Auto Cemli File Upload for OMCS ACE Cemli Patcher.

# Disclaimer

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# Introduction

The Oracle Cemli ACE patcher has been developed to allow users to upload files and create patches in OMCS environments. Uploading files manually is a tedious job when more than a few files are required to be uploaded. OMCS have created the possibility to upload a zip file into the ACE Cemli portal but this requires a manifest file to be created. If this is a manual process it is error prone.

We have created a script to export the changes out of our file system and automatically create a zip file and manifest file.

# Scope

The process has been developed using Atlassian Source tree and GIT. We are using GITFLOW to structure changes. We don’t believe Source Tree is necessary if a Git Bash shell can be run. The process has been developed to work with the OMCS ACE Cemli patcher. This code has been developed on Windows.

Code is available at

<https://github.com/alantelford/Oracle-ACE-Portal>

# Instructions

* Make sure your code is backed up in case of any problems.
* Download the create\_zip\_file.sh script.
* Copy the script a directory e.g.
  + C:\git\_scripts

This will be references as SCRIPT\_HOME in the script

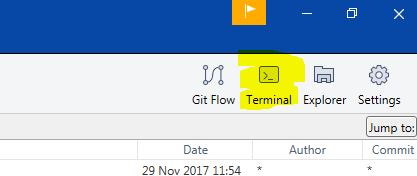
* Create a directory for the zip files and manifest to go into. This will be set to **TARGET\_DIR** in the script
  + e.g. c:\git\_output

Edit the file create\_zip\_file.sh and set the variables appropriately. These will be UNIX style directory strings

* TARGET\_DIR
* REPO\_HOME – this is the GIT repository home directory where code will be extracted from
* SCRIPT\_HOME

Identify a GIT feature to migrate. Ensure changes are committed and pushed to Origin

Start a GIT bash shell. From Source tree use the Terminal option



Use the command **git branch** to list out current features.

Change directory to the **SCRIPT\_HOME**

Execute the **create\_zip\_file.sh**

The script has two parameters

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Optional** | **Description** |
| 1 | No | Feature name to extract eg. feature/my\_new\_development |
| 2 | Yes | Comment to be inserted into the description field in the manifest file. You can use this to tag all the files which are uploaded and search for them by the description. This is handy for finding files in the cemli patcher |

### Result

Check in the **TARGET\_DIR\archive\feature**

The feature extracted should be listed out and within that directory will be found a zip file with the code which is altered in the feature and a manifest file. Which can be posted into the cemli patcher.

## Options

Putting a line in the file which is in the form

CEMLI\_FILE\_TYPE=xxx

Allows the user to specify the type of file to be uploaded. E.g. c\_pks is a custom package. So the user does not need to choose the file type after it has been loaded up into the cemli patcher.

c\_pks and c\_pkb are automatically substituted in for files of pks and pkb as OMCS requires packages to be loaded into the Business Online Interface by default.

Similarly files with the .prog extension are defaulted to type shell (for UNIX shell script copy only)

If the CEMLI\_FILE\_TYPE of **nopatch** is used the files will not be copied across into the cemli patch.

### Header Checking

Where the file extension is not in fmb, rdf or class the file header is checked to ensure the file name is contained in the line. This is a basic file validation to avoid upload errors.