

# Setting the default text encoding for Eclipse

## Overview

Creating a new WO Application/WOnder application project in today's version of WOLips generates the following warning:

WOO Encoding type UTF-8 doesn't match component MacRoman (Main.woo)

Changing the encoding type in the .woo and in the .html file manually does not get rid of the warning. There are a couple of ways to go about this:

## Change the encoding for your entire Workbench

In Eclipse, go to Preferences>General>Workspace and select UTF-8 as the Text File Encoding. This should set the encoding for all the resources in your workspace. Any components you create from now on using the default encoding should all match.

## Note from Eclipse Help pages about text encoding

*Given the ability to use the platform in a networked team environment, it's certainly possible that users will want to work with text files that use a different encoding scheme than their native encoding scheme so that they can easily interchange files with another team. For this reason, the workbench defines its own encoding profile that is specified by the user in the Preferences dialog. Users may choose from the available encoding choices in the General > Workspace preference page or type in their own encoding. Plugins that interpret text files, such as editors and builders, should consult the workbench encoding preference rather than assume that the installed operating system encoding is in use.*

Currently, creating a new project will generate the above warning in the Main component, but that can be fixed by doing the following:

## Change the encoding for a resource (project, folder or file)

Right (or Control) click on the resource in the WO Package Explorer and select the Properties option at the bottom of the contextual menu. Go to Resource>Text File Encoding. You should see "Inherited from container (UTF-8)" if you've made the change above, otherwise it will be (MacRoman). Select Other - UTF-8 and Apply to make the warning go away.

## Original note about text encoding in WOLips from Q

*It should now be possible to change the encoding type of a component using just the eclipse "Resource->Text file encoding" context menu on the component in the WO Package Explorer. This will automatically edit the woo file, or create one if necessary, and explicitly set the encoding type of the .html and .wod files. This process requires that you use a valid Java charset instead of the old NS\* obj-c style charset name.*

*This change also means that the encoding type of a component can be inherited from the parent folder. So you can manage the encoding type of all components within a folder by changing the default encoding type of the parent folder (assuming the contained components still inherit their encoding type, the default behaviour).*

*The WOComponent creation wizard hasn't yet been updated to reflect these changes, so a newly created component will still be created the same as before, and may have warnings about the character encoding of the woo file not matching that of the component, this will change in the future.*

You may also want to refer to this page for using UTF-8 in your WebObjects application: [EOF-Using EOF-Localization and Internationalization](#)