

Konveyor / Tackle

Tackle Config Discovery



Konveyor Projects



Rehost virtual
machines to KubeVirt



Rehost apps
between Kubernetes
clusters



Replatform
applications to
Kubernetes



Refactor applications
for Kubernetes



Measure software
delivery performance

**You just broke up your monolith into
microservices!**



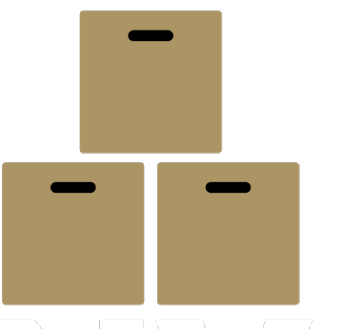
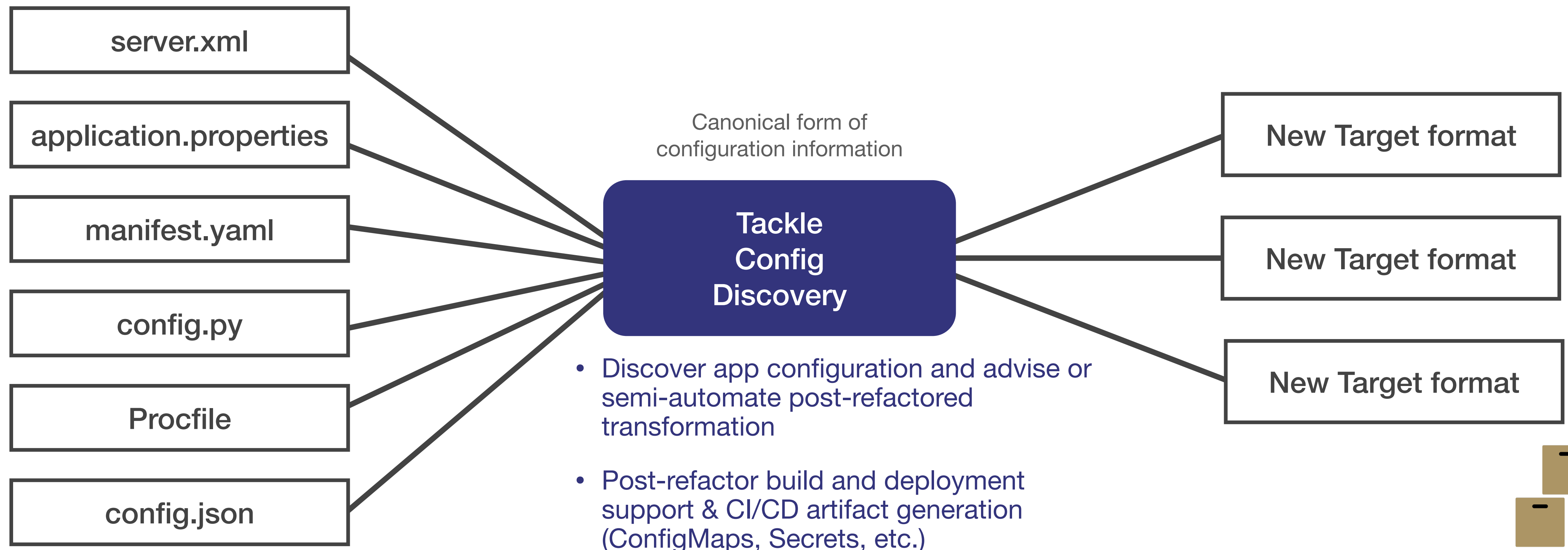
Where does the configuration info go?

Tackle Config Discovery



Overview

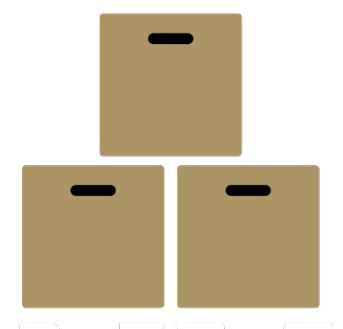
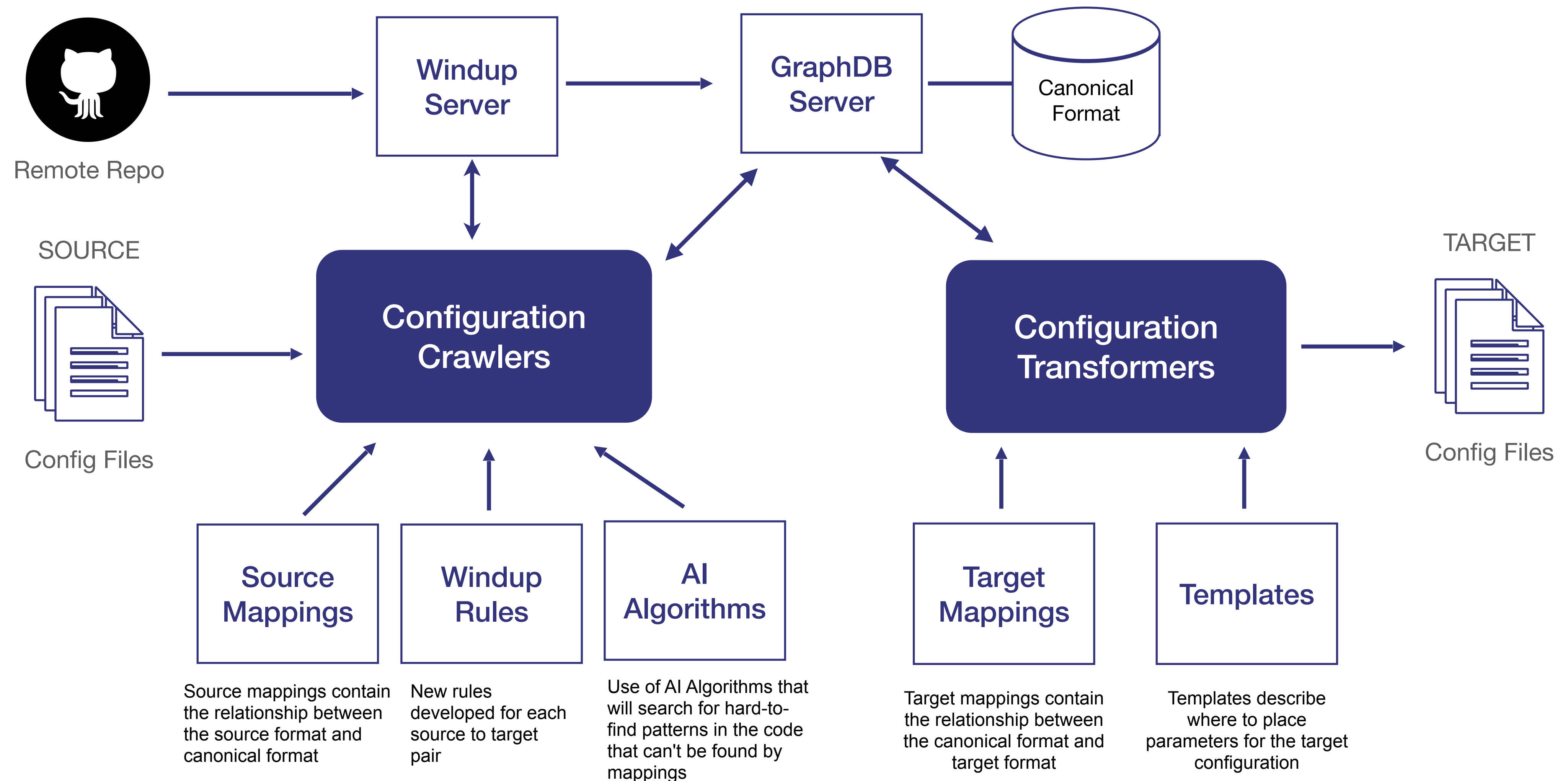
As an Architect I'd like to discover existing configuration parameters so that I can semi-automate the post-refactor build



Tackle Config Discovery



Architecture



KONVEYOR

Config Discovery Example

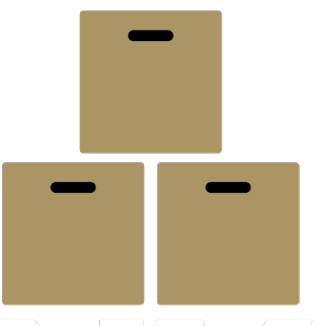


DayTrader 7 Java EE xml config file example:

```
./pom.xml
./daytrader-ee7/pom.xml
./daytrader-ee7/bin/main/META-INF/application.xml
./daytrader-ee7/src/main/application/META-INF/application.xml
./daytrader-ee7-ejb/pom.xml
./daytrader-ee7-ejb/bin/main/META-INF/ibm-ejb-jar-bnd.xml
./daytrader-ee7-ejb/bin/main/META-INF/persistence.xml
./daytrader-ee7-ejb/bin/main/META-INF/ejb-jar.xml
./daytrader-ee7-ejb/src/main/resources/META-INF/ibm-ejb-jar-bnd.xml
./daytrader-ee7-ejb/src/main/resources/META-INF/persistence.xml
./daytrader-ee7-ejb/src/main/resources/META-INF/ejb-jar.xml
./daytrader-ee7-web/pom.xml
./daytrader-ee7-web/bin/main/META-INF/persistence.xml
./daytrader-ee7-web/src/main/webapp/WEB-INF/ibm-web-bnd.xml
./daytrader-ee7-web/src/main/webapp/WEB-INF/classes/META-INF/persistence.xml
./daytrader-ee7-web/src/main/webapp/WEB-INF/beans.xml
./daytrader-ee7-web/src/main/webapp/WEB-INF/faces-config.xml
./daytrader-ee7-web/src/main/webapp/WEB-INF/web.xml
./daytrader-ee7-web/src/main/webapp/WEB-INF/ibm-web-ext.xml
./daytrader-ee7-web/src/main/java/META-INF/persistence.xml
./daytrader-ee7-wlpcfg/pom.xml
./daytrader-ee7-wlpcfg/servers/daytrader7Sample/server.xml
```

Summary

pom.xml	5
persistence.xml	5
application.xml	2
ejb-jar.xml	2
ibm-ejb-jar-bnd.xml	2
beans.xml	1
faces-config.xml	1
ibm-web-bnd.xml	1
ibm-web-ext.xml	1
server.xml	1
web.xml	1

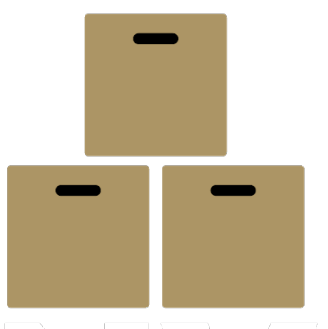
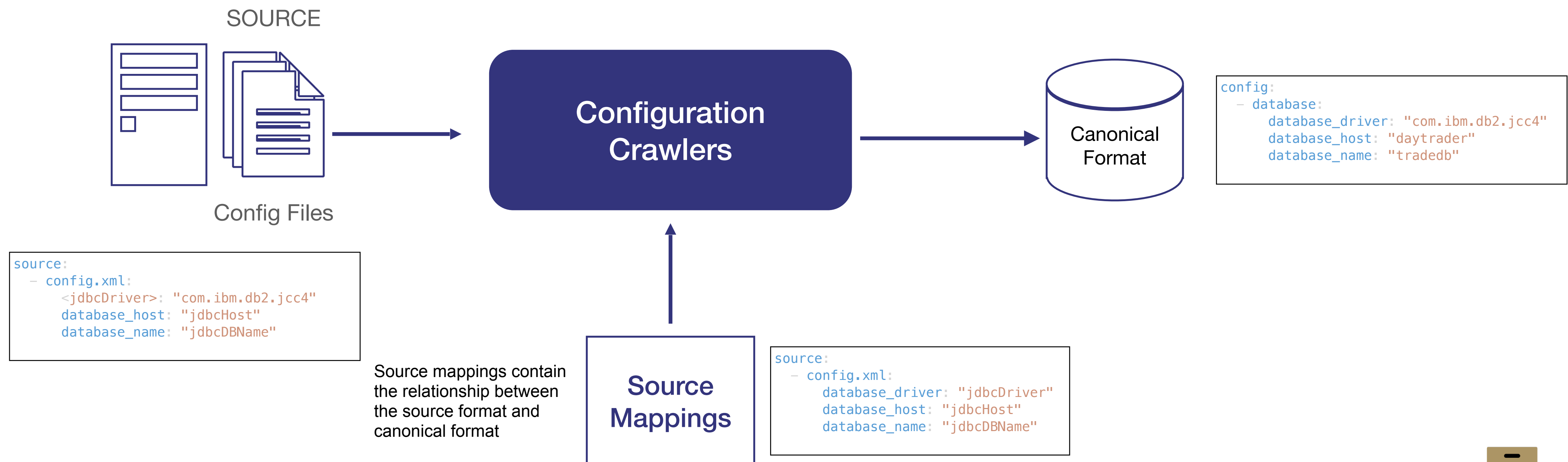


Configuration Crawlers



Source Mapping

Config crawler uses the source mapping file to determine how the values in each of the source files gets mapped into the canonical format and stores it.



Mapping Example

```
<server description="DayTrader Processor Server">

  <httpEndpoint httpPort="9080" httpsPort="9443" id="defaultHttpEndpoint" />

  <keyStore id="defaultKeyStore" password="Liberty" />

  <webApplication id="DayTraderProcessor" location="DayTraderProcessor.war"
    name="DayTraderProcessor"/>

  <connectionManager agedTimeout="-1" connectionTimeout="0" id="conMgr1"
    maxIdleTime="-1" maxPoolSize="100" minPoolSize="100"
    purgePolicy="FailingConnectionOnly" reapTime="-1"/>

  <jdbcDriver id="DB2Driver" libraryRef="DB2JCC4Lib"/>

  <library id="DB2JCC4Lib" filesetRef="DB2Fileset" />

  <fileset id="DB2Fileset" dir="${shared.resource.dir}/db2jars"
    includes="db2jcc4-10.1.jar"/>

  <authData id="TradeDataSourceAuthData" password="${env.DB_PASSWORD}"
    user="${env.DB_USER}"/>

  <dataSource jndiName="jdbc/TradeDataSource" jdbcDriverRef="DB2Driver"
    connectionManagerRef="conMgr1" id="DefaultDataSource"
    isolationLevel="TRANSACTION_READ_COMMITTED" statementCacheSize="60">
    <properties.db2.jcc databaseName="${env.DB_DATABASE}" serverName="${
      env.DB_HOST}" portNumber="${env.DB_PORT}" user="${env.DB_USER}"
      password="${env.DB_PASSWORD}" />
  </dataSource>
</server>
```

```
server:
  description: "DayTrader Processor Server"
  httpEndpoint:
    httpPort: 9080
    httpsPort: 9443
    id: defaultHttpEndpoint
  keyStore:
    id: defaultKeyStore
    password: Liberty
  webApplication:
    id: DayTraderProcessor
    location: "DayTraderProcessor.war"
    name: DayTraderProcessor
  jdbcDriver:
    id: DB2Driver
    libraryRef: DB2JCC4Lib
  library:
    id: DB2JCC4Lib
    filesetRef: DB2Fileset
  fileset:
    id: DB2Fileset
    dir: "${shared.resource.dir}/db2jars"
    includes: "db2jcc4-10.1.jar"
  authData:
    id: TradeDataSourceAuthData
    password: "${env.DB_PASSWORD}"
    user: "${env.DB_USER}"
  dataSource:
    jndiName: "jdbc/TradeDataSource"
    jdbcDriverRef: DB2Driver
    connectionManagerRef: conMgr1
    id: DefaultDataSource
    isolationLevel: TRANSACTION_READ_COMMITTED
    statementCacheSize: 60
    properties.db2.jcc:
      databaseName: "${env.DB_DATABASE}"
      serverName: "${env.DB_HOST}"
      portNumber: "${env.DB_PORT}"
      user: "${env.DB_USER}"
      password: "${env.DB_PASSWORD}"
```


Quarkus Database Config

application.properties

<https://quarkus.io/guides/all-config>



quarkus.datasource.jdbc	boolean
quarkus.datasource.jdbc.driver	string
quarkus.datasource.jdbc.transactions	enabled, xa, disabled
quarkus.datasource.jdbc.enable-metrics	boolean
quarkus.datasource.jdbc.url	string
quarkus.datasource.jdbc.initial-size	int
quarkus.datasource.jdbc.min-size	int
quarkus.datasource.jdbc.max-size	int
quarkus.datasource.jdbc.background-validation-interval	Duration
quarkus.datasource.jdbc.acquisition-timeout	Duration
quarkus.datasource.jdbc.leak-detection-interval	Duration
quarkus.datasource.jdbc.idle-removal-interval	Duration
quarkus.datasource.jdbc.max-lifetime	Duration
quarkus.datasource.jdbc.transaction-isolation-level	undefined, none, read-uncommitted, read-committed, repeatable-read, serializable
quarkus.datasource.jdbc.detect-statement-leaks	boolean
quarkus.datasource.jdbc.new-connection-sql.	string
quarkus.datasource.jdbc.validation-query-sql	string
quarkus.datasource.jdbc.pooling-enabled	boolean
quarkus.datasource.jdbc.additional-jdbc-properties	Map<String,String>



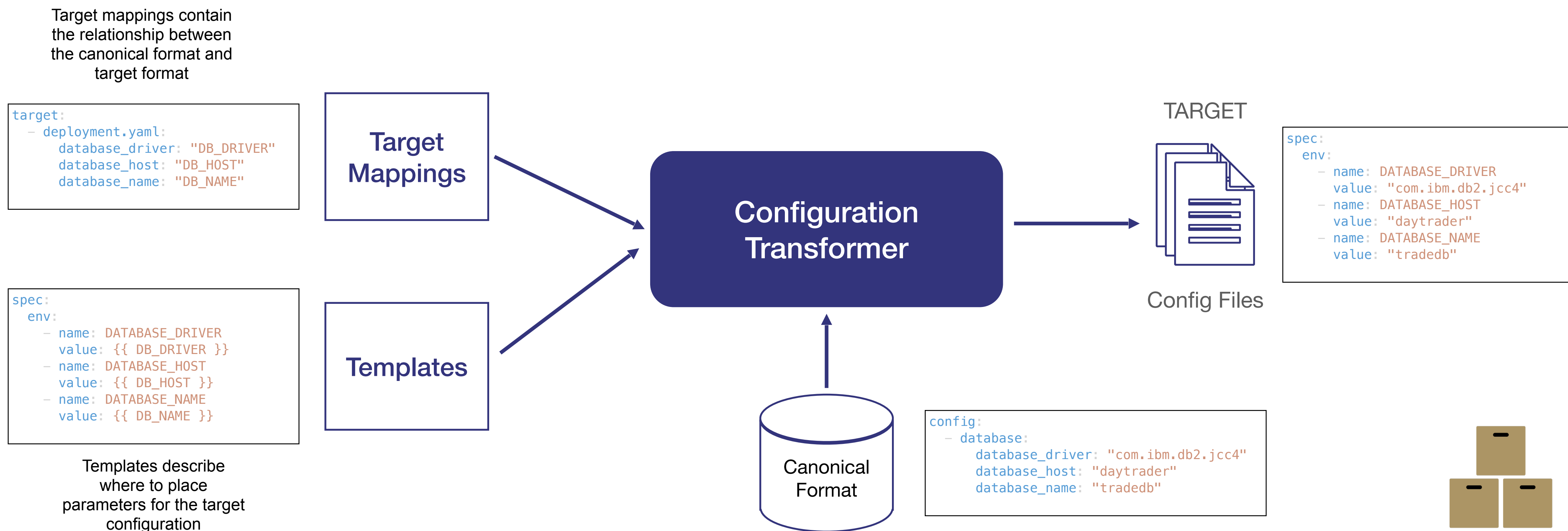
© 2020 Konveyor Tackle

Configuration Transformers



Artifact Generation

Config transformer takes the mapping file to get the configuration information out of the canonical store and uses the template file to create a new configuration file that inserts the correct values into the template



Live Demo

Convert SpringBoot to Quarkus

<https://github.com/spring-petclinic/spring-petclinic-rest>



The screenshot shows the GitHub repository page for `spring-petclinic/spring-petclinic-rest`. The repository is public and has 24 watchers, 292 stars, and 616 forks. It is currently on the `master` branch, which has 2 branches and 0 tags. The repository contains 8 issues, 3 pull requests, and 660 commits. The file list shows various configuration files and source code files, including `.github/workflows`, `.mvn/wrapper`, `src`, `.editorconfig`, `.gitignore`, `LICENSE.txt`, `mvnw`, `mvnw.cmd`, `petclinic-ermodel.png`, `pom.xml`, and `readme.md`. The right sidebar shows the repository's description, tags, and license.

Repository Information:

- Repository: `spring-petclinic/spring-petclinic-rest` (Public)
- Watchers: 24
- Stars: 292
- Forks: 616
- Issues: 8
- Pull requests: 3
- Commits: 660

File List:

File Name	Description	Last Commit
<code>.github/workflows</code>	Make clean install work again	3 days ago
<code>.mvn/wrapper</code>	Bump to spring-boot version 2.2.4.RELEASE, maven version 3.6.3	2 years ago
<code>src</code>	Make clean install work again	3 days ago
<code>.editorconfig</code>	#96 change EditorConfig in order to impact other files than Java an...	6 years ago
<code>.gitignore</code>	Easier branch switching	5 years ago
<code>LICENSE.txt</code>	Create LICENSE.txt	15 months ago
<code>mvnw</code>	Bump to spring-boot version 2.2.4.RELEASE, maven version 3.6.3	2 years ago
<code>mvnw.cmd</code>	Bump to spring-boot version 2.2.4.RELEASE, maven version 3.6.3	2 years ago
<code>petclinic-ermodel.png</code>	Bump libs to latest versions, improve /owners POST,PUT, add tests	2 years ago
<code>pom.xml</code>	Make clean install work again	3 days ago
<code>readme.md</code>	Remove old springfox swagger configuration and starting springdo...	14 days ago

About:

REST version of the Spring Petclinic sample application

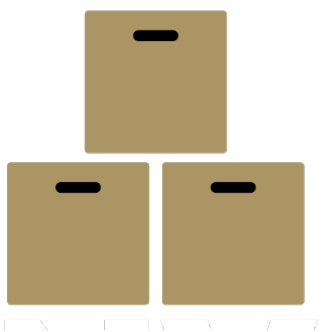
Tags:

- `sample`, `rest`, `spring`, `spring-data`, `jdbc`, `rest-api`, `springfox`, `swagger`, `jackson`, `hibernate`, `mysql-database`, `springframework`, `spring-mvc`, `spring-data-jpa`

License: Apache-2.0 License

Releases: No releases published

Packages:



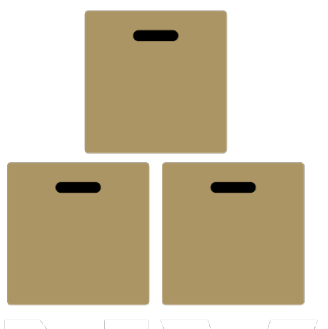
Convert SpringBoot to Quarkus

Several properties files in this repo



The screenshot shows the GitHub repository page for `spring-petclinic/spring-petclinic-rest`. The repository is public and has 24 watchers, 292 stars, and 616 forks. The current branch is `master`. The file list shows the following files and their commit history:

File	Commit Message	Time Ago
db	Upgrade to Spring Boot 2.1.5	2 years ago
messages	general cleanup	9 years ago
api-docs.yml	feat: Add api-docs.yml service contract	16 days ago
application-hsqldb.properties	close issue #12	4 years ago
application-mysql.properties	Replace deprecated property 'spring.datasource.initialize' with 'spri...	9 months ago
application-postgresql.properties	Replace deprecated property 'spring.datasource.initialize' with 'spri...	9 months ago
application.properties	Remove old springfox swagger configuration and starting springdoc con...	2 months ago
logback.xml	Fix #155 Fix logback + JMX memory leak on web application reload	6 years ago



TCD Command Line Interface



```
$ tcd --help
```

```
Usage: tcd [OPTIONS] COMMAND [ARGS]...
```

Tackle Configuration Discovery.

Options:

```
-v, --verbose    Show verbose output
--help           Show this message and exit.
```

Commands:

```
collect    Collect configuration files for a given framework.
config     Displays the current configuration
init       Initialize a new configuration.
list       List the available frameworks.
translate  Translate configuration into target framework.
```



KONVEYOR TACKLE

TCD Init CLI



Initializes the common parameters for running discovery including setting the location of the windup server

```
$ tcd init -d ./out -w 127.0.0.1:8180
```

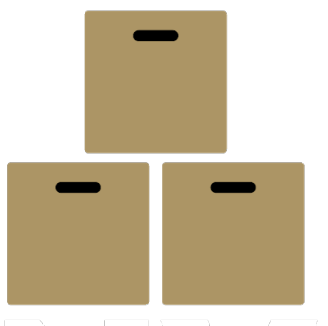
```
$ tcd init --help
```

```
Usage: tcd init [OPTIONS]
```

Initialize a new configuration.

Options:

<code>-d, --data-directory PATH</code>	Directory of configuration database
<code>-w, --windup TEXT</code>	tcd-windup server host
<code>-s, --windup-ssl</code>	Enable SSL for tcd-windup server
<code>--help</code>	Show this message and exit.



TCD Collect CLI



Collects all of the configuration files in the source folder for a given framework.

```
$ tcd collect --framework liberty --source ./src
```

```
$ tcd collect --help
```

```
Usage: tcd collect [OPTIONS]
```

Collect configuration files for a given framework.

Options:

<code>-f, --framework TEXT</code>	The application framework to look for [required]
<code>-c, --collector TEXT</code>	The collector to use [file windup]
<code>-s, --source TEXT</code>	Source folder to scan
<code>-r, --repo TEXT</code>	Public git repository to scan
<code>-a, --all</code>	Collect all filenames
<code>-o, --output [yaml json ndjson]</code>	Output format for collected data
<code>--help</code>	Show this message and exit.



TCD Windup Collector



Input

GitHub URL of Target Application Codes

TCD Frontend CLI
built by Python

Quarkus Template (Jina2)

Step 1. Collect application configurations

Step 2. Generate target framework configuration

2-1. Query canonical configurations

2-2. Transform to Quarkus configurations

Output

Quarkus Application Configuration

TCD Backend
running as Docker Container

REST API Handler

CLI

Web Socket

Canonical Configuration DB

Windup

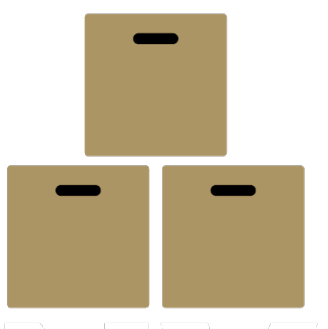
JanusGraph

1-1. Crawl configurations from codes

SpringBoot Config Crawler (Windup Rule)

JanusGraph Data Extractor (Windup Rule)

1-2. Transform/Store as canonical configurations



TCD Translate CLI



Translates configuration information found by a discovery run into a new target configuration

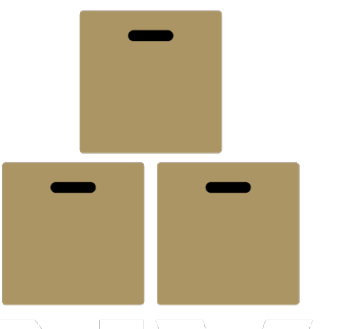
```
$ tcd translate --framework quarkus --target ./out
```

```
$ tcd translate --help
Usage: tcd translate [OPTIONS]
```

Translate configuration into target framework.

Options:

<code>-f, --framework TEXT</code>	The application framework to translate to [required]
<code>-t, --target TEXT</code>	Target folder to place config files
<code>--help</code>	Show this message and exit.



Join the Konveyor Community

- **Chat**

#konveyor on slack.k8s.io

Join the [Konveyor-community list](#)

- **Share**

Propose a meetup talk ([form](#))

- **Contribute**

Join the next quarterly project planning

Invites sent to [Konveyor-community list](#)



www.konveyor.io

Thank you!

www.konveyor.io

