Table of Contents

Build & Test Web Application

Improvements

Dependencies

(as from SA1 discussion held on 14th Nov 2008)

- **Dependencies view** - listed like children nodes of the "Dependencies" node in the workspace
    - The behaviour of the 'dependencies' node does not change: by clicking on it, the list is still shown in the right panel
    - Children dependencies nodes appears are modules (and not as configurations)
    - Children dependencies nodes can be further expanded (if projects or subsystems)
    - Children dependencies nodes should allow both editing of the dependency relation and editing of the module (if allowed)
    - Default action when clicking on children dependencies: two options:
        - Show module details
        - Show dependency relationship details (e.g. type of dependency, contraints, ...)

- **Dependency editing** - the current approach (click on trees/lists, then add) should be replaced by the following
    - focus on dependencies and switch to edit mode
    - click 'add', in the same way as properties/environment are added
    - a new entry in the dependency list is added. The entry is much like the current one, with the exception of the module name, which is not defined and replaced with two GUI elements:
        - a 'suggest field' where the user can enter a substring of the module to be added; by exploiting a new WS operation 'search', a list of modules matching the query are shown just below for selection. This is faster when you know (part of) the name of the dependency.
        - a button showing a popup window containing the project tree. Here the uses can select the correct module. This is helpful when the name is not correctly known.
    - Still in the current widget for dependencies, when the configuration is static and when in editing mode, the name of the configuration is replaced by a drop-down list with all configurations for the given module. This allows to update a static dependency without removing and adding it again.

- **Dependency Resolution** - it will be possible to preview how dependencies will be resolved.
    - How it's done:
        - static: trivial, done on the WA
        - dynamic (either 'default' or contraint-based): exploit a new operation on the WS. In some cases it's required to specify a parent configuration to be used as context for the resolution.
    - How the resolution is triggered:
        - automatically depending on the tree in the workspace (the context should be clear to the users). Also depends on how heavy is the operation on the WS.
        - manually on the right panel (both view and edit mode) and on the nodes in the workspace. This might require the request of a resolution context (i.e. project-config)

Properties

(as from SA1 discussion held on 14th Nov 2008)

- **Properties View/Editor** - should be split in three (maybe four in the future) sections:

- Dependency resolution Properties
- System Properties
- User Properties
- (Prereq Properties - still under discussion)

- **Dependency Resolution Properties** - Edit Enhancements
    - The **key** can be set in two ways:
        - via a suggest field, populated with all existing modules
        - via a popup dialog showing the structure of all projects in the system
    - The **value** is set though a drop-down list containing all configurations for the selected module
    - The '.DEFAULT' suffix in the key is automatically appended when saving and stripped out when the property is displayed.

- **System Properties** - Edit Enhacements
    - The **key** is set with the help of two drop-down lists:
        - First the user can select a 'category'
        - Then, within a category, a number of properties are available
    - If available, a short description is presented to the user
    - The **value** is free-text, unless the selected property is defined in a restricted domain. In the latter case, a drop-down is exploited
    - The WA is made aware of available system properties through a configuration file (xml) local to the WA installation.
        - The configuration file organizes system properties in 'categories'; provides allowed values when applicable; optionally provides a description to be presented to the user.

- **User Properties**
    - A check is needed to avoid Dep o System properties to be inserted here.

- **Prereq Properties (Under discussion)** - Properties to specify the software required to be installed on the host in order to run the build/test. Not every software in the 'external' project can be specified in this section, but only a little sub-set of them (e.g. gcc, python)
    - **Keys** should be of the form '<comp_name>.PREREQ'
    - The **value** is the configuration name
    - The view/edit approach should be the same as for Dependency Resolution Properties
    - In order to know what software can be pre-installed with this mechanism, the WA is provided with a configuration file local to the WA installation.

External Requirements

New methods on the WS:

- resolveDependency(...) - to be detailed

- **Module[] searchModules(key, limit)**
    - **parameters**:
        - key: a pattern to match in the module name
        - limit: the maximum number of configurations in the result set. To reduce the response size
    - **Matching**: should not be exact wrt special characters as [-_.,] They all should match each other. Example: etics_R_1_0_0 should also match etics_R_1.0.0 and etics-R-1-0-0
    - **Ranking**: the match where the key is closer to the most specific part of the name (i.e. the right side) should be preferred. Example: when looking for 'etics', the project 'org.etics' is preferred to 'org.etics.portal'
    - **Usage in the BTWA**: in the definition of 'Dependency Resolution Properties'when looking

for module names
    - **Note**: the method name is 'searchModules' on purpose, since it will be probably needed in the future a 'searchConfigurations' one.

- **Configuration[] listModuleConfigurations(moduleName)**
    - **parameters**:
        - moduleName: the name of the module to search configurations for
    - **Matching**: note that module names are not unique within Etics, thus the result set should contain configurations for all modules named 'moduleName'
    - **Ranking**: no particular ranking expected
    - **Usage in the BTWA**: in the definition of 'Dependency Resolution Properties'when looking for module configurations
    - **Note**: this method is not strictly needed. Its functionality can be obtained by combining an exact search plus the listXxxConfigurations

-- PaoloFabriani - 13 Jan 2009

This topic: ETICS > SA1WebApplication
Topic revision: r5 - 2009-01-14 - PaoloFabriani