

Automated Build Generator - Bug #1880

Behavior of @ {...|[] } seems wrong

05/12/2014 05:44 PM - J. Moringen

Status:	Closed	Start date:	05/12/2014
Priority:	Normal	Due date:	
Assignee:	J. Moringen	% Done:	0%
Category:	Description Language	Estimated time:	0.00 hour
Target version:	0.4		
Description			
Check with this recipe by removing the patches: [] line:			
<pre>{ "name": "xqilla", "templates": ["cor-lab", "autotools"], "catalogue.component.title": "xqilla", "catalogue.component.nid": "0", "variables": { "description": "XQilla: XPath + XQuery library", "keywords": ["xml", "xquery", "xpath"], "branches": ["trunk", "2.3.0", "2.2.0"], "upstream-version": "2.3.0", "xerces.version": ["trunk"], "patches": [], "configure.args": "--with-xerces=\"\${toolkit.dir}\"", "shell.command": " wget http://downloads.sourceforge.net/project/xqilla/xqilla/\${upstream-version}/XQilla-\${upstream-version}.tar.gz tar xzvf XQilla-\${upstream-version}.tar.gz @{shell.command.patch []} cd XQilla-\${upstream-version} \${next-value} ", "shell.command.patch": "patch -l -p0 < \${patches []}", "redmine.instance": "https://projects.cit-ec.uni-bielefeld.de", "redmine-project": "xcf", "extra-requires": [["freestyle", "xercesc", "\${xerces.version}"]], "extra-provides": [["freestyle", "xqilla", "\${upstream-version}"]] } }</pre>			

```

    ]

},

"versions": [
  {
    "name": "2.2.0",
    "variables": {
      "upstream-version": "${version-name}",
      "xerces.version": [ "2.8.0" ],
      "patches": ["~rhaschke/src/GAR/contrib/xqilla/files/gar-base.diff"]
    }
  }
]
}

```

History

#1 - 10/09/2014 03:35 PM - J. Moringen

- Target version changed from 0.3 to 0.4

#2 - 02/03/2015 06:50 PM - J. Moringen

- Status changed from New to Closed

The problem is `${patches[]}` which defaults to the string `[]` instead of an empty list. Using `${patches}` instead, fixes the problem.

#3 - 02/03/2015 09:35 PM - R. Haschke

Wouldn't `${patches}` expand to the empty string, and thus `shell.command.patch` to `"patch -l -p0 < "`, which would be wrong too?

#4 - 02/04/2015 12:15 PM - J. Moringen

R. Haschke wrote:

Wouldn't `${patches}` expand to the empty string, and thus `shell.command.patch` to `"patch -l -p0 < "`, which would be wrong too?

No, the empty string is handled specially in this context. I tested the suggested modifications for the "2.3.0" and "2.2.0" versions of the project.

#5 - 02/04/2015 12:27 PM - R. Haschke

OK. How can you then use something like

"value of variable is: `${myvar}`"

In this case, `${myvar}` should expand to the empty string if not defined, thus leaving "value of variable is: " as the final output. How do you specify the difference between empty set and empty string?

#6 - 02/04/2015 12:42 PM - J. Moringen

I'm inclined to say "just try what happens and extend the README file, if you have the time and motivation" as this is the legacy version of the build-generator and very little effort went into the variable substitution code in that version. Discussing and documenting this code extensively seems like wasted effort as it has been/will be redesigned and rewritten no matter how the upcoming roadmap discussion goes.