

# DHIG Eng Onboarding Guide

- [Overview](#)
- [Digital HIG Background](#)
- [What exactly is Digital HIG Engineering?](#)
- [Working with Key Stakeholders](#)
- [Development and Team Procedures](#)
- [Onboarding Checklist](#)
- [Key Resources to Review First](#)
- [Additional Important Resources](#)
- [Keep in Mind](#)

## Overview

This onboarding guide is intended for engineers new to **building** Digital HIG components.

- For engineers **consuming** DHIG components and CSS, please see [DHIG Eng Guides](#).

## Digital HIG Background

Digital HIG (DHIG) is a comprehensive system of reusable code components, together with standards, guidelines, documentation, and toolkits. The goal is to enable people to rapidly create cohesive, precise, and accessible web experiences for the Autodesk Digital Platform. [Digital HIG](#) is a collaboration between DPE XD and DPE Engineering, as well as other stakeholders.

'HIG' refers to Human Interface Guidelines. Digital HIG is the design system for Autodesk's digital properties (e.g., company website, product centers, account, profile, AKN). It is distinct from the Autodesk's [Product HIG](#).

A number of atomic components (such as color, typography, spacing), foundational patterns, and more complex components are completed and in use, while numerous others are in progress or on the roadmap.

### HIG vs DHIG

In the most basic terms, the HIG is for product UI and the Digital HIG is for Autodesk branded online experiences. They share some common elements, as the Digital HIG was originally created as a "branch" of the HIG, with deviations as necessary to meet unique needs. However, because it applies to external-facing properties like the Autodesk website, it's critical that Digital HIG fully reflect the new brand. The result is a greater difference in look & feel between HIG and Digital HIG.

- DHIG originally used HIG as its foundation, but currently, DHIG uses Material UI as its foundation.

### Rosetta vs non-Rosetta

'Rosetta' refers to the new Autodesk branding introduced in September 2021.

- Note that some consuming teams are using Rosetta DHIG components and utility CSS, while some are using pre-Rosetta DHIG.

## What exactly is Digital HIG Engineering?

The **Digital HIG Engineering** project is a materialization in code, documentation, and other consumable assets of the larger Digital HIG design system.

Digital HIG Engineering owns the source and produces all style assets (CSS, SASS, React components, etc.) used to build patterns for web experiences, with the assumption that these patterns are designed directly from all elements and patterns defined in the Digital HIG Figma UI Toolkit.

DHIG owns the legos to build the Pattern Library castle but does not own the castle.

It's also important to keep in mind that Digital HIG engineering does NOT produce a repository of site-specific patterns.

**The implementation is comprised of the following components:**

Name	Link	Description
git repository	<a href="#">Digital HIG</a>	Source code repository holding the implementation
Storybook	<a href="https://storybook.digital-help-stg.autodesk.com/digital-hig/current/index.html?path=/docs/overview-page">https://storybook.digital-help-stg.autodesk.com/digital-hig/current/index.html?path=/docs/overview-page</a>	Main documentation site, generated automatically from the contents of the git repo
npm package	<a href="#">Artifactory</a>	Collection of libraries containing the implementation that can be consumed as a dependency in project builds
Assets in CDN	<a href="#">Artifakt fontset CSS (QSG with CDN link)</a> , <a href="#">Utility CSS (QSG with CDN links)</a> , <a href="#">Icon sprites</a>	Assets deployed to publicly accessible locations that can be consumed during runtime

## Working with Key Stakeholders

### Designers

We work closely with VisD (Visual Design) and XD (Experience Design).

Figma UI Toolkit is the primary tool they use to document the designs on which our code is based.

- Current (Rosetta 2021)
  - <https://www.figma.com/file/NpV47EII6zCxpPoChejg0/%E2%9D%96-Digital-HIG-UI-toolkit?node-id=0%3A1> (cover page)
  - <https://www.figma.com/file/NpV47EII6zCxpPoChejg0/%E2%9D%96-Digital-HIG-UI-toolkit?node-id=3198%3A190092> (getting started)
- Pre-Rosetta (DHIG 2020) - Deprecated (but may still be relevant to you as an engineer in some use cases)
  - <https://www.figma.com/file/AWw6MUsaHMgGnHuXCXUcej/DHIG-2020---deprecated?node-id=119%3A3693>
  - [Figma Toolkit & Design Workflow](#)

Zeplin is used by consuming teams for handoff from design to dev.

Digital HIG Design and Engineering collaborate via this slack channel: #priv-dhig-design-engineering (This is a channel where DHIG Design and Engineering teammates can discuss issues that arise, and align on strategy and execution.)

### Consumers

We work closely with the teams that consume what we build. These include:

Dotcom  
Checkout  
Web Publishing  
XP (Me Menu)  
Universal Header  
Universal Search/IPH  
AKN  
UHM  
Contact App  
AVA  
Community (Forums)  
Learning & Credentialing  
Apollo: O2P, OC  
Profile  
Account  
ODIS Installer  
ADA (Autodesk Desktop App)

## Development and Team Procedures

- [DHIG Eng Scrum Agreements](#) contain important information about the team and our scrum processes.
- [Code Review Guidelines](#)

### QA/UAT

- [Digital HIG Testing Strategy](#)
- [Testing and Validation \(UAT\) for Digital HIG Components](#)







### Support

- [DHIG Eng Support & Escalation](#)
- [#dpe-dhig-engineering-support](#)

## Onboarding Checklist

At present, DHIG engineers must use a Mac. We've been unable to make the dev environment workable on a Windows machine. We hope to revisit this requirement and offer more flexibility in the future.

Follow this checklist to get access to the resources you'll need as a Digital HIG engineer:	
<a href="#">blocked URL</a> Enable 2FA	<ul style="list-style-type: none"><li>• Download DUO and enable it to access internal applications(wiki, jira, jenkins...)<ul style="list-style-type: none"><li>◦ For Android: <a href="https://dl.duosecurity.com/DuoMobile-latest.apk">https://dl.duosecurity.com/DuoMobile-latest.apk</a></li></ul></li><li>• <a href="https://landing.autodesk.com/">https://landing.autodesk.com/</a></li></ul>

 Github access	<ul style="list-style-type: none"> <li>• Sign the SAM agreement, GitHub only and ignoring TFS <ul style="list-style-type: none"> <li>◦ <a href="https://share.autodesk.com/sites/sam/Home.aspx">https://share.autodesk.com/sites/sam/Home.aspx</a></li> </ul> </li> <li>• Login: <a href="https://git.autodesk.com/">https://git.autodesk.com/</a></li> </ul>
 Pull the code	<ul style="list-style-type: none"> <li>• DHIG Engineering related repos <ul style="list-style-type: none"> <li>◦ <a href="#">Apothecary</a></li> <li>◦ <a href="#">Digital HIG</a></li> <li>◦ <a href="#">Digital HIG Reference app</a></li> </ul> </li> </ul>
 npm registry	<ul style="list-style-type: none"> <li>• Follow steps in <a href="#">README.md#setup</a> to add adsk and dhig registry locations to your local .npmrc</li> </ul>
 Email	<ul style="list-style-type: none"> <li>• Add to team email alias: <a href="mailto:dhig.engineering@autodesk.com">dhig.engineering@autodesk.com</a></li> </ul>
 Slack	<ul style="list-style-type: none"> <li>• <a href="https://autodesk.slack.com/">https://autodesk.slack.com/</a></li> <li>• Download and install the app</li> <li>• Join channels <ul style="list-style-type: none"> <li>◦ <a href="#">#priv-rosetta-dhig-engineering</a></li> <li>◦ <a href="#">#dpe-dhig-engineering-support</a></li> <li>◦ <a href="#">#priv-rosetta-dhig-notifs</a></li> <li>◦ <a href="#">#dpe-digital-hig</a></li> <li>◦ <a href="#">#digital-hig-slack-standup</a></li> </ul> </li> <li>• Add to @dhig_engineers Slack alias using the Edit User Group Info workflow in <a href="#">#adsk-slack-help</a></li> </ul>
 JIRA access	<ul style="list-style-type: none"> <li>• Add to developer role</li> <li>• Team board: <a href="https://jira.autodesk.com/secure/RapidBoard.jspa?rapidView=11738">https://jira.autodesk.com/secure/RapidBoard.jspa?rapidView=11738</a></li> </ul>
<a href="#">blocked URL</a> Jenkins access	<ul style="list-style-type: none"> <li>• <a href="#">Digital HIG on Master-5 in Jenkins</a></li> </ul>

## Key Resources to Review First

Note that some of these resources require VPN.

- [Onboarding Checklist](#) above
- Rachel's Digital HIG Engineering [lightning talk](#) (may require authenticating to MS Stream first).
  - Her brief presentation touches upon the following topics:

*DHIG website; Figma UI Toolkit; Primitives and tokens; Theme data; Storybook documentation; Quick Start Guides; Pharmacopeia repo (static assets); Components*
- Digital HIG website: <https://digital-hig.autodesk.com/>
- Storybook: <https://storybook.digital-help-stg.autodesk.com/digital-hig/current/index.html?path=/docs/overview-page>
  - See theme data, components, community sections
  - Examining the source code in the Github repo alongside reading the Storybook docs is a very useful approach
- Codebase:
  - Digital HIG repo: <https://git.autodesk.com/dpe/digital-hig>
    - README provides useful information
- [Code Contribution Workflow](#)
- [Theme data build process](#)
- [Material UI](#) - Familiarize yourself with how it works, how to manage the theme
- Digital HIG Demo/Reference App
  - app -<https://pages.git.autodesk.com/dpe/digital-hig/#/ref-app/>
  - code -<https://git.autodesk.com/dpe/digital-hig/tree/master/packages/ref-app>
- Digital HIG component development process for JIRA
- [Digital HIG quick start guides](#) to get a sense of how teams consume what we build

## Additional Important Resources

- [Digital HIG \(DHIG\) Engineering Roadmap](#)
- (Draft) [Digital HIG Engineering Usage & Contribution Model](#)
- [Digital HIG Eng troubleshooting](#)
- Consumption Apps
  - [DHIG Consumption App](#) - shows monthly trend on number of repos using DHIG packages

- The procedure for updating the Digital HIG app is documented here: <https://git.autodesk.com/dpe/dhig-rosetta-adoption#data-gathering>
- [Apothecary Consumption App](#) - shows monthly trend on number of repos using Apothecary packages
  - The procedure for updating the Apothecary app is documented here: <https://git.autodesk.com/dpe/digital-hig-consumption#data-gathering>
- [Dev Audit Template](#)
- [Accessibility](#)
- [EXPM-826: Automate publishing of Digital HIG assets](#)
- [Digital HIG Engineering](#) (this wiki will be updated/re-organized)
- [Customer Facing Platform Home](#)
- Community (Apothecary) codebase: <https://git.autodesk.com/dpe/apothecary/>
  - [Readme.md](#): <https://git.autodesk.com/dpe/apothecary/blob/master/README.md>
  - Review setup & node version
    - [Contributing.md](https://git.autodesk.com/dpe/apothecary/blob/master/CONTRIBUTING.md): <https://git.autodesk.com/dpe/apothecary/blob/master/CONTRIBUTING.md>
      - Review setup, testing standards, review process, component standards

## Keep in Mind

- When you have questions about implementing a design or notice unexpected design behavior, don't hesitate to consult directly with the designer
- Subscribe to follow the DHIG PRs and Github issues
  - Seeing debugging and support processes from the Digital HIG side in action allows you to see how the fixes were tackled, which will improve your understanding of the codebase.
- We encourage new engineers to try and document processes as much as they can; not only does it help to have documentation, but it's a great way of getting familiar with how things work.
- The following are some points that were sources of confusion to Rachel when she was onboarding:
  - Dev Process
    - The Storybook **docs usually only feature a yarn command under the "installation" section**. All my projects used npm, so this initially threw me off (I also encountered a few errors until I updated my node version).
    - What the correct "source of truth" was; **comps from designers on other teams did not always match the Digital HIG specifications** or map to a specific variant. I've recently dealt with a few support questions asking about the same, so I would advise onboarding engineers to be clear about Digital HIG design patterns (e.g. awareness of our existing tokens and primitives).
    - I didn't know how many approvals were generally needed before merging was allowed. The answer is technically only one, although I find it is usually worth waiting for more eyes, as rushing to merge after one approval can lead to certain issues being missed. If you're otherwise occupied and don't quickly add your own PR comments, **it can be easy to miss the reviewing window**, so it's best to make reviews a priority – just something to be aware of.
    - **Missing access**: there were a few instances of wikis/channels/specs that I was restricted from without realizing. Having access from the start could have saved me some time as well as prevented me from taking other engineers' time by asking questions that already had well-documented answers. Additionally, it was **not always clear who to ask for access**; in one instance I asked several engineers for admin access to the Jenkins pipeline before landing on the correct person. Similarly with the Pharmacopeia repo – I ultimately needed to ask Kevin Compton, who is outside of the direct Digital HIG team.
    - **It is very tricky to test that the fixes we implement will work before merging** since this varies a lot depending on the consumer's project set-up. **It is also challenging to test that the work we implement won't break anything**. I was given a test repo to use for that purpose, but it requires a lot of manual copy & pasting of build styles (initially wasn't even sure what that entailed), and it is easy to miss something.
    - The **distinction between our theme-data and theme-mui packages** wasn't instantly clear to me.
    - One piece of feedback I was given on the [Integrate DHIG Rosetta theme data into your SASS](#) guide was that the Webpack example is not specific to every case, although I am not sure how this can be avoided, as everyone will have a slightly unique set-up.
  - Repos
    - Initially, I wasn't clear about **which work was part of the Apothecary repo and which was part of the Digital HIG repo**. The confusion arose from the team being always referred to as "Digital HIG", but at the time I joined (start of June), the Apothecary repo was the main one in use. It wasn't clear at first glance that the "Digital HIG" repo was for the new Rosetta styles. For that matter, I was not entirely clear about what "Rosetta" was – although this particular detail may no longer be relevant for engineers onboarding now. I received a support question from a consumer just a couple of days ago asking about which of the two libraries they should be using.
    - When setting up the Apothecary repo, there is a "fontawesome" error that is [documented here](#), but still seems easy to miss.
- The following are some more things to keep in mind, provided by Arjun:
  - Dev Process
    - Make sure you do not make a lot of changes to Storybook at once, as the error log is not very helpful. Simple issues like missing closing tags completely nuke the Storybook page and remove the page from the list.
    - Rebasing the code in order to pass visual regression is a pretty important thing that we end up doing, make sure you are familiar with the process.
    - Always run a local VR test before pushing to the repo, and make sure when you build the PR build, you go to Jenkins and re-run the build with the parameters to disable the VR test. This way you will have a pre-release build that you can demo to design etc and add to your open PR build comment.
    - It's **never** a good idea to disable VR and merge your branch.
    - We use `git-cz` to do our commit messages, please make sure you go through the process of committing and pushing (see [Code Contribution Workflow](#)).
    - Make sure if you have not been able to complete 100% of your ticket, please communicate this with the product owner/manager and make an attempt to publish some work before the end of the sprint. It's always better to show work in progress than to show nothing.
    - When doing CSS work please make sure to use the primitives and tokens defined by the design system. It seems very cumbersome at first but makes life much easier in the long run.
    - No unnecessary comments should be left in the code - only if absolutely necessary and unless needed for storybook documentation.

- Write Less Code, but write Cleaner code. Write once, refactor many times. Write code that is readable and maintainable such as comments are un-needed. We are the providers of Digital HIG and we are committed to making it easier to consume for our users.
- Troubleshooting
  - Consider using an external terminal like `iTerm` to run the Storybook and build pipelines in parallel so you can cross-reference the outputs. Also helps to run `htop` to see system performance.
  - If something is not showing up correctly in your code, be it CSS or Storybook output, try doing a `yarn build` or just `yarn` to see if it fixes it. Also, it's a good idea to restart the Storybook server too as a preliminary step.
  - If all else fails try running `yarn nuke` followed by the `yarn` command.
  - If you are getting a

```
AADSTS50105: Your administrator has configured the application Enterprise Git to
block users unless they are specifically granted ('assigned') access to the application
```

error when trying to access [git.autodesk.com](https://git.autodesk.com), try waiting up to an hour to let the source code policy to take effect.