

Eclipse Tool Stack Interchange Format

DRAFT - Version 0.1

Abstract

This document defines a standard interchange format for Eclipse-based development environments. The goal of this specification is to provide a transparent description of development environments for their transfer, archival, and instantiation. This document describes version 0.1 of this specification.

Statement of Draft Status

This document is a draft and is incomplete. It is the intention of the authors that it be released in this state to foster discussion about its applicability before further work is initiated on the specification. Incomplete sections of this document are denoted via "TODO:" text.

Copyright

Copyright (C) Genuitec, LLC (2013). All Rights Reserved.

Outline

TODO: Generate Outline

Introduction

Purpose

The goal of this document is to create an standard interchange format for describing Eclipse-based development environments. This document outlines the definitions of what software is considered part of the tool stack, the requirements of instantiating a tool stack, and how the format can be extended to support different flavors of tooling and configuration.

The format aims to allow easy transport of tool stacks between environments. Since tool stacks are complicated collections of software, it is a goal that humans to be able to author, inspect, browse, and verify interchange files. Additionally, the format must be machine readable so that reconstitution of the tool stack in a new environment can be performed. Finally, the specification aims to make possible the archival of tools, configuration, and source code association so that tool stacks can be archived in any location.

Requirements

For purposes of this specification, the following words have significance in terms of defining the compliance of implementations.

MUST, REQUIRED

Items designated by "must" or "required" are considered critical to the core functionality defined by this specification. Implementations which fail to comply to an item designated by "must" or "required" are considered non-compliant with this specification.

SHOULD, RECOMMENDED

Items designated by "should" or "recommended" are considered part of the specification, but there may be situations when not supporting the item makes sense.

MAY, OPTIONAL

Items designated with "may" or "optional" may be included by an implementation or it may be omitted.

Terminology

- TODO: Define common terms

Not Covered by This Document

- TODO: Define what is not included

Interchange Format

Overview

This specification at its simplest defines an [XML file](#) which utilizes the Eclipse Tool Stack Interchange Format (ETSIF) schemas to define a tool stack. However, these schemas reference resources and so for ease of transport any implementation of the specification must handle either an XML file or a [ZIP](#) file containing at the root level one or more XML files.

TODO: Further discussion over whether a zip can contain only one tool stack or many

The URIs used to reference resources in the ETSIF schemas can be absolute or relative. When they are relative they must refer to either files and folders on the file system in the case of a plain XML, or virtual files and folders contained in the ZIP file in the case of a ZIP.

The XML schemas split the specification into different namespaces. These namespaces provide an extensible model so that the specification can be extended publicly or privately to support further tools. This document includes the specification for three namespaces namely, the Core ETSIF namespace (TODO: finalize location <http://.../xml/ns/etsif>), the ETSIF Eclipse namespace (TODO: finalize location <http://.../xml/ns/etsif/eclipse>), and the ETSIF Java namespace (TODO: finalize location <http://.../xml/ns/etsif/java>). Utilizing these three namespaces and their corresponding schemas, an Eclipse tool stack can be defined,

transported, and instantiated.

The ETSIF Namespace

The ETSIF Namespace and its schema (<http://.../xml/ns/etsif.xsd>) define common elements required to define a collection of tools. These elements are considered rudimentary enough to be shared across all tools and even used within elements from other ETSIF namespaces. These elements must be understood by implementations of this specification.

TODO: More discussion of the core elements

Core Elements

- *tool-stack* - the root element, defines the unique identifier and version for the tool stack
- *identifiers* - a grouping element for *identifier* elements
- *identifier* - defines one of the aliases this tool stack represents
- *tool* - defines a tool that is included in this tool stack
- *uses* - a grouping element for *used-tool* elements
- *used-tool* - indicates a tool that is used by another tool
- *applicability* - a group element for *platform* elements
- *platform* - defines one or more platforms on which this tool can operate by specifying optionally an operating system, architecture, and/or windowing system
- *branding* - collects common branding elements together
- *description* - provides a human readable description as part of the branding
- *icons* - collects a group of icons together
- *icon* - defines an icon as part of the branding

ETSIF Eclipse Namespace

The Eclipse ETSIF Extension Namespace and its schema (<http://.../xml/ns/etsif/eclipse.xsd>) define elements required to define an Eclipse instance. These elements must be understood by implementations of this specification.

At the core of this extension is the *contents* element and its sub-elements *extends*, *include*, and *include-all*. These three elements define which bundles are in the Eclipse installation.

The *extends* element allows for the target Eclipse to be based off of a known Eclipse be it an EPP defined package or another ETSIF Eclipse tool.

The *include* element allows for the grouping of repositories with required installable units or more generic P2 requirements. The tooling should only include the installable units that satisfy the *iu* and *requires* elements and their dependencies in as a minimal spanning solution to the dependency graph.

The *include-all* element allows for the grouping of repositories which should be included in the Eclipse instance completely. This allows for common use cases like including plugins from

target platform locations or dropins folders.

TODO: Discuss the lesser elements in the Eclipse ETSIF namespace

ETSIF Eclipse Elements

- *eclipse* - defines an Eclipse instance including extra bundles and workspace configuration
- *contents* - designates which bundles make up the Eclipse installation
- *extends* - indicates that this tool stack extends the referenced Eclipse
- *include* - indicates that the specified bundles and their requirements should be included in the Eclipse instance from the given sources
- *include-all* - indicates that all of the bundles located in the given location should be included in the Eclipse instance from the given sources
- *software* - designates a self-contained software item that is to be included in the Eclipse instance and which may be branded independently from the Eclipse
- *repository* - locates a P2 repository either within the archive zip or via URL from which bundles or metadata may be acquired
- *iu* - designates a P2 installable unit that must be included (with its dependencies) from a collection of repositories
- *requires* - designates a P2 requirement which must be fulfilled (with its dependencies) from a collection of repositories
- *install-configuration* - provides install configuration settings
- *install-folder* - indicates the folder name into which the tool stack should be installed
- *ini-settings* - groups settings that need to be put into the ini file to configure the launching of the Eclipse instance
- *java-virtual-machine* - indicates the Java virtual machine that should be used to launch the Eclipse instance
- *program-arguments* - groups the program *argument* elements
- *virtual-machine-arguments* - groups the virtual machine *argument* elements
- *argument* - defines a single argument
- *config-properties* - groups the *property* elements that should end up in the config.ini file
- *property* - defines a Java .properties file property
- *workspace-configuration* - groups and orders the workspace configuration elements
- *project-set* - indicates an Eclipse project set file (.psf) that should be imported into the workspace
- *preferences* - indicates an Eclipse preferences file (.epf) that should be applied to the workspace

ETSIF Java Namespace

The ETSIF Java namespace and its schema (<http://.../xml/ns/etsif/java.xsd>) define elements which define a Java Virtual Machine instance. These elements must be understood by implementations of this specification.

TODO: Discussion of the elements in this namespace

ETSIF Java Elements

- *java-virtual-machine* - defines a java virtual machine, either a JDK or JRE
- *archive* - defines an archive containing the JVM

Extensions

This specification can be extended both publicly and privately by the definition of new namespaces and schemas. New namespaces should define elements which appear as children of *tool* elements or within other namespace elements to further customize their behavior.

TODO: Further discussion on extensions and possible example

Full Copyright

Copyright (C) Genuitec, LLC (2013). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to Genuitec, LLC or other organizations, except as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by Genuitec, LLC or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and GENUITEC, LLC DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Appendix A: Example Tool Stack XML File

TODO: Include example file, not considered part of the specification