Apache Ant Tasks for IBM® InfoSphere® Streams

Version 1.1.2
Overview

The com.ibm.ssb.ant.jar defines a number of Apache Ant tasks for IBM InfoSphere Streams

Supported Tasks

- `indextoolkit` – Index an SPL toolkit
- `maphostpooltags` – Map host pool tags in an application's ADL to different values
- `splcompile` – Compile an SPL application
- `spldoc` – Generic SPLDOC for a toolkit using spl-make-doc
- `sc` – an alias for splcompile
- `submit` – Submit compiled applications to a Streams instance

Using the Streams tasks within Apache Ant

It is assume that you are familiar with Apache Ant and have it installed. This tasks have been tested with Apache Ant 1.8.4.

To compile the code:

```
cd com.ibm.ssb.ant
ant
```

This generates the library com.ibm.ssb.ant.jar in the lib folder.

To use this in ant the jar must be copied into `$HOME/ant/lib/`. This can be achieved using ant:

```
cd com.ibm.ssb.ant
ant install
```

The Streams tasks are defined to ant as follows:

```
<taskdef resource="com/ibm/ssb/ant/streams.xml"/>
```

This assumes the com.ibm.ssb.ant.jar library has been copied into `$HOME/ant/lib/`.

The tasks use the ant property `streams.install` to define the location of the InfoSphere Streams install. This property may be set explicitly in a build file or from the environment variable $STREAMS_INSTALL like this:

```
<property environment="env"/>
<property name="streams.install"
       value="${env.STREAMS_INSTALL}"/>
```

Tasks that require an instance, default to the instance defined by the ant property `streams.instance` otherwise the instance can be defined as an attribute of the task.

Example build files can be found under `com.ibm.ssb.ant/examples`.

Task: indextoolkit

Description

Indexes an SPL toolkit. The task invokes the 'spl-make-toolkit' command. Specific details of spl-make-toolkit's behaviour can be found in the InfoSphere Streams documentation set.

Mixed mode processing is not supported, spl-make-toolkit is always executed with the –no-mixed-mode-preprocessing flag.

Parameters

Attribute	Description	Equivalent spl-make-toolkit option	Required
location	Provides the path to the toolkit root directory.	-i, --directory	Yes
clean	Clean the generated files, does not index the toolkit.	-c, --clean	No, defaults to false

Example

```
<indextoolkit location="${vwap.toolkit}"/>
```

Task: splcompile

Description

Compiles an SPL application. The alias sc may also be used. The task invokes the SPL compiler 'sc'. Specific details of the SPL compiler behavior can be found in the InfoSphere Streams documentation set.

When a parameter is not required, the default corresponds to the default selected by the SPL compiler.

Parameters

Attribute	Description	Equivalent SPL Compiler option	Required
mainComposite	Main composite to use as the root of the expansion. This should be a fully qualified name.	-M, --main-composite	Yes
clean	Clean compiler generated	-c, --clean	No

Attribute	Description	Equivalent SPL Compiler option	Required
	artifacts prior to compilation.		
static	Generate statically linked PEs.	-s, --static-link	No
standalone	Create a standalone application executable. The executable will be named standalone and can be located under the bin directory within the output directory.	-T, --standalone-application	No
optimize	Generate optimized code with less runtime error checking.	-a, --optimized-code-generation	No
verbose	Put the compiler into verbose mode.	-v, --verbose-mode	No
noToolkitIndexing	Do not perform toolkit indexing in the compilation directory.	--no-toolkit-indexing	No
outputDirectory	Set the directory where the application artifacts are placed to the specified path.	--output-directory	No
dataDirectory	Set the location of the data directory to be used to the specified path.	--data-directory	No
checkpointDirectory	Set the directory to be used for checkpointing to the specified path.	-K, --checkpoint-directory	No
transport	Set the default transport protocol to one of tcp, llm_rum_tcp, or llm_rum_ib.	-F, --use-transport	No
fusion	Specify the fusion mode. Available options are: • *manual* or *FDEF*: Default fusion (user-controlled) • *optimized* or *FOPT*: Optimized fusion	-p, --part-mode	No

Parameters specifies as nested elements

Element	Description	Equivalent SPL Compiler option	Required
toolkit	Add a toolkit location to the toolkit path.	-t, --spl-path	No

Example

```
<splcompile
       mainComposite="${mainComposite}"
       standalone="true"
       optimize="true"
       noToolkitIndexing="true"
       dataDirectory="${vwap.toolkit}/data">
   <toolkit location="${vwap.toolkit}" />
 </splcompile>
```

Task: submit

Description

Submits compiled SPL applications to a running Streams instance. The task invokes the 'streamtool submitjob' command. Specific details of job submission can be found in the InfoSphere Streams documentation set.

Parameters

Attribute	Description	Equivalent streamtool submitjob option	Required
instance	InfoSphere Streams Instance Id. If not specified, default to using the value of the property *streams.instance*. If that is not defined streamtool submit job defauls to the value of the STREAMS_DEFAULT_IID	-i,--instance-id	No

Attribute	Description	Equivalent streamtool submitjob option	Required
	environment variable.		
outfile	Name of the output file to write the list of submitted job IDs. The file can be used to cancel the jobs using the the "--file" option with the "streamtool canceljob" command. The file will be overwritten if it already exists.	--outfile	No, when not provided no file is written

Parameters specifies as nested elements

Element	Description	Required
applications	Defines a collection of applications to be submitted. The applications element supports any number of nested: • *fileset* elements to select adl files to submit. • *parameter* elements to specify job submission time parameters (-P option to submitjob) • *config* elements to specify an application configuration setting The parameter and config settings apply to all jobs identified by the fileset elements.	Yes,one and only one must be supplied.

Example

```
<submit outfile="jobids.out">
<applications>
    <fileset dir="output" includes="*.adl"/>
    <config name="tracing" value="debug"/>
  </applications>
</submit>/>
```

Task: maphostpooltags

Description

Creates a copy of an application's ADL file, mapping host pool tags to new values. This can allow an existing application to be adapted to an different host tagging scheme without modifying the SPL source and recompilation.

Parameters

Attribute	Description	Required
source	Source ADL file	Yes
target	Target ADL file. The target ADL file will be a copy of the source ADL with the host pool tags modified according to the nested tag elements. It is recommended that the target file is in the same directory as the source.	Yes

Parameters specifies as nested elements

Element	Description	Required
tag	Supports two mandatory attributes, *old* and *new*. The task will map any tag with the *old* value to the corresponding *new* value in the target ADL.	No

Example

Example where the application uses the tags *source* and *sink* for host pools, but a specific Streams instance has a different tagging scheme, using *ingest* and *egress* for hosts that provide and consume streaming data respectively.

```
<maphostpooltags
    source="${basedir}/output/SampleTag.adl"
    target="${basedir}/output/SampleTagIngest.adl">
  <tag old="source" new="ingest"/>
  <tag old="sink" new="egress"/>
</maphostpooltags>
```

Task: spldoc

Description

Generate SPLDOC for SPL toolkits. The task invokes the 'spl-make-doc' command. Specific details of spl-make-doc's behaviour can be found in the InfoSphere Streams documentation set.

Mixed mode processing is not supported

Parameters

Attribute	Description	Equivalent spl-make-doc option	Required
location	Provides the path to the toolkit root directory.	-i, --directory	Yes, unless toolkit nested elements are provided.
clean	Generate clean .pm files and a toolkit.xml file.	-c, --clean	No, defaults to false
author	Include the name of the toolkit author in the documentation	--author	No, defaults to no author
title	Add a title to the documentation	--title	No
sourceCode	Include SPL source code in the doc	--include-source	No, defaults to not including SPL source
compositeGraphs	Include diagrams for composite graphs in the doc	--include-composite-operator-diagram	No, defaults to not Including composite graphs
outputDirectory	Output directory for the documentation	--output-directory	No

Parameters specifies as nested elements

Element	Description	Required
`toolkit`	Specifies a toolkit to create documentation for, using the `location` attribute. Multiple toolkit elements are supported, and a single SPLDOC set is created for all toolkits. The single view is made by copying all	No

Element	Description	Required
	the toolkits into a single toolkit in a temporary directory and then running SPLDOC against the merged toolkit. If the same SPL artifact exists in multiple toolkits then the behaviour is undefined, thus this is intended to document non-overlapping toolkits. Specifically this will not work with multiple versions of the same toolkit. The merged toolkit description in SPLDOC is a concatenation of the all the toolkit descriptions, each preceded by the toolkit name in bold.	

Example

```
<spldoc location="${vwap.toolkit}" compositeGraphs="yes"
     title="VWAP Example"/>
```

For multiple toolkit support see the supplied example in:

```
examples/MultiToolkit
```