

Wikis

IBM TRIRIGA

- ▶ [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](#) > [Commanding UX \(Polymer 3\)](#)

Commanding UX (Polymer 3)

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

Edit

Page Actions

UX Framework

UX
App BuildingUX
in Classic ToolsUX App
Designer ToolsUX
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.

Commanding UX (Polymer 3): Managing your UX view with NPM-based tools

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, for convenience, we'll repeat how to get the NPM and TRIRIGA tools, and we'll list all of the NPM-based tools in one place.

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-template	<p>npm install @tririga/tri-template -g</p> <p>This command installs the tri-template tool.</p> <p>tri-template</p> <p>This is a simple tool that generates UX view skeletons from available templates. This tool resembles the WebViewSync addview -s starter view command. If you're curious, feel free to check out the tri-template options and details.</p>
tri-proxy	<p>npm install @tririga/tri-proxy -g</p> <p>This command installs the tri-proxy tool.</p> <p>tri-proxy</p> <p>This is a simple tool that serves UX views from your local file system and proxies all other view files and calls to a TRIRIGA server. This tool resembles the WebViewSync sync -a command, but provides a continuous preview (after each file save) without permanent changes. If you're curious, feel free to check out the tri-proxy options and details.</p>
tri-deploy	<p>npm install @tririga/tri-deploy -g</p> <p>This command installs the tri-deploy tool.</p> <p>tri-deploy</p> <p>This is a simple tool that deploys UX views to a TRIRIGA server. It updates the UX view files on the server with the files from the specified local directory, and deletes any files on the server that does not exist in the local directory. This tool resembles the WebViewSync push or sync -a command, but provides a one-time action (not continuous) with permanent changes. If you're curious, feel free to check out the tri-deploy options and details.</p>
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-polymer-upgrade	<p>npm install @tririga/tri-polymer-upgrade -g</p> <p>This command installs the tri-polymer-upgrade tool.</p> <p>tri-polymer-upgrade</p> <p>This is a simple tool that automatically converts TRIRIGA UX views, UX components, and UX apps from Polymer 1 to Polymer 3. If you're curious, feel free to check out the tri-polymer-upgrade options and details.</p>
tri-vulcanize	<p>npm install @tririga/tri-vulcanize -g</p> <p>This command installs the tri-vulcanize tool.</p> <p>tri-vulcanize</p> <p>This is a simple tool that bundles a TRIRIGA UX view in Polymer 1 into a single output component file. This tool resembles the tri-bundler tool in Polymer 3. If you're curious, feel free to check out the tri-vulcanize options and details.</p>

	The tri-vulcanize tool is designed specifically for Polymer 1 .
tri-bundler	npm install @tririga/tri-bundler -g This command installs the tri-bundler tool. tri-bundler 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. The tri-bundler tool is designed specifically for Polymer 3 .

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 (4)
- Attachments (0)
- About

There are no comments.

[Add a comment](#)

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