

# How to Create a Tech Docs Builder to Automate Builds

PRESENTED BY

Jenny Pittman, Sr. Technical Writer, BeyondTrust



# PROJECTS ARE LIKE BORED RABBITS

- Tech Com projects were multiplying

- Master project
- Four children
- Up to 39 languages

= 44 projects

- But wait, there's more!

- Acquired Lieberman Software: one new project
- Acquired Avecto: two new projects
- Acquired BeyondTrust: forty new projects

**NOW, THAT'S A LOT OF PROJECTS!**



# WHEN I WAS GOING TO ST. IVES...

- Currently working with 22 projects
- Each project has 2 to 165 targets

Mean	23 targets per project
Median	11 targets per project
Sum	514 targets



## THE SOLUTION?

**It's madbuild.exe  
to the rescue!**

# CODE IN A CODE IN A CODE

- Complex code to simplify a complex process for non-technical users
  - HTA skin
  - VBScript
  - Command line
  - Madbuild.exe
  - Text file
- User experience: easy-peasy





# STARTER CODE TO BUILD YOUR TARGETS

- Lightweight script
- Works in most environments
- Modify to your own environment

Demo time!



So, how does it work?



# Let's talk about **madbuild.exe**

- Madbuild.exe is a command line executable that allows you to run builds outside of the Flare application.
- Madbuild.exe is run from a command line or a batch file.

# FIRST, FIND FLARE

- The batch file should show:

```
CD /D "C:\Program Files\MadCap Software\MadCap Flare 14\Flare.app"
```

- The command line must run from where the Flare app lives, and **CD /D** moves both its drive and its directory.
- To get HTA to write this to the batch file:

```
outFile.WriteLine "CD /D ""C:\Program Files\MadCap Software\MadCap  
Flare 14\Flare.app"""
```

- The **first quote** tells the script to start writing.
- The **second two quotes** tell the script to write a quote.
- The **third three quotes** tell the script to write a quote, then stop writing.

## NEXT, DESIGNATE TARGETS

- The batch file should show:

```
madbuild -project "C:\MadWorld\EN-Knit\EN-Knit.flprj" -target "knitting-  
for-nerds.fltar"
```

- To get HTA to write this to the batch file:

```
outFile.WriteLine "madbuild -project "" & tarObj.tarFullPath & "" -target  
"" & targetFold & tarObj.tarName & """"
```

- The **first quote** tells the script to start writing to the output file.
- The **next three quotes** tell the script to write a quote, then stop (or start) writing literally.
- The **ampersands** join a literal with a variable (or two vars).
- The **last four quotes** start a literal, write a quote, then stop writing.



Key knowledge:  
object-oriented programming

# LEGO EXAMPLE: LEGO OBJECT CLASS

- A LEGO person is an object with its own class, **objLego**.
- The **objLego** class has four variables: **.name**, **.legs**, **.arms**, and **.head**.



# I NEED A HERO ARRAY!

- We create a **heroArr** list to store our LEGO objects.
- Each time we add a hero, we create a new **objLego**.

Set heroArr(0) = New objLego    Set heroArr(1) = New objLego    Set heroArr(2) = New objLego    Set heroArr(3) = New objLego

Batman



Wonder Woman



Superman



The Flash



## ACCESS A SPECIFIC HERO

- If we call `heroArr(1).name`, we get **"Wonder Woman"**.
- `heroArr(1).head` returns
- `heroArr(1).arms` returns
- `heroArr(1).legs` returns



# REAL EXAMPLE: AVAILABLE TARGET CLASS

Given the path `C:\MadWorld\EN-Knit\Project\Targets\patterns\yoda.fltar`

Class availTarget	
Public tarPath	<code>C:\MadWorld\EN-Knit\Project\Targets\patterns</code>
Public tarFold	<code>patterns</code>
Public tarName	<code>yoda.fltar</code>
Public tarProjPath	<code>C:\MadWorld\EN-Knit</code>
Public tarProj	<code>EN-Knit</code>
Public tarFullPath	<code>C:\MadWorld\EN-Knit\EN-Knit.flprj</code>
End Class	



# BUILD YOUR KNOWLEDGE

- For a deeper but still comprehensible explanation, read **How to explain object-oriented programming concepts to a 6-year-old**
- <https://medium.freecodecamp.org/object-oriented-programming-concepts-21bb035f7260>



Other important stuff to know

# GET TO KNOW THE WINDOWS SCRIPT HOST

- Necessary to interact with files and folders

Create objects	<pre>Set objFSO = CreateObject("Scripting.FileSystemObject") Set WshShell = CreateObject("WScript.Shell")</pre>
Get the directory	<pre>Set objThisFile = objFSO.GetFile(thisHTALocation) Set objFolder = objFSO.GetFolder(objThisFile.ParentFolder) Dim objCurrDir = objFolder.ShortPath WshShell.CurrentDirectory = objCurrDir</pre>
Write the output	<pre>Dim fileDate = objFSO.GetFile("BuildOutput.txt").DateLastModified Set objFile = objFSO.OpenTextFile("BuildOutput.txt", ForReading)</pre>
Run the command	<pre>Set outFile = objFSO.CreateTextFile("build.cmd", True) WshShell.Run "build.cmd"</pre>



# HAVE A PLAN TO CHECK RESULTS

- **Summary log:** Key output messages to show build start, errors, and completion
- **Full log:** All messages shown for full details
- **Text file:** All messages saved to a text file for later review
  - You could modify this code to save the text file with a timestamp in the name so your records don't get overwritten

# BE SELECTIVE WITH SELECT BOXES

- Great way to link two bits of info
- Assign really useful info as the value
- Keep the text short and sweet

```
Set objOpt = document.CreateElement("OPTION")
objOpt.Value = projPath
objOpt.Text = projName
document.getElementById("sel1").appendChild(objOpt)
```



Where do we go from here?



## LESSONS LEARNED

- Keep that code clean
- Comment, comment, comment!
- Account for different structures and settings
- Version your code
- Scalable solutions are great...
- But sometimes, you have to just make it work *right now*



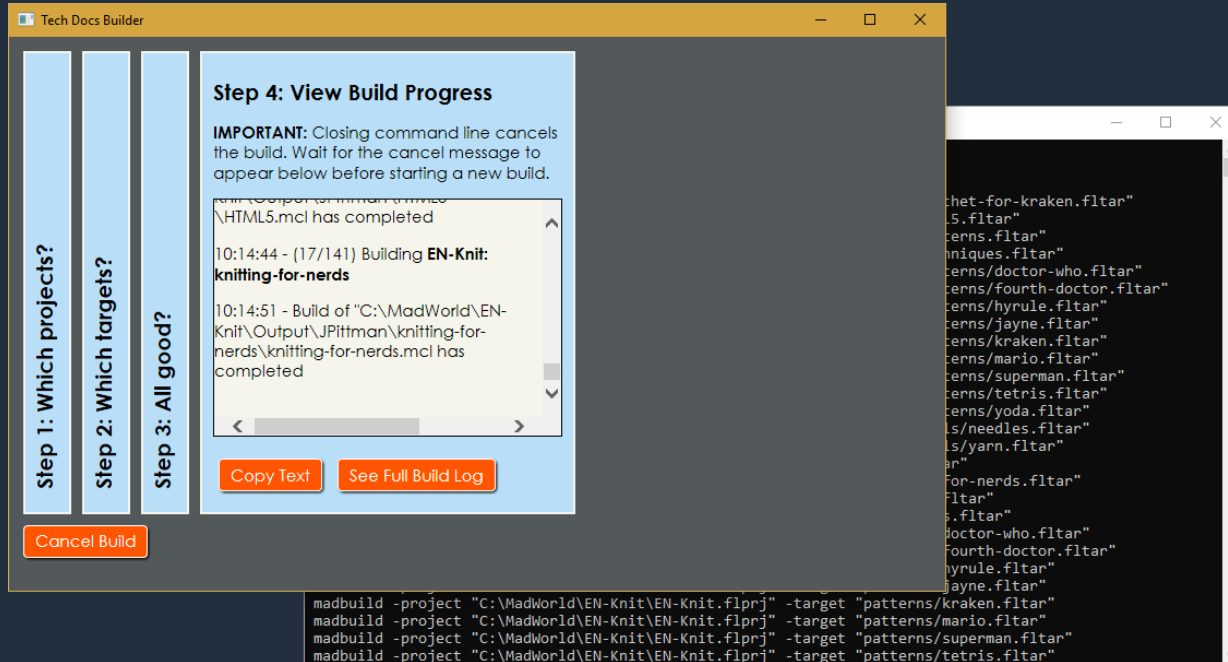
## DREAM ON

- Make the builder work with source control: pull, push, branch, merge, etc.
- Schedule builds to run later and/or on a recurring basis
- Import resources through Flare and/or directly through the file system
- Search projects and targets by name
- Filter targets by condition tag
- Filter targets by type: PDF, HTML5, Word, etc.



# DOWNLOAD THE BUILDER!

- <https://github.com/jennyapittman/tech-docs-builder>



# RESOURCES

- <https://help.madcapsoftware.com/flare2018r2/Content/Flare/Output/Building-Targets-Using-Command-Line.htm>
- <https://help.madcapsoftware.com/flare2018r2/Content/Flare/Exporting/Exporting-Projects-Using-Command-Line.htm>
- <http://tregner.com/flare-blog/tag/madbuild-exe/>
- <https://devblogs.microsoft.com/scripting/?s=hta>
- <https://www.599cd.com/tips/hta/>