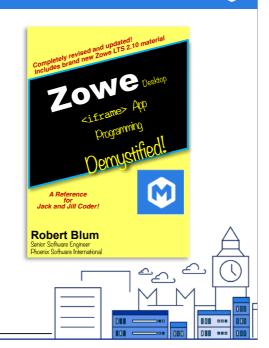


I'm Robert Blum, a developer at Phoenix Software International.

Experience It, Don't Transcribe It



- Think of this as a 10 minute book discussion
 - The conference makes this presentation available for download, as does Phoenix Software International
 - Includes GitHub repository links
 - Includes Zowe Documentation links
 - My talk covers only highlights
 - Take notes for questions or important concepts
 - Refer to presentation later to do actual work
 - If you have the video, pause between slides so as to read the details



In this talk, I want to demystify taking a web app

making it show up on the Zowe desktop.

It's actually quite easy!

It's just that the documentation is reference material, **not** a tutorial.

I spent weeks figuring out what turned out to be pretty trivial.

I'm not a z/OS Unix programmer. I'm a web developer AND I sometimes use the mainframe.

I've analyzed what's needed, and what you can ignore -- so you

don't

have to.

The slides, which are available from the conference, are filled with details.

Think of the slides as a book.

This talk is the Cliffs Notes version of that book.

Lessons Learned



- Your web app on Zowe Desktop without learning the Zowe API
- · Minimum you need to know to succeed
 - z/OS unix command essentials
 - GitHub and cloning
 - Finding and cycling the Zowe desktop STC job
 - Finding necessary paths
 - The zwe command and installation
 - What to modify or remove
- Learn what I learned the easy way
 - No reading reference manuals written for the already expert
 - No experimenting to figure out what you missed



In this talk, I'm going to review what you need and nothing more, including:

Listing z/OS Unix commands

Using GitHub and cloning projects
Finding and cycling the Zowe desktop STC job
Finding the necessary paths needed for commands
Explaining the z-w-e command

Explaining what to modify and what to remove

I'm also giving you links for further reading.

Why is the Zowe Desktop Attractive?



- Securely available anywhere corporate access to a web address is available
- Strategic direction for many z/OS clients
- Management may want (or policy may dictate) existing web apps be available on the new platform



Here are reasons you'll port to Zowe. Consider them later.

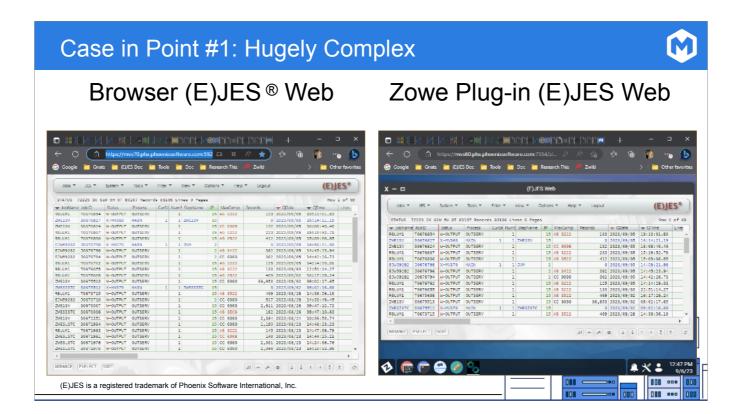
Reduce, Reuse, Recycle



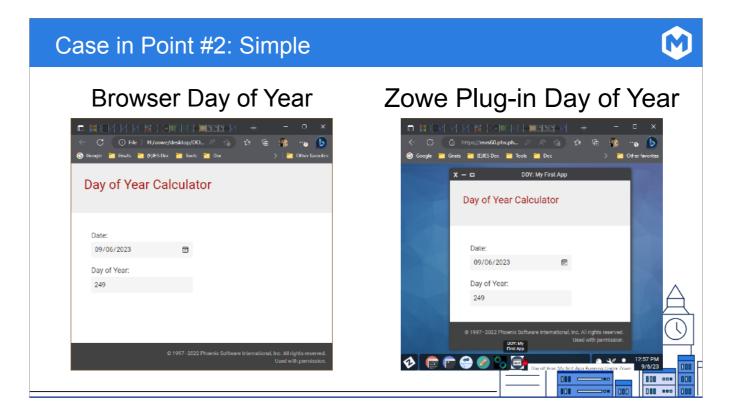
- "Learn how reducing, reusing, and recycling can help you, your community, and the environment by saving money, energy, and natural resources." EPA website
 - Getting your program running in a flash saves:
 - Your company money
 - Energy powering your computers
 - · Resources (i.e., you) for more difficult problems.
- · Why do this?
 - Existing apps are tested and iterated for the corporate environment
 - Don't reinvent the wheel: No need to implement for a new API if old one provides solution
 - The less time spent porting, the more resources for other projects
 - No need to program in Typescript; use what you used for your web app



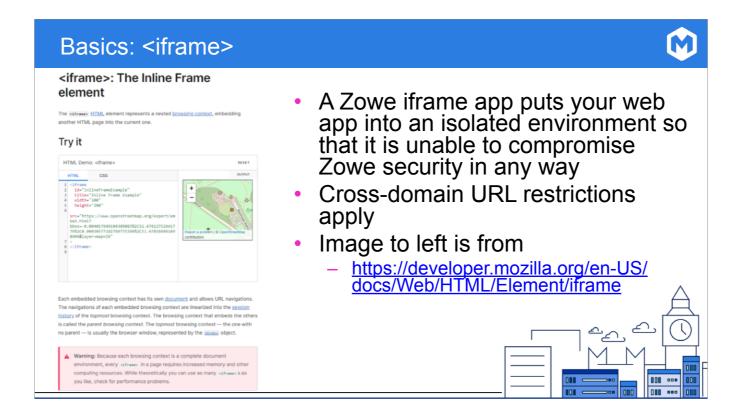
This whole slide boils down to, "Don't reinvent the wheel." Well, if you want to... you'll have to do that afterwards.



Case in point. On the left is a very complex web app, and on the right it is running on the Zowe desktop.



Second case in point. It can be any web app, even a simple one.



Why an iframe? By putting your web app inside an iframe tag, you isolate it from the Zowe desktop with **no** chance of compromising Zowe security.

z/OS Unix Essentials



- Unix primer for the non-cognoscenti (me)
 Upper and lowercase matter; get it correct!

 - Ask a colleague how to copy and paste in your Linux terminal shell
 - Linux commands to know:
 - cat file Display a file
 - cd Change directory, just like in PowerShell or the DOS prompt
 - chtag Change z/OS tag attributes to iso8859-1 or binary
 - find root-path -name "text" -exec Is -ld {} \ Find text in a directory and subdirectories
 - Is List directory
 - Is -laF Full information
 - Is -laT Tag information
 - man command Display manual page, documentation for a command
 - more file Display a file one screen at a time, using q to quit
 - pwd Display current directory path
 - More at:
 - https://github.com/IBMRedbooks/USSCheatSheet/blob/main/USSCheatSheet.md





If you are new to z/OS Unix, use this cheat sheet. You won't need more than this.

Git Essentials



- Popular multiplatform version control program
 - Samples stored in GitHub repositories (PSI's and Zowe's)
 - Install GIT on your machine (https://git-scm.com)
 - Install Nodejs (https://nodejs.org/)
- Clone (download project to your file system)
 - git clone https://github.com/phoenixsoftware/day-of-year-zowe-desktop-plugin
 git clone https://github.com/zowe/sample-iframe-app

 - git clone https://github.com/phoenixsoftware/ejes-web-zowe-desktop-plugin
- · You are working with your own copy of their repositories
 - Delete the hidden .git directory and other things that start with .git
 - What remains are your project files



Install Git, if you haven't already. Download these three projects, which I'll be discussing in this talk. The process is known as cloning.

Optional: Initialize a New Git Repository



- In the root of your project
 - git init
 - git add . There is a space before the period
 - git status
 - git commit -m "Initial commit of iframe app source"
- Tutorial
 - https://kbroman.org/github_tutorial/ pages/init.html

If you want version control for your custom iframe project, read this slide later.

Find Essential Zowe Information



000 000

- Find the Zowe STC job in JES
 - ZWESLSTC or ZWE1SV if vanilla parameters used
- Find ZWEL0023I in the job output
 - Provides full path on zFS
 - Typical config path: /local/zowe/zowe.yaml
- · Browse the zowe.yaml file.
 - Record these directories
 - runtimeDirectory: Where Zowe is installed, including all preinstalled apps
 - workspaceDirectory: Root where your apps and data can be found
 - extensionDirectory: Symbolic link to installed apps
 - logDirectory: Log directory where components write error information

```
Deline Do Holler on Companyanica, 19849, 198124111 18889

Deline Do Holler on Companyanica, 19849, 198124111 18889

Deline Do Holler on Companyanica, 19849, 19812411 18889

John Land Companyanica, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 19849, 1
```

You need to know where Zowe keeps its configuration file. This information is needed to cycle Zowe and install apps. Find the Zowe STC job in the JES queues. It will probably be ZWESLSTC or ZWE1SV. Find the ZWEL0023I message in the job output. Browse the yaml file indicated.

The zowe.yaml file has the directories listed on this slide. Write them down. $\label{eq:control_eq}$

This is what the yaml file looks like AND some of the information you can find there. Note the location of the directories.

zwe Command Primer



- https://docs.zowe.org/stable/appendix/zwe_server_command_reference/zwe/
- cd to and execute all zwe commands from runtimeDirectory/bin
 - Cycle Zowe (enter both commands)
 - zwe stop -c /path/zowe.yaml
 - zwe start -c /path/zowe.yaml
 - Install the app
 - zwe components install -c /path/zowe.yaml -o /path/myapp
 - Run after changing app (recycling Zowe only if metadata changed)
 - zwe components upgrade -c /path/zowe.yaml -o /path/myapp
 - Uninstall app completely
 - zwe components uninstall -c /path/zowe.yaml -o /path/myapp

Note: /path/myapp is the path to your app's directory that contains manifest.yaml



You will use the z-w-e command to do all the work.
You must change the directory to the highlighted directory.
This slide describes the minimum five commands you'll need:
stop and start to cycle Zowe,

install, uninstall, and upgrade to maintain Zowe.

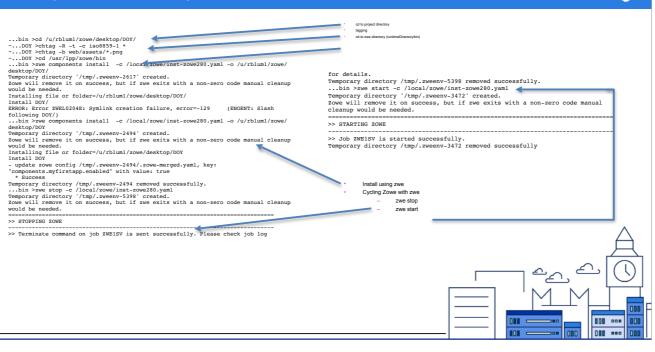
I show the syntax, but Zowe-style help is available at the command line.

To illustrate the last slide better, here's a z/OS Unix session. At the top-left, I've pointed out tagging, which we'll get to later. Below that, I've installed my sample app. Since it is an install, I finish by showing how to cycle Zowe by stopping starting it.

Sample z/OS Output

>> STOPPING ZOWE





Tagging Files Tutorial



- Files on z/OS must be properly tagged
 - Editing files, especially the icon file, may remove or modify tags
 - Tag the icon file as binary
 - Tag all text files as iso8859-1
 - Use the Zowe desktop Editor program to tag files
 - Alternatively use a command in project root (scriptable)
 - chtag -R -t -c iso8859-1 * There is a space before the asterisk
 - chtag -b web/assets/*.png
 - Check result
 - Is -laT



Tagging is important, because if you get it wrong Zowe will see garbage and the desktop may fail to come up.

Sample z/OS Output (tags)



```
~...DOY >pwd
/u/rblum1/zowe/desktop/DOY
~...DOY >la -laT
total 96
         drwxr-xr-x 4 RBLUM1 DEV 8192 Sep 5 14:07 /
         drwxr-xr-x 5 RBLUM1 DEV 8192 Sep 5 14:07 ../
t ISO8859-1 T=on -rwxr-xr-x 1 RBLUM1 DEV 618 Sep 5 15:26 manifest.yaml*
t ISO8859-1 T=on -rwxr-xr-x 1 RBLUM1 DEV 1243 Sep 5 15:40 pluginDefinition.json*
         drwxr-xr-x 2 RBLUM1 DEV 8192 Sep 5 14:07 schemas/
         drwxr-xr-x 6 RBLUM1 DEV 8192 Sep 5 14:07 web/
~...DOY >cd web/assets/
~...assets >la -laT
          drwxr-xr-x 2 RBLUM1 DEV 8192 Sep 5 14:07 ./
         drwxr-xr-x 6 RBLUM1 DEV 8192 Sep 5 14:07 ../
- untagged T=off -rwxr-xr-x 1 RBLUM1 DEV 4621 Sep 16 2022-tcon.png*
~...assets >chtag -b icon.png
~...assets >la -laT
          drwxr-xr-x 2 RBLUM1 DEV 8192 Sep 5 14:07 /
          drwxr-xr-x 6 RBLUM1 DEV 8192 Sep 5 14:07 ../
b binary T=off -rwxr-xr-x 1 RBLUM1 DEV 4621 Sep 16 2022 icon.png*
```

Because I changed icon.png, I had to retag the file.



This is what tags look like in a z/OS Unix session and when they aren't there.

No excuse to worry what this looks like, now.

Install and Run the Day of Year Sample



- We've learned enough to run the sample app on the Zowe desktop
- Procedure summary:
 - Tag files
 - Install using zwe
 - Cycle Zowe using zwe
 - Run the app
 - Uninstall the app because we plan to customize its identifying metadata

The Day of the Year app project comes ready to install. Here's a summary of the procedure we will follow. It's really pretty simple.

What we're going to do is tag files, install using z-w-e, cycle Zowe, run the app,

then uninstall it

Install



- Project directory must be accessible to Zowe
- Change directory to the zwe directory discussed on a prior slide
 - cd runtimeDirectory/bin
- Install the app
 - zwe components install -c config -o plugin
 - · config is the fully-qualified path to the previously discussed zowe.yaml file
 - plugin is the path to the directory containing your manifest.yaml file
- Cycle Zowe
 - zwe stop -c config
 - zwe start -c config
- · Logon to Zowe and find the app in the Zowe start menu





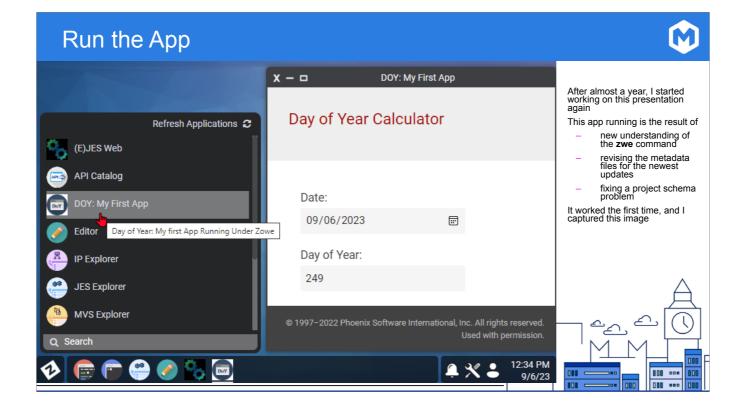




Change to the z-w-e command directory as demonstrated previously. Install the app using the **z-w-e components install** command. Cycle Zowe.

AND

We run the new app from the Zowe start menu.



AND following all the procedures I've just outlined. I hadn't memorized the material. I found the job, found the yaml file, got the paths, then installed my sample app. I captured this image when it ran the first time.

I was a happy camper.

Believe it or not, I ended up reading this presentation

Uninstall



- Project directory must be accessible to Zowe
- · Change directory to the zwe directory discussed on a prior slide
 - cd runtimeDirectory/bin
- Uninstall the app
 - zwe components uninstall -c config -o plugin
 - · config is the fully-qualified path to the previously discussed zowe.yaml file
 - plugin is the path to the <u>directory</u> containing your manifest.yaml file
- Cycle Zowe to remove Zowe's metadata references
 - zwe stop -c config
 - zwe start -c config
- · App should not appear in the start menu



Because metadata identifies the app to Zowe,

if you are planning on modifying the metadata, you must uninstall the app.

Think of it like installing Robert.exe on Windows.

If I rename Robert.exe to JoeWinchester.exe, Windows will no longer be able to uninstall it.

You're welcome to keep the Day of the Year app, of course!

Just copy the project to a new directory and change the metadata...

Customization Overview



- Applicable to all Zowe iframe apps
- Tips:
 - Customizing Metadata files and web files are related but separate topics
 - You may wish to customize only the metadata before installing, then subsequently customize the web files
 - This minimizes the errors you may have to correct at one time
 - Metadata changes are visible:
 - On the start menu
 - By looking at the properties of your app
 - Metadata errors will display in the Zowe STC job output
 - Learn at your own rate



A Zowe iframe **project** consists of two sets of files:
Metadata, which we'll cover next, and web app files.
I suggest modifying the metadata first rather than **both** it **and** the web app files.
Less problems to troubleshoot that way.
Changes to metadata show up in the start menu **and** app properties.

Essential YAML File Notation



The following is valid YAML

```
schemas:
    configs: schemas/trivial-schema.json
schemas.configs: schemas/trivial-schema.json
```

- Lines 1 and 2 above are the same as line 3 alone
- Whitespace indentation is meaningful and signifies levels
 - Inserting a line 3 of "enable: true" is the same as specifying "schemas.enable: true".
- · Zowe documentation uses dot notation for brevity, as do I
- I suggest you use the same notation style you find in the Zowe files (lines 1 and 2) for consistency
- · This link is one of many YAML primers you can find
 - https://getopentest.org/reference/yaml-primer.html



I use YAML dot-notation in my slides.
AS DOES
Zowe documentation.
This slide explains the little bit about YAML that you need to understand Line 3 of the illustration is "dot-notation"
AND
is the same as lines 1 and 2.

Mind that indentation is meaningful.

Metadata vs Web App



- The iframe app bundle consists of metadata and your web app
- The zwe command installs and manages both
- Your metadata
 - Configures Zowe to use your app
 - It identifies the app to Zowe, so you need to uninstall before modifying the metadata that identifies the app to Zowe
 - You must cycle Zowe to see changes
- · Your web app is
 - A web app!
 - You must refresh the Zowe desktop in the browser to see changes, exactly like any other web app





Metadata describes your app to Zowe. You have to cycle Zowe anytime you change metadata. I hope you know what a web app is... NEXT SLIDE

Metadata vs Web App



- The iframe app bundle consists of metadata and your web app
- The **zwe** command installs and manages both
- Your metadata
 - Configures Zowe to use your app
 - It identifies the app to Zowe, so you need to uninstall before modifying the metadata that identifies the app to Zowe
 - You must cycle Zowe to see changes
- Your web app is
 - A web app!
 - You must refresh the Zowe desktop in the browser to see changes, exactly like any other web app





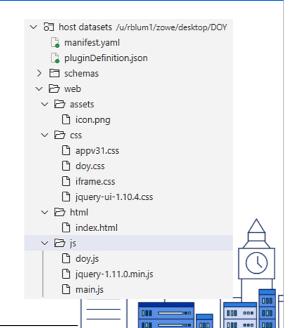


I've highlighted this line because it isn't obvious. Your Zowe iframe app is just a web app.

Day of Year File Layout



- Metadata
 - manifest.yaml
 - pluginDefinition.json
 - schemas (folder)
- Canonical web app (may be customized)
 - icon.png
 - iframe.css
 - index.html
 - main.js
- Supporting web app files
 - Everything else you add
 - Note that I've added jquery files for my own use
 - The doy.js is the core Javascript functionality of my app
- · This is the layout used by Zowe
 - Use it, unless you have a good reason not to
 - I keep my repository files on a different filesystem
 - I edit on the host, but use Git and GitHub on Windows
 - I need to synchronize the filesystems
 - Remember you may need to tag files



This slide explains the file layout of the iframe app.
Unless you have a good reason NOT TO, stick to this layout.

Managing Metadata



- The repositories I've provided are already customized
 - Use this information to understand and customize your app
 - Use zwe stop and zwe start any time you modify metadata
- manifest.yaml
 - "The Zowe server component manifest file defines the name and purpose of the component. It also provides information about how this component should be installed, configured, and started." From https://docs.zowe.org/stable/appendix/server-component-manifest/
 - You need to change only: name, id, version, and title
 - Modifying title, description, and repository is recommended
 - The property **schema.configs** is <u>critical</u>
 - It must point to schema/trival-schema.json
 - · It causes any valid JSON in pluginDefinition.json to validate without error
 - · Validation errors prevent app installation or Zowe startup
 - . The schemas folder appears in the DOY repository and other sample repositories for this reason



Anytime you modify metadata, you need to run z-w-e stop and z-w-e start.

Manifest.yaml describes your app to Zowe.
Here are the files you'll change.
Do not change the value of schema.configs.

If you do, chances are your app won't validate and the desktop won't come up.

Managing Metadata continued



- pluginDefinition.json
 - This file describes an application Plugin to the Zowe Application Server. From https://docs.zowe.org/v2.8.x/extend/extend-desktop/mvd-plugindefandstruct/
 - You need to change only these properties:
 - identifier
 - pluginVersion (zoweMajor.zoweMinor.yourVersion, e.g., 2.10.0906)
 - webContent.launchdefinition.pluginShortNameKey (short tooltip text)
 - webContent.launchdefinition.pluginShortNameDefault
 - webContent.descriptionkey
 - webContent.descriptionDefault (long tooltip text)
 - Depending on the sample, you'll probably want to remove the **dataservices** node
 - I'll mention more about this in advanced topics

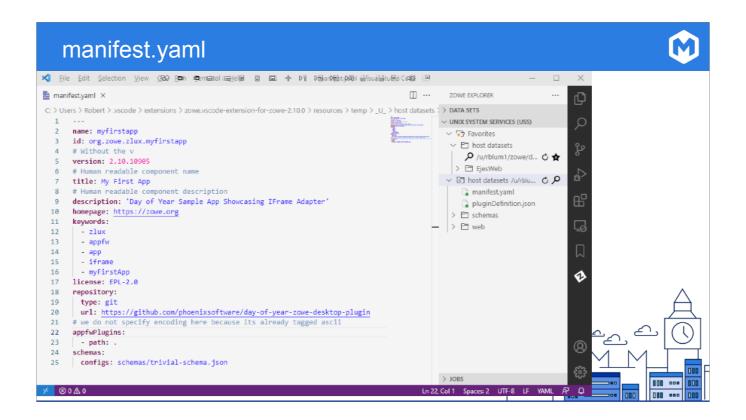


pluginDefinition.JSON describes the app and its properties to Zowe. These two slides highlight a list of things to change.

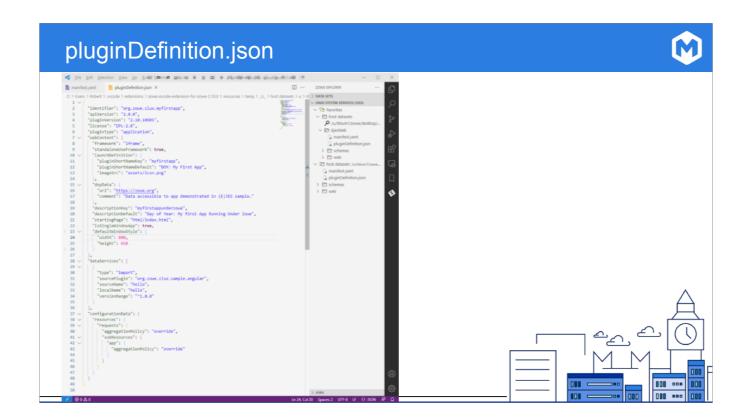
Dataservices is a bit arcane

AND

I'll discuss it later.



Here's what the changes I've discussed look like in the manifest.yaml file.



Here's what the changes I've discussed look like in the pluginDefinition.json file.

Managing the Web App



- Use Zowe app file layout (see earlier slide)
- Copy supporting files to /web/assets, /web/css, /web/html, and /web/js
- Modify
 - /web/html/index.html as discussed below to run your code
 - /web/css/iframe.css if necessary for your code
 - /web/assets/icon.png
- Specific changes
 - The DOY sample includes support files
 - If you remove the DOY code in index.html, you can remove everything but iframe.css in / wéb/css and main.js in /web/js
 - In the index.html file, find the </head> element
 - Replace the DOY code spanning from the </head> tag to the </html> tag with your web app
 - Change the <title> tag
 - Add <link> tags for your CSS
 - Add <script> tags to include your Javascript files







Now we'll talk about web apps.

You'll copy your web app files into these standard directories.

You'll modify the index.html file in all cases.

You'll want to change the iframe.css file or link to your own file.

You can save modifying the icon until you've gotten the app up and running.

The rest is a list of specific changes to the index.html file.

Managing the Web App continued



- · Additional specific changes
 - The icon supplied is a placeholder
 - Insert your own graphic, optionally putting your image in a round cutout as suggested by this spec: https://docs.zowe.org/stable/contribute/guidelines-ui/appicon/
 - I've left <!-- UNIT TEST --> functions from the original iframe sample app that exercise features like notification for reference
 - · Remove them if you don't want them
 - If the location of your starting HTML page is not the default html/index.html
 - Change the webContent.startingPage property in pluginDefinition
 - If the location of your app icon is not the default web/assets/icon.png
 - Change webContent.launchDefinition.imageSrc in pluginDefinition.json

Here's more additional changes you can read about later.

Changes Made to index.html



000 000 000

Add <script> and <link> elements

Add html between <body> and </html>

```
chtml>
dead)
cheal
charact charset="UTF-8")
cmeta nome="viewport" content="width-device-width, initial-scale=1.0">
clink rel="stylesheet" href="../css/iframe.css">
cscript
if(exports){
    var ZomeZliW_tempExports = exports;
}
var exports = {"_esModule": true};

//script>
cscript type="text/javascript" src="../../../lib/org.zowe.zlux.logger/e.9.0/logger_js">c/script>
cscript type="text/javascript" src="../../../lib/org.zowe.zlux.bootstrap/web/iframe-adapter.js">c/script>
cscript type="text/javascript" src="../../../org.zowe.zlux.bootstrap/web/iframe-adapter.js">c/script>
cscript type="text/javascript" src="../js/main.js">c/script>
cscript type="text/javascript" src="../js/main.js">c/script>
cscript type="text/javascript" src="../js/main.js">c/script>
clink rel="stylesheet" href="../css/japusy-ui-1.10.d.min.css" type="text/css">clink rel="stylesheet" href="../css/japusy-ui-1.10.d.min.css" type="text/css">clink rel="stylesheet" href="../css/japusy-ui-1.10.d.min.css" type="text/css">cscript charset="uif-8" src="../js/jauscy-li1.10.e.min.js">c/script>cscript charset="uif-8" src="../js/jauscy-li1.10.e.min.js">cscript charset="uif-8" src="../js/jauscy-li1.10.e.min.js">cscript>cscript charset="uif-8" src="../js/jauscy-li1.10.e.min.js">cscript>cscript charset="uif-8" src="../js/jauscy-li1.10.e.min.js">cscript>cscript>cscript charset="uif-8" src="../js/jauscy-li1.10.e.min.js">cscript>cscript>cscript charset="uif-8" src="../js/jauscy-li1.10.e.min.js">cscript charset="uif-8" src="../js/jauscy-li1
```

Here's all the changes I've made to the index.html file.

Install the Customized App



- Follow the tagging and installation instructions
- Deal with errors
 - Zowe desktop didn't come up?
 - · Check the Zowe STC job output for validation errors caused by bad metadata
 - Did you try to change the schema? Don't do that!
 - · Search the Zowe STC job output for your app's name
 - Are you receiving error messages about loading a file like manifest.yaml, e.g. ZWEL0066W?
 - Tag your files again
 - Your app isn't in the Zowe start menu?
 - Check the Zowe STC job output for minor errors
 - · Your issue is most likely the metadata files, but it could be misplaced or missing web app files
- Note
 - Once you've finished customizing your metadata, you no longer need to uninstall it to before you modify it; you can upgrade instead

Installing the custom app is pretty much the same as what we did before.

Note that I've included a list common errors in the pink area of this slide,

what to do about them.

I learned what to do the hard way, of course.

Debugging



- Use the JavaScript debugger built into your browser, exactly like for any other web app
 - Run your application
 - Display developer tools and then the debugger
 - Open the webpack folders (typically at the bottom) until you find your app
 - Set break points
 - At least in Firefox, breakpoints, watches, etc., stay set for subsequent runs of a Zowe iframe app



I was surprised to learn that debugging a Zowe iframe app was almost identical to debugging my original web app. The only major difference was finding my sources.

There's also the issue of inspecting into an iframe, but you'll figure a way around that pretty quickly. The big takeaway here is

you

don't need

to worry.

Updating a Modified App



- Change directory to the zwe directory discussed on a prior slide
 - cd runtimeDirectory/bin
- Upgrade the app
 - zwe components upgrade -c config -o plugin
 - · config is the fully-qualified path of the previously discussed zowe.yaml file
 - plugin is the path to the directory containing your manfest.yaml file
- Changed metadata?
 - Cycle Zowe to update Zowe's metadata references
 - zwe stop -c config
 - zwe start -c config
- Web app files changed but not metadata?
 - Logoff in the Zowe desktop browser tab
 - Refresh the tab
 - Log on



When you make a change to your Zowe app, here's what you need to do. At its simplest, you issue the **z-w-e components upgrade** command, logoff the Zowe desktop browser tab, refresh it,

logon to Zowe again,

If your tools change file tags, you'll have to tag them again before issuing the upgrade command.

Advanced: Insert a Web App from a Server URL



- Provide your server URL so it is available in index.html
 - Embed your server URL in the webContent property of pluginDefinition.json, or hardcode in index.html
 - This is fully implemented in the (E)JES Web GitHub repository
- Change tag id values as necessary to position your web app
- Add JavaScript that creates an iframe that fills the Zowe window:

```
if ( configuredDomainEjesWeb ) {
   let node = document.createElement('iframe');
   node.setAttribute('style', 'position: fixed;top: 0px;bottom: 0px;right: 0px;width: 100%;
   border: none;margin: 0;padding: 0;overflow: hidden;z-index: 999999;height: 100%;');
   node.setAttribute('id', 'ejesweb_iframe');
   node.setAttribute('src', configuredDomainEjesWeb + '/EjesWeb/');
   document.getElementById('ejesweb_reference_frame').appendChild(node);

   __ejesZlux = document.getElementById('ejesweb_iframe').contentWindow;
```

You can also insert a web app from a **server** URL.
I'll leave that for you to read.
It's what I did for that hugely complex company web app I showed at the beginning.

Advanced: Customize the Official Zowe iframe Sample



Procedure

- Remove the .github, build, and dco-signoff directories. You'll make your own repository.
 - Delete the .gitattributes .gitignore, CHANGELOG.md, CONTRIBUTING.md, sample-iframeapp.ppf, and sonar-project.properties files
- If you wish to run the official sample
 - Install the required angular sample app that is provided but not preinstalled with Zowe 2.x
 - zwe components install -c config -o runtimeDirectory/components/app-server/share/sample-angular-app
 - Follow the tagging and installation steps
 - Cycle Zowe
 - Run the angular sample from the start menu
- Modify the README.md and LICENSE files as necessary
- Follow previous steps to change the metadata files and to insert your own web code
- The angular sample above demonstrates iframe inter-application communications
 - If you don't plan to use this feature, remove:
 - The dataServices property from pluginDefinition.json
 - The data services code from index.html



Now that we gotten the Day of the Year app running, I've extended the procedure to the official Zowe iframe sample.

There's a gotcha in their sample, however

It requires the angular sample app be installed first, but it isn't explained.

Fortunately, that app is included with the Zowe distro.

This slide explains what you need to do to install it as late as 2.10 LTS.

I've added tips for customizing the sample beyond what I talked about earlier.

The official sample includes "unit tests."

These demonstrate things like notifications,

Zowe-style menus,

Zowe inter-application communication.

You may later become interested in these things,

OR NOT

I'm betting your web app already has it all, which is the point of this talk

Advanced: (E)JES Sample



- (E)JES Web iframe app framework Zowe plug-in
 - https://github.com/phoenixsoftware/ejes-web-zowe-desktop-plugin.git
 - **Note:** Running this plug-in requires (E)JES and its REST API, however, the code is still usable as a starting point for your application!
 - This plug-in uses postMessage to message between Zowe and iframe code (web/js/ webapp.js and index.html). The served up web app knows it runs under Zowe and behaves differently. It accesses persistence data and includes a function to request the Zowe plug-in to do work.
 - Functions in main.js are modified to return data via callback functions. instead of inserting data in HTML tags. Console log messages aid tracing.
 - Requires (E)JES V6R3, expected GA September 29, 2023.
 - It is in active development—I will be updating with further features in the near future!



At Phoenix Software, beyond our Zowe CLI offerings, we decided we wanted our (E)JES product on the Zowe desktop. You're welcome to examine the Zowe interface part of that.

To actually run it, you'll require (E)JES installed on z/OS.

Even not running (E)JES, the repo demonstrates communicating between the iframe and a server app, which you can modify and use.

The following two images are snapshots from the licensed part of the (E)JES web app, showing only the code that pertains to communicating with Zowe.

Match that with the code in the (E)JES Web Zowe desktop app repository to paint yourself a full picture.

webapp.js Part One



webapp.js Part Two



```
else {
    __log('clientZlux05E: Connect attempt with incorrect token rejected.');
    __app.zowe = undefined
}

return;
}

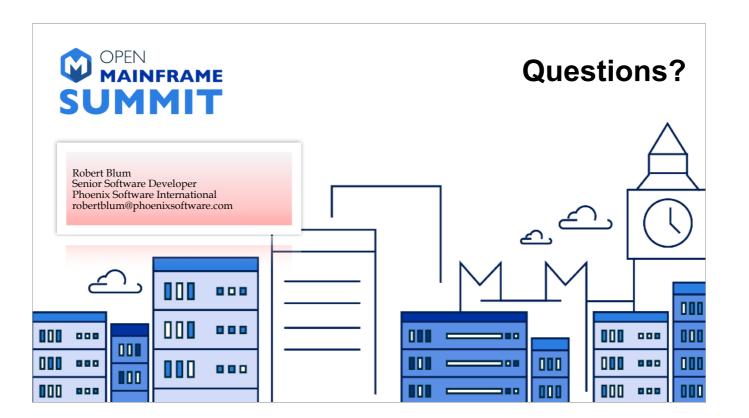
if (event.origin !== _app.zowe.configuredZlux0rigin) {
    __log('clientZlux07E: Message of unconfigured origin rejected: "' + event.origin + '"');
    return;
}

// "Assuming you'we verified the origin of the received message (which
// you must do in any case), a convenient idiom for replying to a
// message is to colt postMessage on event.source and provide
// event.origin as the torgetOrigin."

if ( event.dota.message != 'refresh' )
    __log('clientZlux08D: Processing Zowe verb ' + event.data.verb + ' with "' + event.data.message + '" message.');
});

/**

* @memberOf outerClosure
*/
function _zoweConfigurationRequestor() {
    _log('clientZlux08D: Zowe Configuration Recuestor');
    document.getElementById('webapp').setAttribute('style', 'display:none');
    _app.zowe.zoweMandle.postMessage('configureZowe', _app.zowe.configuredZluxOrigin);
}
```



And that's all folks.