

Process for Publishing a FHIR IG

- 1 [Publishing a FHIR IG](#)
- 2 [Pre-conditions](#)
- 3 [Record Keeping](#)
- 4 [Checks](#)
- 5 [Local Build](#)
- 6 [Publishing the IG](#)
- 7 [Milestone releases](#)
- 8 [Finishing the Publication Process](#)

Publishing a FHIR IG

This process is followed by anyone when publishing a FHIR IG to <http://hl7.org/fhir/>

Pre-conditions

To follow this process, you must have:

- permission from the FHIR Product Director to publish FHIR IGs (including write access to the [\[FHIR IG Publication Record\]](#))
- permission from FMG+TSC to publish the particular FHIR IGs
- an FTP account with access to the source for <http://hl7.org/fhir>
- permission to push changes to <https://github.com/FHIR/ig-registry>
- A local clone of the FHIR IG registry github repository <https://github.com/FHIR/ig-registry> to {registry}
- A local clone of the FHIR History Template github repository <https://github.com/HL7/fhir-ig-history-template> to {history}

You must also know how to run the IGPublisher locally

Record Keeping

All publication runs must be recorded in the [\[FHIR IG Publication Record\]](#)

When you being processing the publication request, create a new row in the sheet, and add the following columns:

- Operator : Your name
- IG Name : The formal name for the ballot or publication request
- Publication Request: a link to the publication request that initiated the work (a link to confluence)
- IG approval: a link to the IG approval by FMG (a link to confluence - if there isn't one, consult the FHIR product director and put "(n/a)")
- PPS Link: a link to the original project approval (either confluence or project manager - if there isn't one, consult the TSC co-chair)
- TSC Approval: a link to the TSC approval (if this is a formal publication, otherwise blank)
- Primary Editor : the name of the editor who will handle any QA issues
- IG Source : The github repository URL where the source is located - Check that the repository URL is HL7 (or an other wise approved Organization - consult CTO for approved list)
- Branch: the branch that is being published (defaults to master)
- new IG : "yes" if this is the first time that any version is being published for this IG (whatever version - draft or ballot)
- Milestone?: "yes" if this a formal publication release (usually approved by TSC)

Checks

The following checks must be run. All these checks confirm that the various HL7 paper work which lives all over the place is all lined up, and that the IG correctly identifies itself as approved. Note that these records are all over the place, and it's really hard to work to get everything aligned, so expect to find problems.

0. Preparation

- Find the IG (based on it's github URL org and name) in <https://fhir.github.io/auto-ig-builder/builds.html>, and check that the IG was built in the last 24 hours. If not, rebuild it. Wait for it to rebuild (refresh the page, or watch <https://chat.fhir.org/#narrow/stream/179297-committers.2Fnotification>). Now find the QA Page.

1. Realm

- IG Approved realm: enter the realm that FMG approved when the IG was approved
- Project Realm: enter the realm that was approved for the project under which this IG is approved. If the project realm does not match the IG realm, consult the FHIR Product Director or the TSC co-chair
- Requested Realm - enter the realm that was specified in the publication request
- IG Actual realm - enter the realm stated for "Realm Rules" in the QA page. Make sure it is lowercase. If it's not, or either the stated or the actual realm doesn't match the approved realm, consult the FHIR Product Director

2. Code

- IG Approved Code: add the code as identified in the IG Approval page (note that the name of the github repository does not need to be the same)
- IG Actual Code: add the code as stated in the QA page ("Publication Code"). If there is an error shown, or if the codes does not match what was approved, consult the FHIR product director (important: this code is case sensitive - make sure you check the case)

3. Version

- Requested Version: enter the version requested in the publication request (should be in Requested name for published standard). If this is not specified in the publication request, then talk to the nominated editor.
- Stated Milestone: if this is a milestone, the stated version (e.g. "STU 1") - you can get this from the version check in the QA file. If it is not populated, or doesn't match the publication request, consult the editor and/or TSC co-chair
- Actual Version: copy this from the version specified in the "Version Check" row in the QA page. if there are any errors shown, or it doesn't match the requested version, consult the FHIR Product Director

4. Publishing Integrity

- fill out the canonical (<http://hl7.org/fhir/{realm}/{code}>)
 - If this is a new IG: Look up the URL {canonical}/package-list.json and check that it doesn't exist
 - if this is not a new IG: Look up the URL {canonical}/package-list.json and check that the canonical url and packageId and the name of the IG in the package list agree with the details for the IG (canonical and packageId must be identical, name can vary a little). Also check that there are no single quotes in the description; use ASCII equivalent "'" instead.
- **(up to here)**
- and packageId (hl7.fhir.{realm}.{code}) values from the Publication Row in the qa page
- Realm: add the realm code identified in the IG approval page. Check that this is consistent with the stated scope of the ballot and in the TSC approval
- Code:
- now check the qa page (rightmost link on the IG Builds page).
 - check that the packageId and canonical URL in the first line are correct against the spreadsheet, and check that the stated path starts with {canonical}
 - Review any errors in the page, paying particular attention the publication QA checks at the top
 - If there are any errors, refer to the FHIR product Director for approval
 - Once ok, record "yes (N)" in Build QA column, where N is the number of errors
- in the version column, record the version as stated in the first line (after the {packageId#} part
 - if this is a new IG: the version should be 0.1.0, or as approved by the FHIR Product Director
 - if this is a not a new IG: check that the FHIR Product Director has approved the version
- in the subDir column, record the subfolder stated in the publication path (in the current entry in the package-list.json file)
 - project team needs to add a new entry if one does not exist since the last ballot
 - If it does not conform to the pattern YYYYMMM (with the date correct) or STU(X), check with the FHIR product director
- Now review the actual guide (left most link in the IG build page)
 - Check that the header conforms to the [FHIR Implementation Guide Publishing Requirements](#):
 - check that the header shows the name of the IG and the ballot status correctly.
 - check that the header shows the correct HL7 logo
 - check that the header includes the yellow publish box that displays correctly (don't worry about what it actually says)
 - if all checks out, "yes" in the header column
 - Check that the footer conforms to the [FHIR Implementation Guide Publishing Requirements](#):
 - check that the following items are shown in the footer: packageId, version, FHIR version, publishing committee, history link, propose changes link, Copyright (© 20YY+), generated date
 - if all checks out, "yes" in the footer column
 - Check that the download package is linked (from index or downloads page)
- check that {registry} and {history} are up to date (git PULL)

Local Build

Now that the main checks have been done, you need to do a local build. To do that, you need a copy of the IG Publisher - see [IG Publisher Documentation](#) for assistance

- clone the github repository to your local drive = {clone}
 - you should call it {packageId}#{version}
- check the package-list.json
 - must include 'category'; if not, editors should consult Grahame
 - ensure package id, version are correct, update date
- run the IG publisher, clearing the terminology cache (run the publisher.jar -ig {clone} -resetTx -publish {url})
 - where {url} is the url that the guide will be published at, which is usually {canonical}/{subDir} - the subdirectory from above. If you are not sure about this valid, consult Grahame
- check that the local {clone}/output/qa.html file matches the CI build qa.html
 - troubleshooting on <https://chat.fhir.org/#narrow/stream/179165-committers>
 - once it matches, "yes" in the local qa column
- Zip up the folder {clone} as it is to a file named {packageId}#{version}.zip and ftp to <ftp://hl7.org/fhir/ig-build-zips>

You are now good to publish the IG

- check that the {clone}/output does not contain a file named package-list.json

Publishing the IG

Note: if you are publishing a set of IGs, you can do this part of the process - and all that follows as a batch

- create a local directory = {root}
- use FTP to copy the current contents of <ftp://hl7.org/fhir> to {root}
 - 1 million+ files. Keep this maintained in advance, or give 48 hours + to build it
 - Check with other publishers before making the copy (publishers = Grahame & Lynn)
 - If you keep this maintained, use FTP to sync your copy each time the contents of <http://hl7.org/fhir/package-feed.xml> changes (subscribe with a feed reader), or as discussed between publishers
- in your local directory
 - if this is a new IG: create the folder {root}/{realm}/{code}, and copy in {clone}/package-list.json (e.g. to {root}/{realm}/{code}/package-list.json)
 - if this is not a new IG: edit the file {root}/{realm}/{code}/package-list.json and add the entry for the {version} found in {clone}/package-list.json
- set the date in {root}/{realm}/{code}/package-list.json entry for {version} to today's (US) date using the format yyyy-mm-dd
 - if this is a new IG: copy the contents of {history} into {root}/{realm}/{code} (except for the [readme.md](#) file), and edit the name in index.html (replace 'XXXXXX')
- create the folder {root}/{realm}/{code}/{subdir}
- copy the content from {clone}/output to {root}/{realm}/{code}/{subdir}
- mark "yes" in the Copied column

Milestone releases

If you are doing a mile stone release (a STU publication of the IG), some additional steps are required:

- delete all files and directories in {root}/{realm}/{code}: publisher.jar -delete-current {root}/{realm}/{code} -history {history}
- re-run the IG publisher: publisher.jar -ig {clone} -publish {url} -milestone
 - check it succeeds
- copy the content from {clone}/output to {root}/{realm}/{code}. Say "yes" to overwrite any existing files
- update the package-list.json to set "current": true on the new version, and remove that from any other past published version (but not from the current build entry)
- mark "yes" in the Milestone column

Finishing the Publication Process

- Run the IGPublisher on {root}: publisher.jar -publish-update -folder {root} -registry {registry}/fhir-ig-list.json
 - Note: if you are doing multiple IGs, you can do this step once for the whole group (do everything to this point for all files), but the remaining steps must be done for each IG
- check the output for the IG(s) being changed: it should report that it changed N files for each published version of the IG (including previously published versions)
 - if problems, consult FHIR product director (and start again from the creating a local root directory)
- if you are doing a milestone release: use FTP to delete the contents of <ftp://hl7.org/fhir/{realm}/{code}>: yes, delete all the files
- use FTP to copy the entire contents of {root}/{realm}/{code} to <ftp://hl7.org/fhir/{realm}/{code}> - no need to upload files that have not changed
- mark "yes" in the FTPed column
- FTP the files {root}/package-feed.xml to <ftp://hl7.org/fhir/package-feed.xml> and {root}/publication-feed.xml to <ftp://hl7.org/fhir/publication-feed.xml>
- check the changes to {registry}/fhir-ig-list.json (use git diff) and fill out any missing information (look for ?? in the changes) - may need to talk to FHIR product director to decide what to put
- check that the fhir-ig-list.json file is valid using <https://jsonlint.com/>
- commit and push the changes to the FHIR IG registry to github to a new branch, let Grahame know, & mark "yes" in the Register IG column
 - Note: if you are doing multiple IGs, you only need to do the previous 3 steps once
- if this is a milestone publication, let Lloyd know that it's time to mark applied tasks as published, and considered for future use tasks as open
- if this is a milestone publication (not for ballots) make an announcement on the <https://chat.fhir.org/#narrow/stream/179240-Announcements/topic/FHIR.20Publication.20Announcement> topic in Zulip using the following template:

```
new Publication: [publication name] of the [IG Name] implementation guide: {canonical}/{subdir}
```

e.g. New Publication: STU Update 3.2 of the QI-Core implementation guide: <http://hl7.org/fhir/us/qicore/STU32/>