# DITA-OT
# Under the Hood

DITA-OT Day 2014
Jarno Elovirta
jarno@elovirta.com @jelovirt https://github.com/jelovirt

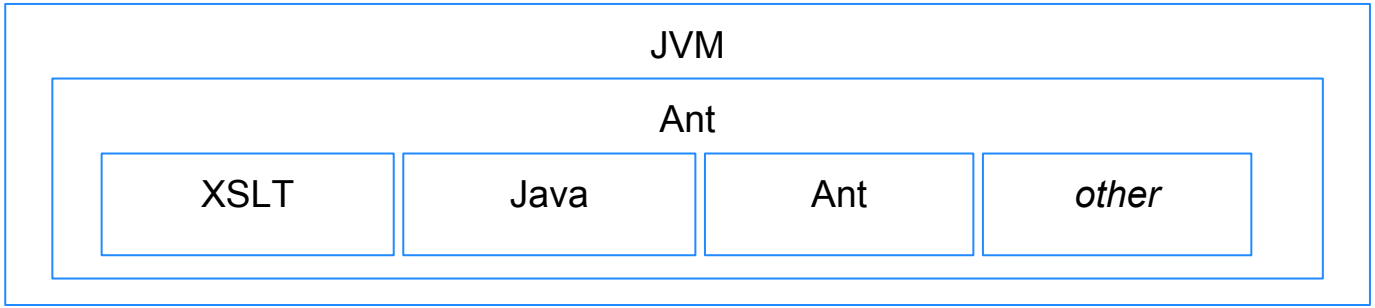
# TL;DL

# Agenda

- DITA-OT at block level
- Disassembling preprocessing
- Future work and possibilities
- Questions

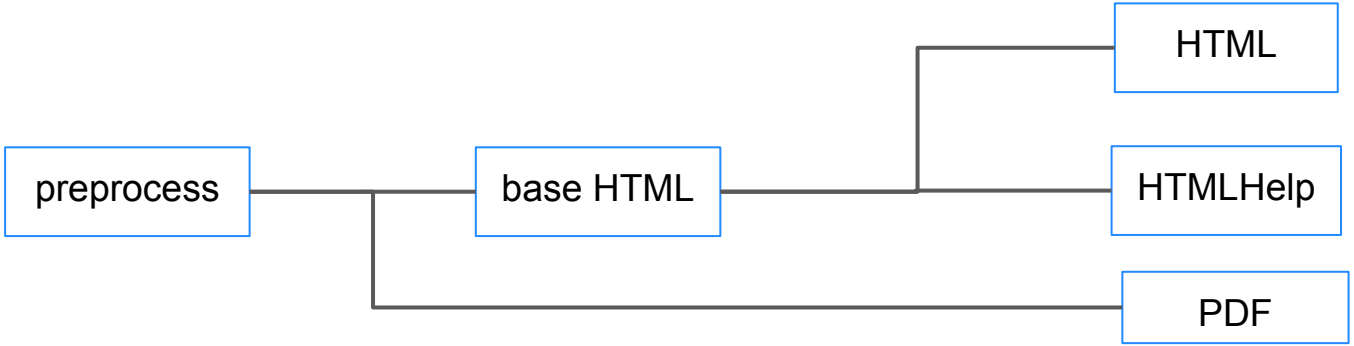
# Technology stack

- 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

# Generate lists & debug and filter (1/2)

- Initial processing steps that recursively read input and linked resources
- Generate lists step collects information what each DITA topic or map contains

# Generate lists & debug and filter (2/2)

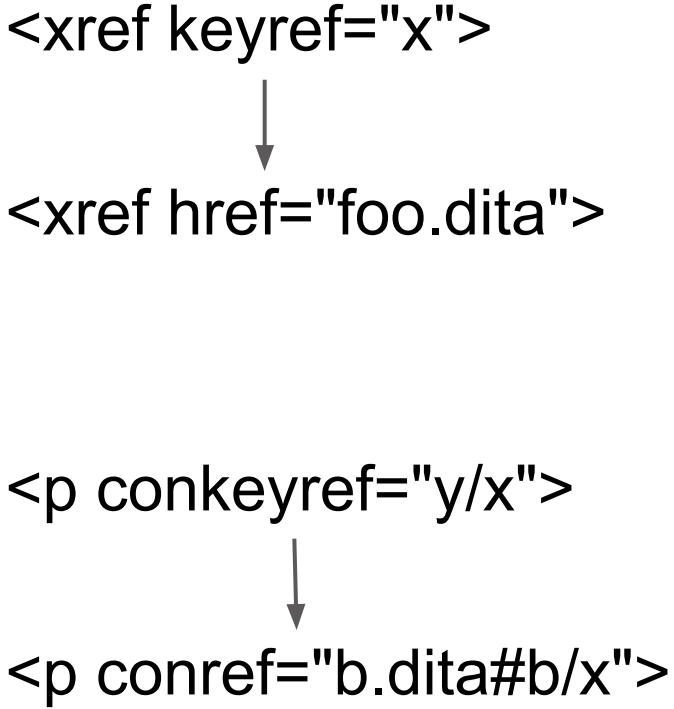
- Debug and filter step cleans, normalises, validates, and serialises DITA into temporary directory
- By default also filters DITA content
- Processes implicit and explicit copy-to

# Copy related files

- Copies non-DITA resources into output.
- Which files are copies depends on transtype configuration.

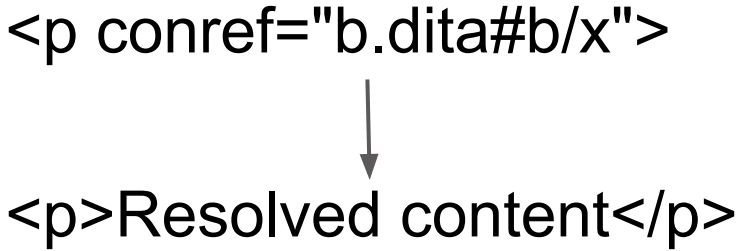
# Resolve key references

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



# Resolve content references

- Processes both push and pull content references
- Resolves links and generates IDs as needed

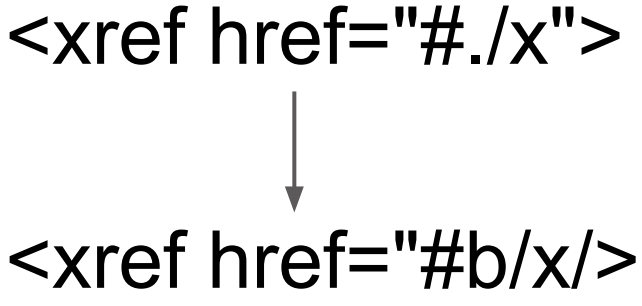


# Filter conditional content

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

# Resolve topic fragments

- Expands same topic topic fragments in URIs
- New processing for DITA 1.3



# Resolve code references

- Expands `coderef` references in `codeblock` elements
- Adds some extensions to DITA spec

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

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

```
<coderef href="unicode.txt" format="
txt; charset=UTF-8"/>
```

# Resolve map references

- Resolves references from one DITA map to another
- Creates a single map that contains all topicrefs and reltables for all maps

# Move map metadata to topics

- Cascades metadata in map and nested topicrefs
- Pushes the map metadata into topics
- Allows topic processing in isolation while retaining all relevant metadata

# Pull content into maps

- Pulls content from referenced topics into maps
- Cascades metadata within maps

# Chunk topics

- Breaks apart and assembles referenced DITA content based on the chunk attributes in maps
- Some chunk tokens are only supported in given transtypes
- Generates new resources with configurable URI generation schemes

# Map-based linking

- Collects links based on a map and moves those links into the referenced topics
- The links are created based on hierarchy in the DITA map, the collection-type attribute, and relationship tables

# 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>
```

# 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

# Clean-up

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

# Future work

- Combine list generation and debug and filter
- Allow use of non-local resources like HTTPS URL or CMS proprietary URIs
- Use memory-based temporary storage or alternate serializations formats
- Implement DITA 1.3 features

# Thank you