

Automated Build Generator - Feature #1684  
Implement "includes" relation for distributions

12/11/2013 12:03 AM - J. Moringen

Status:	Resolved	Start date:	12/11/2013
Priority:	Normal	Due date:	
Assignee:	J. Moringen	% Done:	100%
Category:	Data Model	Estimated time:	0.00 hour
Target version:	1.0		
Description			
<a href="#">Ingmar's report</a>			
<div><div></div><div><i>Distributionen, wie z.B. die RSB-0.9 Distribution, sollen in anderen Distributionen als Bestandteil angegeben werden können. Ähnlich wie bereits Projekte.</i></div></div>			
<a href="#">Duplicate</a>			
<div><div></div><div><i>Is there a way to define a "sub-distribution" that can be included in your distribution? E.g. the flobi simulation needs a set of ~8 projects. Everytime we define a distribution that uses that flobi sim we have to copy and paste these 8 projects. When we add a dependency we have to do that on all packages... It would be really nice to have such a feature.</i></div></div>			

Associated revisions

Revision 832e17a7 - 01/10/2019 05:27 PM - J. Moringen

Model changes for distribution includes in src/model/project/classes-spec.lisp

- New concepts {[resolved-]project,distribution}-include represent inclusion of distributions and projects with parameters.
- A new `direct-includes' slot stores the distributions included by the a given one.
- The `version' slot is now called `direct-versions' since project version can now also be included indirectly through an included distribution.

refs #1684

- src/commands/functions-input.lisp (header): updated copyright  
(locate-projects): adapted to model changes  
(load-project/versioned): likewise  
(load-projects/versioned): likewise  
(resolve-project-versions): likewise
- src/commands/command-generate.lisp (header): updated copyright  
(generate-analyze): adapted to model changes
- src/commands/package.lisp (header): updated copyright  
(package jenkins.project.commands): added shadowed symbol version

- src/model/project/classes-model.lisp (header): updated copyright  
(include-context): new class; include context for project versions  
(version::context): new slot; stores include context  
(lookup version t): look in specification, include context, parent, context distribution and toplevel distribution
- src/model/project/classes-spec.lisp (header): updated copyright  
(distribution-include): new class; represents an included distribution  
(print-items append distribution-include): new method; print items  
(project-include): new class; represents an unresolved project version include  
(print-items append project-include): new method; print items  
(resolved-project-include): new class; represents a resolved project version include  
(print-items append resolved-project-include): new method; print items  
(distribution-spec::direct-includes): new slots; stores included distributions  
(distribution-spec::direct-versions): renamed versions -> direct-versions  
(versions distribution-spec): new method; collect and return direct and transitively contained versions  
(instantiate distribution-spec): instantiate transitively contained versions with appropriate include contexts  
(instantiate version-spec): accept include context
- src/model/project/package.lisp (header): updated copyright  
(package jenkins.model.project): added exported symbols  
direct-{includes,versions}, versions distribution-include, distribution, project-include, project, version, resolved-project-include and context

#### Revision d13cf6fb - 01/10/2019 05:31 PM - J. Moringen

Syntax for distribution includes in src/model/project/concrete-syntax/yaml.lisp

refs #1684

- src/model/project/concrete-syntax/yaml.lisp (parse-distribution-include-spec):  
new function; parse the different forms of distribution include specification  
(define-yaml-load one-distribution): renamed distribution -> one-distribution; accept include key in distribution recipes  
**(distributions)**: new variable; already loaded distributions  
(find-distribution): new function; find a loaded distribution  
(ensure-distribution): new function; find a distribution or call a thunk  
**(distribution-load-stack)**: new variable; stack of distributions currently being loaded  
(find-or-load-distribution): new function; find or load a distribution  
(resolve-distribution-dependency): new function; load a distribution from a file if necessary

## Revision 19f5f473 - 01/10/2019 05:31 PM - J. Moringen

Avoid inheritance of distribution-name in src/model/project/classes-model.lisp

Projects will only ever "see" the value of the distribution-name variable in the "outermost" distribution, i.e. the one specified on the commandline.

refs #1684

- src/model/variables/schema.lisp (header): updated copyright (inheritance-mode): added value :outermost-only
- src/model/schema.lisp (header): updated copyright (define-variable :distribution-name): new variable; name of the distribution with :outermost-only inheritance
- src/model/project/classes-model.lisp (lookup version t): don't inherit variables specified as :outermost-only

## Revision e536cafe - 01/10/2019 05:35 PM - J. Moringen

Added enhancement entry in changes.sexp

fixes #1684

- changes.sexp (release 0.26): added enhancement entry

## History

---

### #1 - 05/20/2014 09:35 PM - J. Moringen

- Target version changed from 0.3 to 0.4

### #2 - 10/31/2014 05:14 PM - J. Moringen

- Description updated

### #3 - 03/02/2015 02:07 PM - J. Moringen

- Target version changed from 0.4 to 0.5

### #4 - 03/05/2015 09:59 AM - S. Meyer zu Borgsen

Could it be a first simple solution to simply join multiple "versions"-lists?

### #5 - 03/05/2015 10:36 AM - J. Moringen

S. Meyer zu Borgsen wrote:

| Could it be a first simple solution to simply join multiple "versions"-lists?

This really depends on the mid- and maybe long-term strategy for this project. There are three major possibilities:

1. Implement the feature in the current architecture and code base (as you suggest). I did this in a feature branch but it stretches the current architecture far beyond its original scope.
2. Implement the feature in the new architecture loosely based on the current code base. I did this in a feature branch as well and it worked quite well. The new architecture solves several other problems as well. For example, distribution version are defined in a single recipe and share most of the definitions like project versions currently already do. As another example, branches and tags are no longer mandatory in projects; instead, distributions can request arbitrary branches, tags or commit without changes to project recipes.
3. Rewrite the project according to the "new architecture" mentioned in 2) or some other architecture. This would of course allow going about the inclusion feature in the same way as in 2) or in some other smart way.

The complications, potential problems and short-term character involved in 1. probably make it not worth the effort.

We attempted to decide between 2. and 3. several times but each time failed to organize the respective meeting. No matter how we decide between 2. and 3., significant effort will be required and the respectively resulting new generator will not be a drop-in replacement, thus another migration will be necessary in addition.

As a result, I think this feature will take some time.

#### #6 - 03/05/2015 02:04 PM - S. Wrede

J. Moringen wrote:

*We attempted to decide between 2. and 3. several times but each time failed to organize the respective meeting.*

Which was also due to holiday of key personnel involved... ;-)

*No matter how we decide between 2. and 3., significant effort will be required and the respectively resulting new generator will not be a drop-in replacement, thus another migration will be necessary in addition.*

This is true. However, this feature may be ranked relatively high on the roadmap for the new build gen architecture / implementation strategy as the lack of it currently impedes scalability of the development process.

#### #7 - 09/30/2016 10:43 AM - J. Moringen

- Target version changed from 0.5 to 0.8

#### #8 - 10/25/2016 12:32 PM - J. Moringen

- Target version changed from 0.8 to 0.9

#### #9 - 02/23/2017 11:35 AM - J. Moringen

- Target version changed from 0.9 to 1.0

#### #10 - 01/10/2019 06:27 PM - J. Moringen

- Status changed from New to Resolved

- % Done changed from 0 to 100

Applied in changeset commit:e536cafea8d4f5a47daf12ccfb905ec363c75a79.