

- ▶ **TRIRIGA Wiki Home**
- ▶ **Facilities Management ...**
- Facilities Maintenance**
- ▶ **Environmental & Ener...**
- ▶ **Real Estate Management**
- ▶ **Capital Project Manag...**
- ▶ **CAD Integrator-Publis...**
- ▶ **IBM TRIRIGA Connect...**
- ▶ **IBM TRIRIGA Anywhere**
- ▶ **IBM TRIRIGA Applicati...**
- ▶ **Release Notes**
- ▶ **Media Library**
- ▶ **Best Practices**
- ▶ **Upgrading**
- ▶ **Troubleshooting**
- ▼ **UX Framework**
 - UX Articles**
 - ▼ **UX App Building**
 - Introducing UX**
 - Implementing UX**
 - Extending UX**
 - Implementing UX (...)**
 - Extending UX (Poly...**
 - Converting UX to P...**
 - ▶ **Bundling UX (Poly...**
 - Commanding UX (...)**
 - ▶ **UX Perceptive Apps**
 - ▶ **UX in Foundation To...**
 - ▶ **UX App Designer Tools**
 - UX Best Practices**
 - ▶ **UX in Foundation Docs**
 - UX Component Docs**
 - ▶ **UX Tips & Tricks**
 - UX Videos**
 - ▶ **UX Archives**

[New Page](#)

[Index](#)

[Members](#)

[Trash](#)

▼ **Tags**

?

Find a Tag

[analysis](#)
[application](#)
[availability_section](#)
[best_practices](#)
[cad](#)
[change_management](#)
[changes](#)
[compare](#)
[compare_revisions](#)
[customizations](#)
[customize](#)
[database](#)
[db2](#)
[exchange](#)
[find_available_times](#)
[gant_chart](#)
[gant_scheduler](#)
[group](#)
[memory_footprint](#)
[modifications](#)
[modify](#)
[object_label](#)
[object_revision](#)
[operating_system](#)
[oracle](#)
[performance](#)
[platform](#)
[problem_determination](#)
[reports](#)
[reserve](#)
[reserve_performance](#)
[revision](#)
[revisioning](#)
[single_sign-on](#)
[snapshot](#)
[space](#)
[sql_server](#)
[sso](#)
[support](#)
[system](#)
[system_performance](#)
[tags:](#)
[track_customizations](#)
[tririga](#)
[troubleshoot](#)
[tuning](#)
[upgrade](#)
[ux](#)
[version](#)
[versioning](#)

Cloud | [List](#)

You are in: [IBM TRIRIGA](#) > [UX Framework](#) > [UX App Building](#) > Bundling UX (Polymer 3)

Bundling UX (Polymer 3)

 Like | Updated June 13, 2019 by [Jay.Manaloto](#) | Tags: *None* [Add tags](#)

Edit

Page Actions

UX Framework

UX
App Building

UX
in Classic Tools

UX App
Designer Tools

UX
Best Practices

See the Polymer website at www.polymer-project.org for more about Polymer 3. See the NPM website at www.npmjs.com for more about Node.js.

Bundling UX (Polymer 3): Packaging your UX view into a single component file

STILL HERE? If you're ready for yet another serving, I admire your appetite! In my [previous articles](#), we explored the concepts, built a simple UX application, and extended it with new fields, buttons, dialogs, toasts, and ways to manipulate records. This time, we'll discuss why you should bundle your custom UX apps, and how to bundle your UX view into a single component file.

- Why should we bundle our UX apps?
- How do we get the NPM and TRIRIGA tools?
- How do we use tri-bundler?
- Still confused or curious?

Why should we bundle our UX apps?

A typical IBM TRIRIGA UX app might initially load **hundreds** of resource files which translate to the same number of HTTP requests from the server to the browser. This remains true in successive requests of these resources even if they are already cached by the browser. The browser still makes the same number of HTTP requests to the server, then waits for the server response to either load it from the cache or from the latest version from the server. This **high** number of HTTP requests can be **costly** and can **negatively** impact the performance of your UX app.

One way to cut down this cost is to concatenate or "bundle" these web resources into a single production file. For **Polymer 1**, we used the [tri-vulcanize](#) tool. But for [Polymer 3](#), we're using the [tri-bundler](#) tool.

Notes:

- The [tri-bundler](#) tool is designed specifically for **Polymer 3**.
- To be clear, our UX framework supports **both** Polymer 1 and Polymer 3 UX apps. Both types of UX apps can **coexist** together in the same environment. But if you're interested in [converting](#) your own **custom** UX apps from Polymer 1 to Polymer 3, see the [tri-polymer-upgrade](#) tool. Be aware that this tool is **not** officially supported at this time.

How do we get the NPM and TRIRIGA tools?

Contact your IBM TRIRIGA representative or business partner if you cannot access the download location of the [Node.js Package Manager \(NPM\)](#) tool. This NPM tool is used to install [several TRIRIGA tools](#) which allow you to populate the JavaScript (JS) files in your view metadata, preview your JS changes, and sync (deploy) your JS changes with the JS files in your TRIRIGA environment. Be aware that these tools are **not** officially supported at this time.

Download and install the Node/NPM file. For example: **node-v8.12.0-x64.msi**.

Next, open your command prompt. Run the following NPM command to install the following TRIRIGA tools.

If you see any NPM-related warnings (optional, unsupported, or deprecated), you can ignore them.

TRIRIGA Tools.

Tool	Description
tri-pull	<p>npm install @tririga/tri-pull -g</p> <p>This command installs the tri-pull tool.</p> <p>tri-pull</p> <p>This is a simple tool that pulls UX views from a TRIRIGA server. It updates the UX view files in the executed local directory with the files from the server. This tool resembles the WebViewSync pull command. If you're curious, feel free to check out the tri-pull options and details.</p>
tri-bundler	<p>npm install @tririga/tri-bundler -g</p> <p>This command installs the tri-bundler tool.</p> <p>tri-bundler</p> <p>This is a simple tool that bundles a TRIRIGA UX view in Polymer 3 into a single output component file. This tool resembles the tri-vulcanize tool in Polymer 1. If you're curious, feel free to check out the tri-bundler options and details.</p> <p>The tri-bundler tool is designed specifically for Polymer 3.</p>

NPM > Install TRIRIGA Tool.

```
C:\>npm install @tririga/tri-bundler -g
npm WARN deprecated sw-precache@5.2.1: Please migrate to Workbox: https://devel
opers.google.com/web/tools/workbox/guides/migrations/migrate-from-sw
npm WARN deprecated sw-toolbox@3.6.0: Please migrate to Workbox: https://develo
pers.google.com/web/tools/workbox/guides/migrations/migrate-from-sw
C:\Users\J8888888\AppData\Roaming\npm\tri-bundler -> C:\Users\J8888888\AppData\
Roaming\npm\node_modules\@tririga\tri-bundler\bin\tri-bundler
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.9 (node_modules\@t
ririga\tri-bundler\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents
@1.2.9: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64
"})

+ @tririga/tri-bundler@1.0.8
added 870 packages from 620 contributors in 58.588s

C:\>
```

NPM > tri-bundler.

```
C:\>tri-bundler
tri-bundler
  A tool for bundling a Polymer 3 TRIRIGA UX view into a single output
  component file.

Synopsis
  $ tri-bundler

Options
  -u, --user user           The user name of your TRIRIGA user. Required.
  -p, --password password   The password for your TRIRIGA user. Required.
  --url url                 The URL of the TRIRIGA server. It must include
                             the context path. Required.
  --basicuser user          The user name for basic authentication.
  --basicpassword password The user password for basic authentication.
  -v, --view view name ... The name of the view to be bundled. Required.
  --component file ...     The root component filename of the view you want
                             to bundle. Required.
  --output file ...        The output component filename that will be
                             generated after bundling. Required.
  --version                   Print tri-bundler version.
  -q, --quiet                 Do not print any non-error message to the
                             console.
  -h, --help                  Print this help information.
```

How do we use tri-bundler?

Step 1: Pull your view from the server.

First, open your command prompt.

To pull your view files from the server, run the following **tri-pull** command: **tri-pull -u username -p password --url http://[hostname:port]/[context_path] -v view-exposed-name** where **-u** applies your TRIRIGA username, **-p** applies your TRIRIGA password, **--url** defines the server URL, **-v** defines your view, and **view-exposed-name** is the exposed name of your view (with the dash).

Step 2: Bundle your view into a single JS file.

Next, in your command prompt, change directory to the folder of the pulled view files that you want to bundle.

To bundle your view files from this folder, run the following **tri-bundler** command: **tri-bundler -u username -p password --url http://[hostname:port]/[context_path] -v view-exposed-name --component view-exposed-name-dev.js --output view-exposed-name.js** where **-u** applies your TRIRIGA username, **-p** applies your TRIRIGA password, **--url** defines the server URL, **-v** defines your view, **view-exposed-name** is the exposed name of your view (with the dash), **--component** defines the root component filename of your view, and **--output** defines the output component filename that is generated for your view.

Step 3: Deploy your view to the server.

Next, return to the command prompt in the same folder of the pulled and bundled view files.

To deploy (or push) your view files to the server, run the following **tri-deploy** command: **tri-deploy -t http://[hostname:port]/[context_path] -u username -p password -v view-exposed-name** where **-t** targets the server URL, **-u** applies your TRIRIGA username, **-p** applies your TRIRIGA password, **-v** defines your view, and **view-exposed-name** is the exposed name of your view (with the dash).

Notes:

- If your environment implements a firewall, the **tri-deploy** command must include 2 additional options and their values, where **--basicuser** applies the username for basic authentication, and **--basicpassword** applies the password for basic authentication. Feel free to check out the [tri-deploy](#) options and details.

Still confused or curious?

If you have any questions about UX that weren't answered in this article, feel free to reach out to your IBM TRIRIGA representative or business partner. Or if you want, I'll go ask the team.

[Next >](#)

Comments (0) | Versions (8) | Attachments (2) | About

There are no comments.

[Add a comment](#)

 [Feed for this page](#) | [Feed for these comments](#)

