

# AtlasMap

The AtlasMap Team

Version 1.31-SNAPSHOT, 2017-10-31

# AtlasMap

|                               |   |
|-------------------------------|---|
| 1. Introduction .....         | 2 |
| 2. Terminology .....          | 3 |
| 3. User Guide .....           | 5 |
| 3.1. Prerequisites .....      | 5 |
| 3.2. Maven Instructions ..... | 5 |
| 4. Release Notes .....        | 6 |
| 4.1. 2.0.0 Release .....      | 6 |



# Chapter 1. Introduction

This is the AtlasMap documentation. Please standby, more to come soon.

# Chapter 2. Terminology

## AtlasMap terminology

| Term         | Definition  | Example   |
|--------------|---|---|
| AtlasMapping | The top-level mapping file containing the mapping instructions to execute at runtime  | atlasmapping.xml, atlasmapping.json   |
| Audit        | An Audit is informational tracing data captured runtime execution in order to provide user feedback   | Useful during testing cycles, or when user does not have direct access to the runtime logs              |
| Collection   | A data type used in conversion that repeats 0 or more times   | Used to process Arrays and Lists  |
| Data Source  | The definition of an input or output data format within an AtlasMapping file  | atlas:java, atlas:json, atlas:xml, etc  |
| Field        | The base type used to describe a data value used in a Mapping   | JavaField, JsonField, XmlField, etc   |
| Field Action | A function to perform on a given value of an input or output field  | CurrentDate, Trim, SubString, etc   |
| Field Type   | Normalized convention to describe the type of data expected within the value of a field   | String, Integer, Long, Number, Date Time, etc   |
| Lookup Table | A cross reference table used at runtime to define a data conversion that varies based on the value of the input field   | Used for mapping enumerations   |
| Mapping      | A grouping of input Field(s), output Field(s) and optionally Field Action(s) that defines the conversion to be performed at runtime within an AtlasMapping file | JavaField → JsonField, XmlField → JsonField + Uppercase, etc  |
| Mapping Type | The specific type of conversion that should be performed at runtime   | Map, Combine, Separate, Collection or Lookup  |
| Validation   | A Validation is a syntax check of a provide mapping file to ensure execution may proceed  | Useful during testing cycles, and at runtime to avoid complicated and misleading exception stack traces |

## AtlasMap Data Format terminology

| Term                | Definition  | Example                     |
|---------------------|---|-----------------------------|
| Boxed Primitive     | A type of primitive that have a 'null' value  | Integer, Long, Boolean, etc |
| (Unboxed) Primitive | A type of primitive that cannot be 'null' and generally has a default initialized value | int, long, boolean, etc     |

| Term        | Definition  | Example   |
|-------------|---|---|
| Rooted      | JSON documents that have a parent node identifying to contents of the JSON document         | { "Contact": { "firstName": "Yu", "lastName": "Darvish" } } |
| Unrooted    | JSON documents that do not have a parent node identifying the contents of the JSON document | { "firstName": "Yu", "lastName": "Darvish" }                |
| Qualified   | Xml elements and/or attributes that contain the namespace prefix                            | <ns1:Contact ns1:firstName="Yu" ns1:lastName="Darvish" />   |
| Unqualified | Xml elements and/or attributes that do not contain the namespace prefix                     | <Contact firstName="Yu" lastName="Darvish" />               |

# Chapter 3. User Guide

The User Guide documents how programmers will use AtlasMap within their code projects.

## 3.1. Prerequisites

- Required: Java 1.8.0u91
- Recommended: Java 1.8.0u131 (especially for running within a container)

## 3.2. Maven Instructions

Add the AtlasMap core runtime to the pom.xml

```
<dependency>
  <groupId>io.atlasmap</groupId>
  <artifactId>atlas-core</artifactId>
  <version>{version}</version>
</dependency>
```

# Chapter 4. Release Notes

## 4.1. 2.0.0 Release

- All new modular runtime
- Supported data formats: Java, Json and Xml
- Support for auto-type detection
- Support for auto-type conversion
- Support for Field Actions
- Support for design-time and runtime mapping file Validation rules
- Support for tracking runtime Audits useful for design-time feedback
- Design Time: Support service
- Design Time: Java class inspection service
- Design Time: Xml Schema and Xml instance inspection service
- Design Time: Json instance inspection service