Implementation Guide Parameters

This page documents the parameters that the IG publisher defines for use in ImplementationGuide.definition.parameter.

Parameter	Repeats?	Type	Documentation
logging	yes	code	Which kinds of log output messages to produce - the value is a code, from this list: • **init**: Messages describing the start up process (for debugging) • **progress**: Overall progress messages • **tx**: Messages describing the use of the terminology server (for debugging) • **generate**: Log when producing an individual resource (troubleshooting only - leave off) • **html**: Log when validating an html page (troubleshooting only - leave off) Default = none
generate	yes	code	Fine grained control over generation of resources: • **example-narratives**: generate narrative in examples if they don't have any (as opposed to conformance resources, which are always generated) • **genExamples**: produce extra examples (todo: document this) Default = none
path-resource	yes	path	Relative path to a location in the IG repository from which to load resources. Default = "\resources"
path-binary	yes	path	Relative path to a location in the IG repository in which to look for binary adjunct files (see below)
autoload-resources	no	true|false	If true, scan the locations in which resources might be found (above) looking for resources, and auto-loading them. Else, only load what is in the IG. Default = false
codesystem-property	yes	uri	A code system property to show when rendering code systems (code system property definitions with a 'rendered-value' extension are always shown, as is any property in the uri space http://hl7.org/fhir/concept-properties
path-pages	yes	path	A relative path in which pages can be found. Default = "\pages"
path-qa	no	path	The relative path to the qa folder (generated fragments to check rendering in browser). If not specified, no qa is produced. Default = "\qc"
path-tx-cache	no	path	The relative path to the terminology cache. (see (todo) for further details). default = no value - txCache not in version control (slower builds)
path-liquid	yes	path	A relative path that contains liquid templates for generating narrative
path-temp	no	path	The relative path to use for jekyll processing. Defaults to "\temp"
path-output	no	path	The relative path where the output is generated. Defaults to "\output", which must be the case for the ci-build scripts
path-history	no	url	The url where the history file is found (usually "(canonical)/history.html"). Used when checking links
path-expansion-params	no	path	The relative path to a parameters resource that contains expansion parameters used when the IG publisher expands a value set
path-suppressed-warnings	yes	path	A file to load suppressed error messages from (one msg per line). Suppressed errors don't chose up in the qa.ini (but will be checked anyway if publishing through HL7 processes)
html-exempt	yes	string	A mask that identifies specific HTML files exempt from having header / footer etc. Discuss with FHIR Product Manager before using in any HL7 IG
extension-domain	yes	url	An external domain fro which extensions are allowed to come (else they'll be flagged as invalid when validating). http://example.org is always allowed
active-tables	no	true|false	Whether to generate active tables in the generated fragments (tables that will expand and contract). Note that turning this on requires that this entry be found somewhere in the header of the generated html pages: Default = false `<script src="fhir-table-scripts.js"></script>`
ig-expansion-parameters	yes	code	? broken - needs reworking
special-url	yes	url	If a canonical resource in the IG should actually have a URL that isn't the one implied by the canonical URL for the IG itself, it must be listed here explicitly (as well as defined in the resource itself). It must be listed here to stop it accidentally being different. Each canonical url must be listed in full as present on the resource; it is not possible to specify a pattern.
template-openapi	no	path	A relative path to the template openapi file to use when generating openapi files (e.g. populate openAPI security details)
template-html	no	path	A relative path to the template to use when rendering html pages
template-md	no	path	A relative path to the template to use when rendering markdown pages

apply-contact	no	true|false	if true, overwrite all canonical resource contact details with that found in the IG. Default = false
apply-context	no	true|false	if true, overwrite all canonical resource context details with that found in the IG. Default = false
apply-copyright	no	true|false	if true, overwrite all canonical resource copyright details with that found in the IG. Default = false
apply-jurisdiction	no	true|false	if true, overwrite all canonical resource jurisidiction details with that found in the IG. Default = false
apply-license	no	true|false	if true, overwrite all canonical resource license details with that found in the IG. Default = false
apply-publisher	no	true|false	if true, overwrite all canonical resource publisher details with that found in the IG. Default = false
apply-version	no	true|false	if true, overwrite all canonical resource version details with that found in the IG. Default = false
validation	yes	code	Fine grained control over validation flags when validating resources / IGs: • **check-must-support**: Does nothing! (for now?) • **allow-any-extensions**: Allow any extensions at all (and therefore miss accidental broken extension references) • **check-aggregation**: check that aggregation is correct (in IGs, bundles are often broken up for publishing / documentation purposes) • **no-broken-links**: don't report broken links in HTML pages (note: never turn this off for HL7 guides. Stay on top of fixing broken links; it's a chore, but if you leave it till later, you won't have time, and your IG won't be approved for publishing) • **show-reference-messages**: when validating, instead of showing a summary of issues from validating the target of references, just show all the validation messages in detail Default = none of these options
show-inherited-invariants	no	true|false	If true, render inherited constraints in the full details and invariants view. Default = true
usage-stats-opt-out	no	true|false	If true, usage stats (information about extensions, value sets, and invariants being used) is not sent to fhir.org (see e.g. http://clinfhir.com/igAnalysis.html). Default = false

Managing Warnings and Hints

The implementation guide publisher generates many Warnings and information messages. They may arise from:

- Errors in the design of the Implementation Guide
- Bugs in the publication tooling
- Necessary outcomes from the specification (not an issue at all)

Every one of these messages needs to be checked. When there are a large number of these, it can be difficult for someone reviewing the qa page to distinguish which warnings they need to pay attention to and which are new/important. Once the message has been checked and is known to be not an issue needing resolution, the message can be added to the suppressed warnings file.

This file has the following format:

```
== Suppressed Messages ==

# {{Reason}}
{{message}}
{{message}}
```

Each {{message}}line contains the message to be suppressed - copy and paste from the QA page. Lines are grouped by # {{Reason}} which explains why the message is considered ok (not an error)

e.g.

```
# The following code systems are external and not supported by terminology server
Code System URI 'http://www.ama-assn.org/go/cpt' is unknown so the code cannot be validated
```

This mechanism can suppress information messages, warning messages and errors related to broken links (e.g. if a link is created to a file produced during post-processing or that otherwise isn't resolvable during the IG publication process).

Binary Adjunct Files

For several resource types, it can be more convenient to edit base64Binary content directly. The obvious candidate resources are:

- Binary
- Library
- DocumentReference

But there are many other resources where this might be useful. The technique described here will work for any Resource with Attachments in it.

Here's how to make it work:

- Create the resource as per normal
- For the attachment resource, leave it empty, and provide only an id property

- the id has the special value "ig-loader-[filename]", where filename is the name of the file to load. e.g. "id" : "ig-loader-MyLibrary.cql" (case sensitive)
- The IG Publisher will scan all the folders identified by the path-binary parameter (see above), and remove the id property, and replace with with the content from the file
- If a matching file cannot be found, the id will be left in place, and an error will be logged in the qa page

Binary is a special case. For Binary, make the content of the data itself equal to "ig-loader-[filename]"

Note: the following file extensions are supported:

- .txt (text/plain)
- .pdf (application/pdf)
- .jpg, .png, .gif (image/xx)
- .dicom - application/dicom
- .cql - text/cql. Note that CQL included into a library is subject to additional processing. A second content will be injected for application/elm+xml - do not create a stub for this manually)