

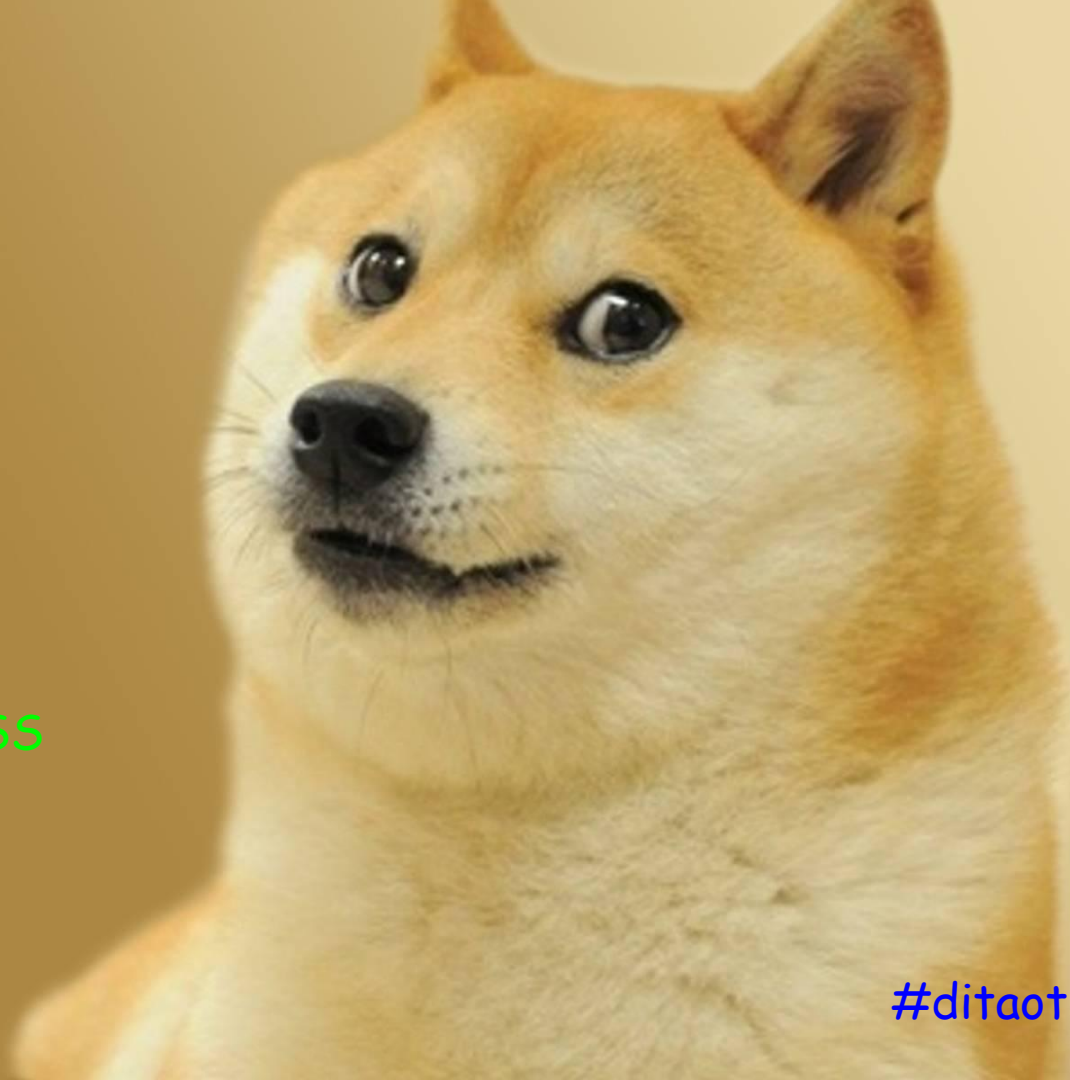
Coffee break 10:30-11:00

WOW

SUCH DITA-OT

MUCH PREPROCESS

#ditaotday



TL;DR



Vanity Slide

Jarno Elovirta

jarno@elovirta.com @jelovirt <https://github.com/jelovirt>

WUNDERDOG[®]

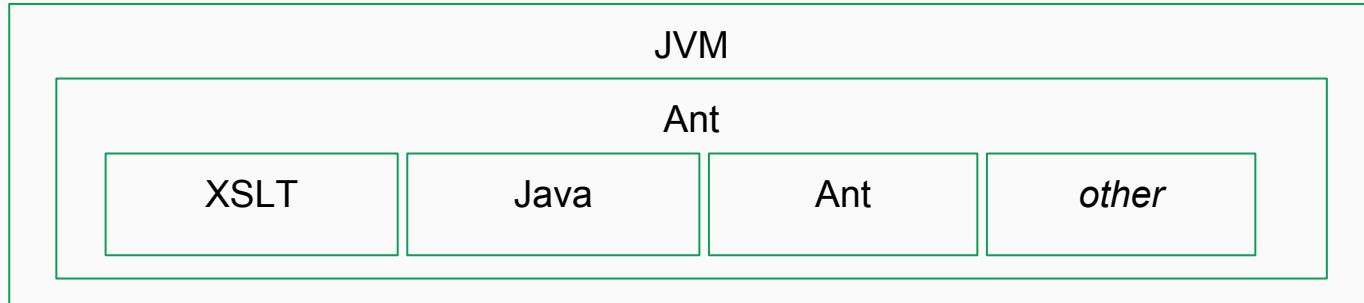
Agenda

- DITA-OT at block level
- Disassembling preprocessing
- Expected work for 3.0
- Questions



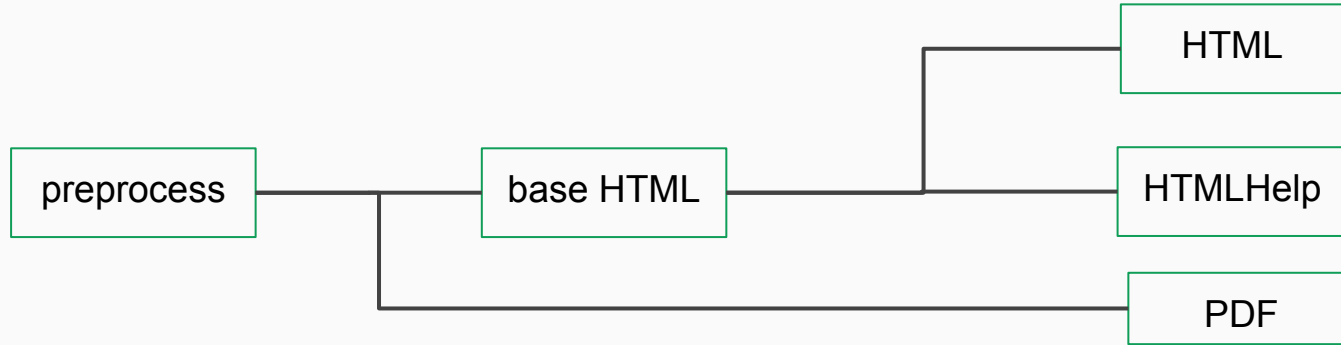
Technology stack

- DITA-OT is a DITA processor implemented in Java, Ant, and XSLT



Processing modules and stages

- Instead of a single operation, input is processed in consecutive stages



- Preprocess step is common to all transtypes

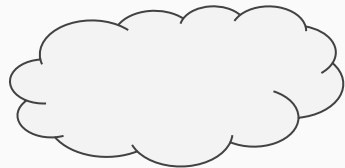
Preprocessing

- Preprocessing prepares the content for transtype-specific processing
- Handles most of the functional features of DITA, like keyref or conref
- DITA spec doesn't mandate a processing order

TL;DR: Make complex DITA structures simple DITA structures

Generate lists & debug and filter (1/2)

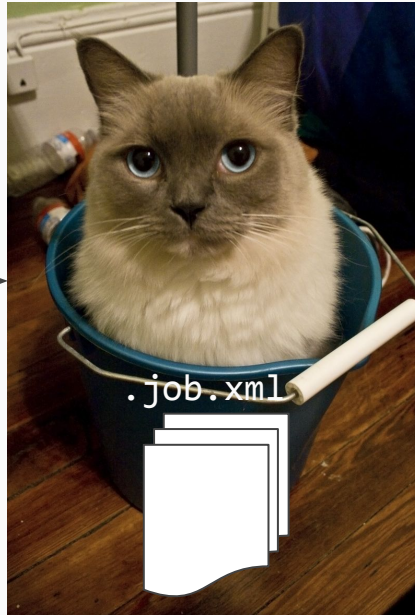
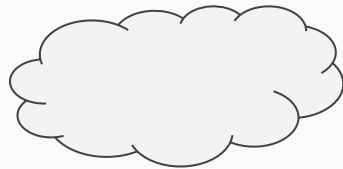
- Initial processing steps that recursively read input and linked resources
- Collect file metadata what each DITA topic or map contains



- **gen-list**
- debug-filter
- mapref
- branch-filter
- copy-files
- keyref
- copy-to
- conrefpush
- conref
- profile
- topic-fragment
- chunk
- move-meta-entries
- maplink
- topicpull
- flag-module
- clean-map

Generate lists & debug and filter (2/2)

- Debug and filter step cleans, normalises, validates, and serialises DITA into temporary directory
- By default also filters DITA content
- Since 2.4 copy-to processing has been split off



- gen-list
- **debug-filter**
- mapref
- branch-filter
- copy-files
- keyref
- copy-to
- conrefpush
- conref
- profile
- topic-fragment
- chunk
- move-meta-entries
- maplink
- topicpull
- flag-module
- clean-map

Resolve map references

- Resolves references from one DITA map to another
 - Creates a single map that contains all topicrefs and reltables for all maps
- gen-list
 - debug-filter
 - **mapref**
 - branch-filter
 - copy-files
 - keyref
 - copy-to
 - conrefpush
 - conref
 - profile
 - topic-fragment
 - chunk
 - move-meta-entries
 - maplink
 - topicpull
 - flag-module
 - clean-map

Branch filtering

- Filters topics using DITAVAL files defined in the map



- gen-list
- debug-filter
- mapref
- **branch-filter**
- copy-files
- keyref
- copy-to
- conrefpush
- conref
- profile
- topic-fragment
- chunk
- move-meta-entries
- maplink
- topicpull
- flag-module
- clean-map

Copy related files

- Copies non-DITA resources into output.
 - Which files are copied depends on transtype configuration.
- gen-list
 - debug-filter
 - mapref
 - branch-filter
 - **copy-files**
 - keyref
 - copy-to
 - conrefpush
 - conref
 - profile
 - topic-fragment
 - chunk
 - move-meta-entries
 - maplink
 - topicpull
 - flag-module
 - clean-map

Resolve key references

- Resolves keyrefs and conkeyrefs to populate link URIs and text replacement.

```
<xref keyref="x">
```



```
<xref href="foo.dita">
```

```
<p conkeyref="y/x">
```



```
<p conref="b.dita#b/x">
```

- gen-list
- debug-filter
- mapref
- branch-filter
- copy-files
- **keyref**
- copy-to
- conrefpush
- conref
- profile
- topic-fragment
- chunk
- move-meta-entries
- maplink
- topicpull
- flag-module
- clean-map

Copy topics

- Makes a copy of original topic resources to new resources defined by @copy-to attribute

```
<topicref href="foo.dita" copy-to="bar.dita">
```



```
<topicref href="bar.dita">
```



- gen-list
- debug-filter
- mapref
- branch-filter
- copy-files
- keyref
- **copy-to**
- conrefpush
- conref
- profile
- topic-fragment
- chunk
- move-meta-entries
- maplink
- topicpull
- flag-module
- clean-map

Resolve content references

- Processes both push and pull content references

```
<p conref="b.dita#b/x">
```



```
<p id="x">Resolved content</p>
```



- gen-list
- debug-filter
- mapref
- branch-filter
- copy-files
- keyref
- copy-to
- **conrefpush**
- **conref**
- profile
- topic-fragment
- chunk
- move-meta-entries
- maplink
- topicpull
- flag-module
- clean-map

Filter conditional content

- Removes content from topics and maps based on DITAVAL or print attribute
- Output can differ based on when filtering is done

```
<li audience="pro">duct tape</li>
```

```
<li>cable tie</li>
```



```
<li>cable tie</li>
```

- gen-list
- debug-filter
- mapref
- branch-filter
- copy-files
- keyref
- copy-to
- conrefpush
- conref
- **profile**
- topic-fragment
- chunk
- move-meta-entries
- maplink
- topicpull
- flag-module
- clean-map

Resolve topic fragments 1/2

- Expands same topic fragments in URIs

```
<xref href="#./x">
```



```
<xref href="#b/x/>
```



- gen-list
- debug-filter
- mapref
- branch-filter
- copy-files
- keyref
- copy-to
- conrefpush
- conref
- profile
- **topic-fragment**
- chunk
- move-meta-entries
- maplink
- topicpull
- flag-module
- clean-map

- Expands coderef references in codeblock elements

```
<codeblock coderef="for.scala"/>
```



```
<codeblock> for (i <- 0 to 10) {  
  println(i)  
}</codeblock>
```

- Adds some extensions to DITA spec

```
<coderef href="unicode.txt"  
  format="txt; charset=UTF-8"/>  
<coderef  
  href="lib.rs#token=XMPSTR,XMPEND"/>
```

- gen-list
- debug-filter
- mapref
- branch-filter
- copy-files
- keyref
- copy-to
- conrefpush
- conref
- profile
- **topic-fragment**
- chunk
- move-meta-entries
- maplink
- topicpull
- flag-module
- clean-map

Chunk topics

- Breaks apart and assembles referenced DITA content based on the chunk attributes in maps



- gen-list
- debug-filter
- mapref
- branch-filter
- copy-files
- keyref
- copy-to
- conrefpush
- conref
- profile
- topic-fragment
- **chunk**
- move-meta-entries
- maplink
- topicpull
- flag-module
- clean-map

Move map metadata to topics & pull content into maps

- Cascades metadata in map and nested topicrefs
 - Pushes the map metadata into topics
 - Allows topic processing in isolation while retaining all relevant metadata
 - Pulls metadata from referenced topics into maps
- gen-list
 - debug-filter
 - mapref
 - branch-filter
 - copy-files
 - keyref
 - copy-to
 - conrefpush
 - conref
 - profile
 - topic-fragment
 - chunk
 - **move-meta-entries**
 - maplink
 - topicpull
 - flag-module
 - clean-map

Map-based linking

- Collects links based on a map and moves those links into the referenced topics



- gen-list
- debug-filter
- mapref
- branch-filter
- copy-files
- keyref
- copy-to
- conrefpush
- conref
- profile
- topic-fragment
- chunk
- move-meta-entries
- **maplink**
- topicpull
- flag-module
- clean-map

Pull content into topics

- Pulls title and description content into xref and link elements
- Partially overlaps with transtype specific link processing

```
<xref href="o-sensei.dita"/>
```

↓

```
<xref href="o-sensei.dita"  
  type="concept"  
>植芝 盛平</xref>
```

- gen-list
- debug-filter
- mapref
- branch-filter
- copy-files
- keyref
- copy-to
- conrefpush
- conref
- profile
- topic-fragment
- chunk
- move-meta-entries
- maplink
- **topicpull**
- flag-module
- clean-map

Flagging

- Evaluates the DITAVAL for flag action and adds DITA-OT specific elements to topics when flags are active
 - Any extended transform type may use these hints to support flagging without adding logic to interpret the DITAVAL
- gen-list
 - debug-filter
 - mapref
 - branch-filter
 - copy-files
 - keyref
 - copy-to
 - conrefpush
 - conref
 - profile
 - topic-fragment
 - chunk
 - move-meta-entries
 - maplink
 - topicpull
 - **flag-module**
 - clean-map

- Any elements and attributes that were added to files to support preprocessing are removed



- gen-list
- debug-filter
- mapref
- branch-filter
- copy-files
- keyref
- copy-to
- conrefpush
- conref
- profile
- topic-fragment
- chunk
- move-meta-entries
- maplink
- topicpull
- flag-module
- **clean-map**

Future work



Alternative work cache URIs

- Change how source URIs are mapped to temporary files
- Default implementation matches old behaviour, relative 1:1 mapping
- Alternative implementations:
 - Hash: 0d23e72be0377ab0e899791a9cbf2f0613813c18.dita)
 - Full path: Users/jessicajones/Work/season2/script.dita)
- Job configuration is used to store the mapping

Map-first preprocess

- Split preprocess into three phases:
 - Read maps and process as far as possible
 - Read topics and process as far as possible
 - Process map and topic relations
- End result is the same as in old preprocess
- Easier to reason about the process and clear separation of steps
- Allows combining filter processing

Alternative work caches

- Support memory-based temporary storages
- Support alternative storage formats



Thank you