filtered_List, 4-5
 - Directory_List, 4-5
 - Directory_Root_List, 4-5
 - Directory_Selection_Dialog, 4-5
 - File_Exists, 4-5
 - File_Is_Directory, 4-5
 - File_Is_Hidden, 4-5
 - File_Is_Readable, 4-5
 - File_Is_Writable, 4-5
 - File_Multi_Selection_Dialog, 4-6
 - File_Open_Dialog, 4-6
 - File_Save_Dialog, 4-6
 - File_Selection_Dialog, 4-6
 - File_Size, 4-5
 - Get_file_Separator, 4-6
 - Get_Path_Separator, 4-6
 - Rename_File, 4-5
- WEBUTIL_FILE_TRANSFER, 2-3
- WebUtil_File_Transfer package
 - about, 4-6
 - AS_To_Client, 4-8
 - AS_To_Client_With_Progress, 4-8
 - Asynchronous_Upload_Success, 4-8
 - Client_To_AS, 4-7
 - Client_To_AS_With_Progress, 4-7
 - Client_To_DB, 4-7
 - Client_To_DB_With_Progress, 4-7
 - DB_To_Client, 4-7
 - DB_To_Client_With_Progress, 4-7
 - Get_Work_Area, 4-8
 - In_Progress, 4-8
 - Is_AS_Readable, 4-8
 - Is_AS_Writable, 4-8
 - URL_To_Client, 4-6
 - URL_To_Client_With_Progress, 4-6
- Webutil_Host
 - about, 4-8
 - Blocking, 4-9
 - Equals, 4-9
 - Get_Callback_Process, 4-9
 - Get_Return_Code, 4-9
 - Get_Standard_Error, 4-9
 - Get_Standard_Output, 4-9
 - Host, 4-9
 - Id_Null, 4-9
 - NonBlocking, 4-9
 - NonBlocking_With_Callback, 4-9
 - Release_Process, 4-9
 - Terminate_Process, 4-9
- WebUtil_SeparateFrame
 - about, 4-10
 - AllowResize, 4-11
 - CenterMDI, 4-11
 - GetSeparateFrameSize, 4-11
 - IsSeparateFrame, 4-11
 - SetIcon, 4-11
 - SetTitle, 4-11
 - ShowMenuBar, 4-11
 - ShowStatusBar, 4-11
- WebUtil_Session
 - about, 4-9
 - Disable_Redirect_On_TimeOut, 4-10
 - Enable_Redirect_On_TimeOut, 4-10
- WebUtil_Util
 - about, 4-12
- webutil.cfg, 3-3
 - installation options, 3-3
 - codes, 3-4
 - library definition string
 - example, 3-4
 - name, 3-4
 - ShowDownLoadDialog, 3-4
 - size, 3-4
 - version, 3-4
 - logging options
 - logging.connections, 3-3
 - logging.enabled, 3-3
 - logging.erroronly, 3-3
 - logging.file, 3-3
- work area, 3-6
- WUC-15 error
 - resolving, 5-1
- WUT-112, B-7