Document History

|  |  |
| --- | --- |
| Nov. 2, 2009 | Document Start |
| Dec. 28, 2009 | Updated markup after adding COM server |
|  |  |

Summary

* Document outlines the project file for RED’s build system, specifically; all xml markup.
* All nodes and attributes listed here have special meaning to the engine. They are reserved for engine use.

Contents

[<\* /> 3](#_Toc250012244)

[<BuildIt /> 3](#_Toc250012245)

[<BuildIt/Properties/Property /> 4](#_Toc250012246)

[<BuildIt/Sequences /> 5](#_Toc250012247)

[<BuildIt/Sequences/Action /> 6](#_Toc250012248)

[Special Actions 7](#_Toc250012249)

[<BuildIt/Sequences/Action type=”Msg” /> 7](#_Toc250012250)

[<BuildIt/Sequences/Action type=”SetProperty” /> 8](#_Toc250012251)

[Standard Actions 8](#_Toc250012252)

[<BuildIt/Actions /> 8](#_Toc250012253)

|  |  |  |  |
| --- | --- | --- | --- |
| <\* /> | | | |
| * attributes available to all nodes | | | |
| Required | Attribute | Values | Meaning |
| No | comment | user defined comment | * comment from user |
| No | help | RED defined comment | * comment about node for user to read * could be displayed in ‘help’ system for future |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| <BuildIt /> | | | |
| * root node for project file | | | |
| Required | Attribute | Values | Meaning |
| Yes | name | user defined name, (project or build name) | * name of project, used as an identifier * needs to be unique for each build on the computer, it is used to determine if the current project is already building * each computer can run many builds at the same time but, only 1 instance of a build project * \*if name is not defined BuildIt generates a name; Bld.GUID (where GUID is dynamically generated) |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| <BuildIt/Properties/Property /> | | | |
| * Our build system will use the concept of properties, instead of global variables * The concept of properties is the same as Windows Installer properties, except we will not have public or private ones. All properties will be treated the same * Properties can be static or set at runtime | | | |
| Required | Attribute | Values | Meaning |
| Yes | name | name of property | * Case-sensitive, no spaces |
| Yes | value | value of property | * Static properties ‘listed directly in the .xml’ are taken as literal strings * The Engine will define and resolve several standard properties, they can be identified by our prefix [r<PropertyName>] |

|  |  |  |  |
| --- | --- | --- | --- |
| <BuildIt/Sequences /> | | | |
| * section that holds the actions to be completed by this build * there can be unlimited <Action /> child nodes, nested actions are not supported * the section works similar to the Windows Installer sequence tables | | | |
| Required | Attribute | Values | Meaning  \* green text specifies default value, if not specified in markup |
| NO *(Key)*  reserved for future use | name | user defined name | * *Primary Key for <Sequences /> section* * not in use yet, reserved for future use (possible multiple sequences in project file) |

|  |  |  |  |
| --- | --- | --- | --- |
| <BuildIt/Sequences/Action /> | | | |
| * the actions to be completed by this build sequence * actions are completed from top down | | | |
| Required | Attribute | Values | Meaning  \* green text specifies default value, if not specified in markup |
| Yes | name | user defined name | * *foreign key to node <Action /> section* * *\*not required for 'special actions' listed below* |
| No | wait | 0 || 1 | * 0 means don’t wait, the action is executed on a separate thread * 1 means wait for the action to complete |
| No | condition | simple condition | * if FALSE, we action is skipped * if TRUE, we run the action |
| No | failaction | ignore || quit || retry | * ignore – action’s return code means nothing, script keeps going * quit – if action returns anything other than ERROR\_SUCCESS (0), then the build stops, return code and error message is written to the log file * retry – if action returns anything other than ERROR\_SUCCESS (0), then the build will retry the action 10 times, if the 11th try is unsuccessful the build stops |
|  |  |  |  |

Special Actions

|  |  |  |  |
| --- | --- | --- | --- |
| <BuildIt/Sequences/Action type=”Msg” /> | | | |
| * ‘Special action’, details of action can be listed directly in node, instead of a separate foreign key | | | |
| Required  \* resolvable from property | Attribute | Values | Meaning  \* green text specifies default |
| No | name | <omit name> | * not required, means nothing |
| Yes | value | message to display | * value can be messaged |
| No | display | 0 || 1 | * 0 means text is written to the log file * 1 means text is displayed on screen (message box)   + NOTE: message box is only displayed if command line parameter /showmsg is passed to the engine, otherwise we default to 0 |
| No | failaction | <always ignore> | * fail action means nothing to special actions |

|  |  |  |  |
| --- | --- | --- | --- |
| <BuildIt/Sequences/Action type=”SetProperty” /> | | | |
| * ‘Special action’, details of action can be listed directly in node, instead of a separate foreign key | | | |
| Required  \* resolvable from property | Attribute | Values | Meaning  \* green text specifies default |
| Yes | name | name of property to set | * name of property to set, if it does not exist a new one is created |
| Yes | value | value of property | * value |
|  |  |  |  |

Standard Actions

|  |  |  |  |
| --- | --- | --- | --- |
| <BuildIt/Actions /> | | | |
| * section that holds the actions the sequencer will call | | | |
| Required | Attribute | Values | Meaning  \* green text specifies default |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| <BuildIt/Actions/Action /> | | | |
| * Action to complete | | | |
| Required | Attribute | Values | Meaning  \* green text specifies default |
| Yes *(Primary Key)* | name | user defined name | - identifies action |
| yes | script | script to call | - needs to resolve to a full path (example: [rProductFolder]Scripts\Test.vbs)  - supported types  - .exe || .dll || .bat || .cmd || <any script with an interpreter>  - specify the interpreter in property [Interpreter.<extension>] – (example; Interpreter.vbs for vbscript, Interpreter.pl for perl |
| no | type | <set by code for standard actions> | - value will be <exe || dll || script> depending on the value set in the script attribute |
| no | cmdline | command line to pass to action | - applies only to .exe || .bat || .cmd |
| no  (yes if script is .dll) | entrypoint | function to call | - signature of function in .dll must be;  UINT \_\_stdcall MyFunction(\_\_in LPCTSTR szName);  - name of action is passed being executed is passed to function |
|  |  |  |  |
| \*NOTE: Interpreted scripts like perl or vbscript will be executed with 2 command line parameters  1 – <BuildIt's automation server id> Example; BuildIt.GUID value can be used with automation objects to find the build's server data  2 -- <Name of action> Example; MyAction, value can be used with the automation objects to get data from the server about the action  (See: .\Scripts\Tester.pl for example) | | | |

Standard Modules

* This section is our custom module section
* All nodes are currently being designed

|  |  |  |  |
| --- | --- | --- | --- |
| <BuildIt/Actions/Svn />  - script to call will be <NodeName>.pl  - Build engine will look at file extension to determine what kind of script (perl, vbs, js) | | | |
| * Get action, grabs some files from a source control | | | |
| Required | Attribute | Values | Meaning  \* green text specifies default |
| Yes *(Key)* | name | user defined name | * key from sequence section |
| Yes | type | svn || ftp |  |
| Yes | src | get from location |  |
| Yes | dest | get to location |  |
| No | options | *<depends on type>* | * command line options for getting files, value will be taken directly as parameters * ‘default’ = script defines what default is |
| No  *(Key to x`credits)* | credits | user defined name, key to <Credits /> node | * required if source control needs credentials |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
|  | | | |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| <BuildIt/Actions/File /> | | | |
| * Manipulates files | | | |
| Required | Attribute | Values | Meaning  \* green text specifies default |
| Yes *(Key)* | name | user defined name | * key from sequence section |
| Yes | type | copy || delete || move || rename |  |
| Yes | src | get from location |  |
| No | dest | get to location | * require for, copy, rename and move |
| No | options | *<depends on type>* | * command line options for copying files, value will be taken directly as parameters |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| <BuildIt/Actions/Run /> | | | |
| * Runs an external script or executable | | | |
| Required | Attribute | Values | Meaning  \* green text specifies default |
| Yes *(Key)* | name | user defined name | * key from sequence section |
| Yes | src | file to execute | * .vbs || .pl || .exe || .bat || .cmd * can use resolvable properties |
| No | cmdline | command line to pass to script | * can use resolvable properties |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| <BuildIt/Actions/Msg /> -- handled by sequence | | | |
| * Message box or write to log file | | | |
| Required | Attribute | Values | Meaning  \* green text specifies default |
| Yes *(Key)* | name | user defined name | * key from sequence section |
| Yes | text | <text to message> | * .vbs || .pl || .exe || .bat || .cmd |
| No | display | 0 || 1 | * 0 means text is written to the log file * 1 means text is displayed on screen (message box)   + NOTE: message box is only displayed if command line parameter /showmsg is passed to the engine, otherwise we default to 0 |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| <BuildIt/Actions/Email /> | | | |
| * Message box or write to log file | | | |
| Required | Attribute | Values | Meaning  \* green text specifies default |
| Yes *(Key)* | name | user defined name | * key from sequence section |
|  | to |  |  |
|  | cc |  |  |
|  | from |  |  |
|  | subject |  |  |
| Optional | body | default.htm | * if present, it can override default.htm – default.txt |
|  | plaintext | 1 || 0 | * 1 means use plain text, not listed or 0 is use htm |
|  | options | parameters  name=”value” name1=”value2” | * parsed by script, replaces values found in default.\* |
| No  *(Key to credits)* | credits | user defined name, key to <Credits /> node | * if you have it, send them email * if don’t have it, don’t error |
|  | savefile | specifies name of file | * save the file here (if specified) |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| <BuildIt/Data/Credits /> | | | |
| * Get action, grabs some files from a source control | | | |
| Required | Attribute | Values | Meaning  \* green text specifies default |
| Yes *(Key)* | name | user defined name | * key |
| Yes | user | user name |  |
| Yes | pass | password | * public version, password should be encrypted (not required for company startup) |
| Options | server | <Mail server and port> |  |

* LIST OF THINGS TO DO / ADD
  + add standard properties (like CommonAppData folder, WinDirFolder, ThisFolder ....)

Interpret

Actions:

Get type=”svn || ftp”

Put type=”svn || ftp”

File type=”copy || delete || rename || remove”

Test

Run type=”<defined by file>” (exe, vbs, pl, bat, cmd)

* digital signatures verification and signing will be done through a ‘Run’ action script

Config type=”” Set property is already handled

Notify type=”email || msg” - msg already handle