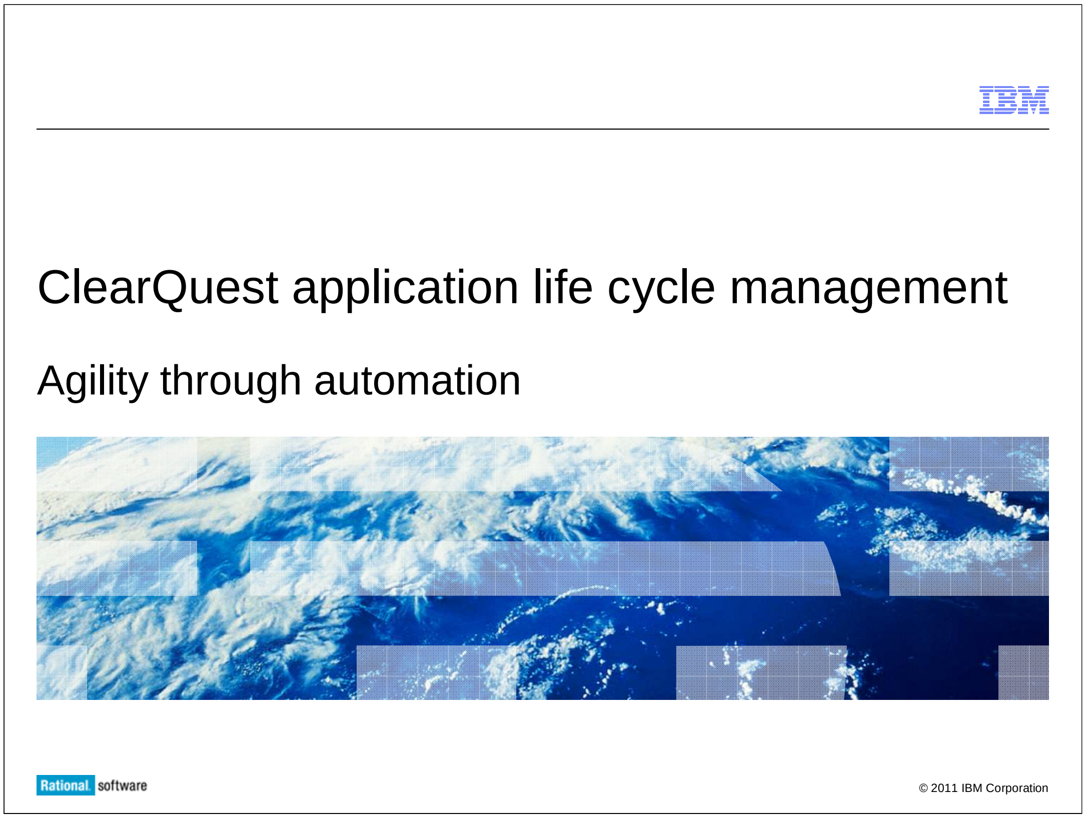


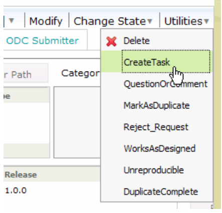
ClearQuest application life cycle management

Agility through automation

This is a presentation for the Agility Through Automation feature for IBM Rational ClearQuest Application Lifecycle Management (ALM).

What is “Agility through automation”?

- The 3-tier hierarchy of Request – Task – Activity provides projects with the flexibility to define their process workflow in a consistent way.
- In the original ClearQuest ALM package, manual steps are required to:
  - Run CreateTask utility to create tasks associated with a request.
  - Run CreateActivity utility to create primary activities associated with a task.
  - Complete task when their associated activities have completed.
  - Accept request when it is associated tasks has completed.
- Agility Through Automation is available starting in version 1.1 of the ALM package.
  It provides teams with the option to automate these and other workflows.
- Examples of provided automation:
  - Automatically create task following request submission.
  - Automatically create primary activities following task creation.
  - Automatically activate a test activity when its associated dev activity has completed.
  - Automatically complete task when its associated activities complete.
  - Automatically accept request when its associated tasks complete.



What is “Agility Through Automation”?

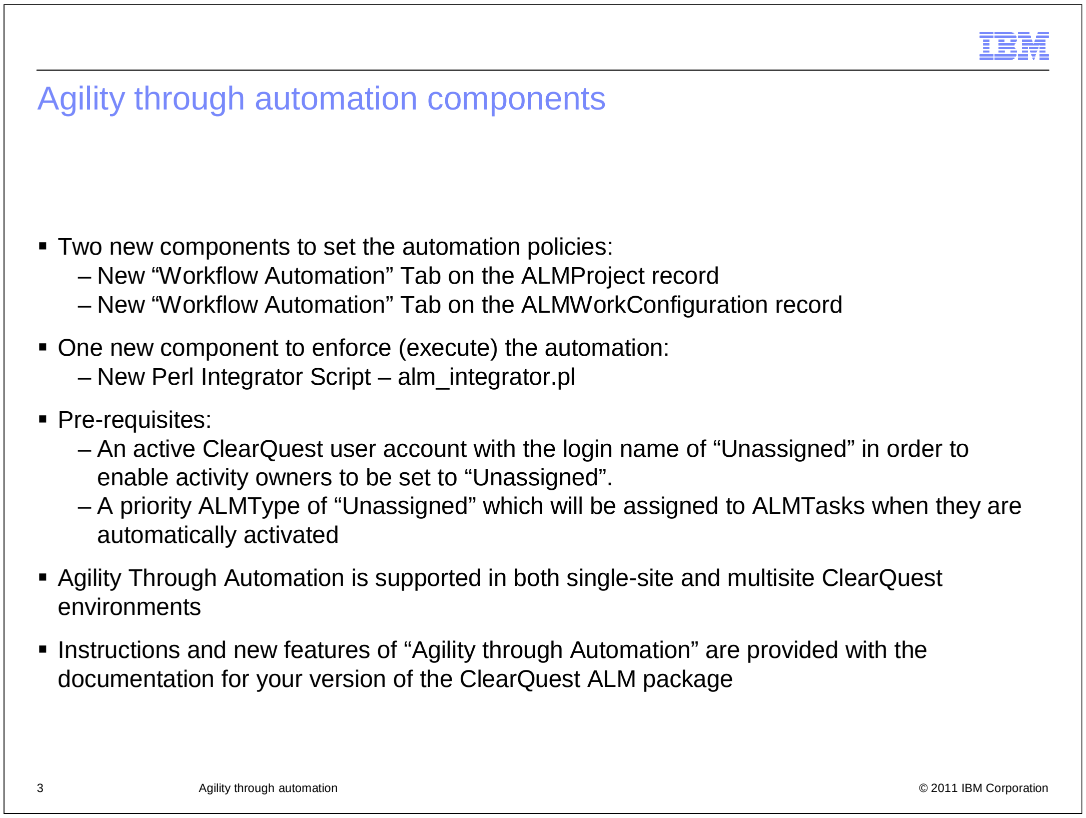
One of the of key features of ClearQuest ALM is its three-tier hierarchy of Requests, Tasks, and Activities. A typical example involves a request that is submitted. That request is triaged by someone on a project team, that identifies solutions required by that team. One or more tasks are created that are related to that solution. Those tasks are investigated by a development lead, and activities are identified to complete the tasks.

In the original ClearQuest ALM package, there are a number of manual steps required in these triaging activities. For example, the triage personnel must run the CreateTask utility to create tasks associated with the request. A development lead assigned to the task would need to run the CreateActivity utility to create the primary activities needed to deliver a solution for the task. Other items needing to be done include having to complete the task when completed. The request is than accepted if the solution provided in the associated task is acceptable. This process in general is adequate for some projects, although there is a number of manual steps. There are situations, particularly in Agile development, where there is a need to automate those steps.

“Agility Through Automation”, available starting in ALM package version 1.1, provides teams with the ability to automate these and other workflows. Effectively, you can automate the triage or development lead roles, allowing Agile teams to progress through the lifecycle of the 3-tier hierarchy with a degree of automation.

Some of the automation provided with this feature includes the ability to automatically create tasks following a request. For example, if someone has submitted a request, and automatic task creation is enabled, a task will be created and assigned to an applicable owner. You can also enable the ability to automatically create primary activates following task creation. Additionally, you can automatically action records within the same hierarchy when certain criteria are met. For example, test activities could be activated automatically upon the completion of an associated development activity.

In similar fashion, you can also the automatically complete a task when its associated activities are completed, or complete a request when its associated tasks are completed. This typically is desirable when the associated records have been completed with a successful result.



Agility through automation components

- Two new components to set the automation policies:
  – New “Workflow Automation” Tab on the ALMProject record
  – New “Workflow Automation” Tab on the ALMWorkConfiguration record
- One new component to enforce (execute) the automation:
  – New Perl Integrator Script – alm_integrator.pl
- Pre-requisites:
  – An active ClearQuest user account with the login name of “Unassigned” in order to enable activity owners to be set to “Unassigned”.
  – A priority ALMType of “Unassigned” which will be assigned to ALMTasks when they are automatically activated
- Agility Through Automation is supported in both single-site and multisite ClearQuest environments
- Instructions and new features of “Agility through Automation” are provided with the documentation for your version of the ClearQuest ALM package

3 Agility through automation © 2011 IBM Corporation

There are three main components for Agility Through Automation.

The first two of these are additional tabs on the ALM records. There is the Workflow Automation Tab on the ALMProject record. This is where you define items like automatically creating a task when a request is submitted, or automatically accept a request when a task is completed. The second Workflow Automation Tab is on the ALMWorkConfiguration record. There you can define predecessor to successor relationships. For example, you can define that once a development activity is completed, a test activity is activated.

The third of the main components is a Perl script called alm_integrator.pl, also known as the Perl Integrator Script. This script enforces the policies defined in the first two components, the Workflow Automation Tabs. The script is run at one location, or at one location at each site of ClearQuest Multisite environment. The script then provides the automation for the database of each site.

There are prerequisites for Agility Through Automation. The first is the existence of an active ClearQuest user account with the login name of “Unassigned”. This is required when an activity owner needs to be set to the value of “Unassigned”. The second prerequisite is having a priority ALMType record of “Unassigned”. This is required when ALMTasks are automatically activated, at which point they are assigned an “Unassigned” priority.

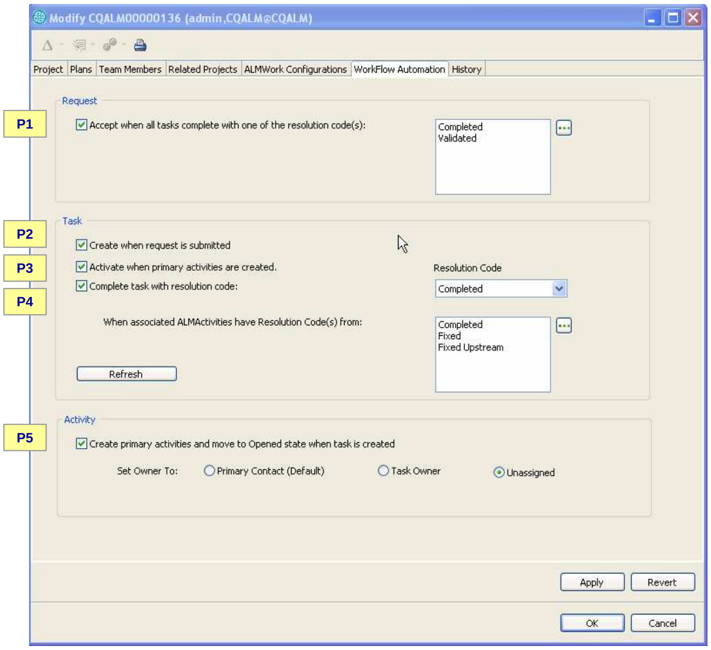
Agility Through Automation is supported in both single-site and Multisite environments. When a record is updated as the result of an automation activity in a Multisite environment, the Perl Integrator Script will make those changes at the site at which the record is mastered.

Instructions regarding the features of Agility Through Automation are provided with the documentation and release notes provided with your version of the ClearQuest ALM package. Prior to implementation, refer to the ALM package documentation of your specific ClearQuest version for information on limitations of this package with your ClearQuest schema.

ALMProject – Set policies using the WorkFlow Automation tab

<u>NOTES</u>:

- P3 – When task is activated it's priority will be set to "Unassigned" only if this priority exists. If not, the priority will remain blank.
- P5 – When primary activities are moved to open state their owner will be set as specified:
  - If "task owner" is selected and does not exist the owner will default to the Primary Contact.
  - If "Unassigned" is selected and does not exist the owner will default to the Primary Contact.



Request

☑ Accept when all tasks complete with one of the resolution code(s): Completed, Validated

Task

☑ Create when request is submitted

☑ Activate when primary activities are created.

☑ Complete task with resolution code: Resolution Code: Completed

When associated ALMActivities have Resolution Code(s) from: Completed, Fixed, Fixed Upstream

Activity

☑ Create primary activities and move to Opened state when task is created

Set Owner To: ○ Primary Contact (Default) ○ Task Owner ● Unassigned

4 Agility through automation © 2011 IBM Corporation

This is the first of the components: The Workflow Automation Tab on the ALMProject record. There are yellow boxes along the left-hand side of the screenshot. These are the identifiers of policies associated with the workflow automation. These identifiers correspond with the equivalent identifiers in the documentation. For example, P1 is an equivalent to policy 1 in the release notes.

The first policy is related to a request record. You have an option to automatically accept a request when all tasks are completed with one or more resolution codes defined in the selection box. This screenshot has a typical example of having a request accepted when all tasks are completed with the codes "Completed" or "Validated", meaning they were completed successfully.

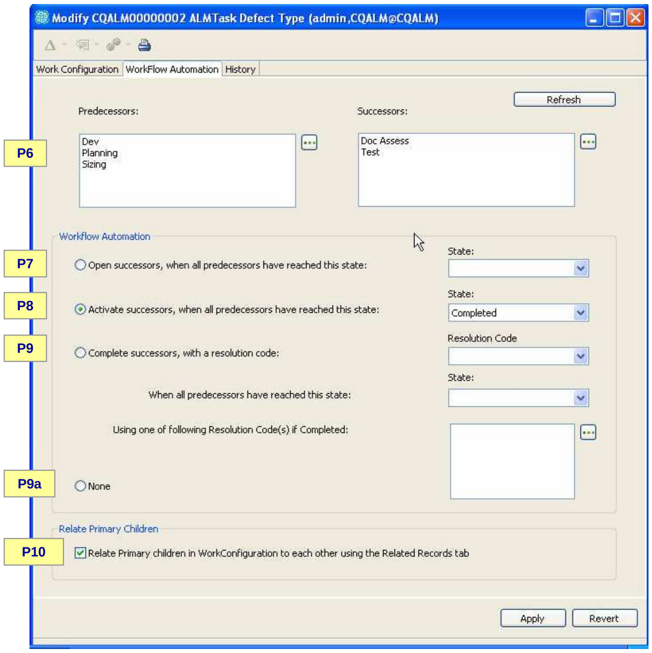
Policies 2, 3, and 4 all relate to activities for task records. You can automate the creation of tasks when a request is submitted and the activation of tasks when primary activities are created. You can also automate the completion of a task that has a particular resolution code when its associated activities are completed with similarly defined resolution codes. Policy 5 automates the creation of primary activities in an open state when a task is created. You can also define an initial owner to associate with the activities: the primary contact, the task owner, or "Unassigned".

Notice that with all of these policies available, you have the ability to automate throughout the entire lifecycle hierarchy. For example, with all of the boxes checked, you will have automation to begin the hierarchy when a request is first submitted, and automation to complete the hierarchy once the activities are completed, assuming the resolutions are successful.

ALMWorkConfiguration - Set policies on the WorkFlow Automation tab

NOTES:

- P6
  - For ALMRequest work configurations predecessors/ successors will be ALMTask types.
  - For ALMTask work configurations predecessors/ successors will be ALMActivity types.
  - For ALMActivity work configurations workflow automation can not be set.
- P10 – Only enforced for tasks and activities created after upgrading to a version of the ALM package with Agility Through Automation.



Predecessors:
Dev
Planning
Sizing

Successors:
Doc Assess
Test

Workflow Automation

○ Open successors, when all predecessors have reached this state: State:

● Activate successors, when all predecessors have reached this state: State: Completed

○ Complete successors, with a resolution code: Resolution Code

When all predecessors have reached this state: State:

Using one of following Resolution Code(s) if Completed:

○ None

Relate Primary Children

☑ Relate Primary children in WorkConfiguration to each other using the Related Records tab

Now to cover the second component, the Workflow Automation Tab for the ALMWorkConfiguration record. This is a deeper level of automation that affects the predecessors and successors for tasks and activities. Once again, the yellow boxes along the left-hand side of the screenshot indicate policy numbers as defined in the package documentation.

The screenshot shows the ALMWorkConfiguration record for a task. Workflow automation is defined for the child records associated with the record. For tasks, the child records are activities. The screenshot shows how to define the workflow automation based upon the types, states, and resolution codes of the activities associated with the task. As you can see for policy 6 lists of one or more predecessor types and successor types must be provided. The workflow automation is applied to the successor activities of the specified types when certain criteria are met for the predecessor activities with the specified types. In this example, the workflow automation will be applied to the two successor types of Doc Assess and Test when the workflow automation rules are satisfied for all three predecessor types of Dev, Planning, and Sizing.

You can set one of the three policies in the Workflow Automation area of the tab which will be actioned against the successor records. Policy 7 will open successors when all predecessors have reached a specified state. Policy 8 will activate successors when all predecessors have reached a specified state. In the screenshot example, when all of the Dev, Planning, and Sizing activities are completed, the successor activities of type Doc Assess and Test will be activated.

Policy 9 defines that successors are completed when they have a specified resolution code when all predecessors have reached a specified state and have one of the specified resolution codes.

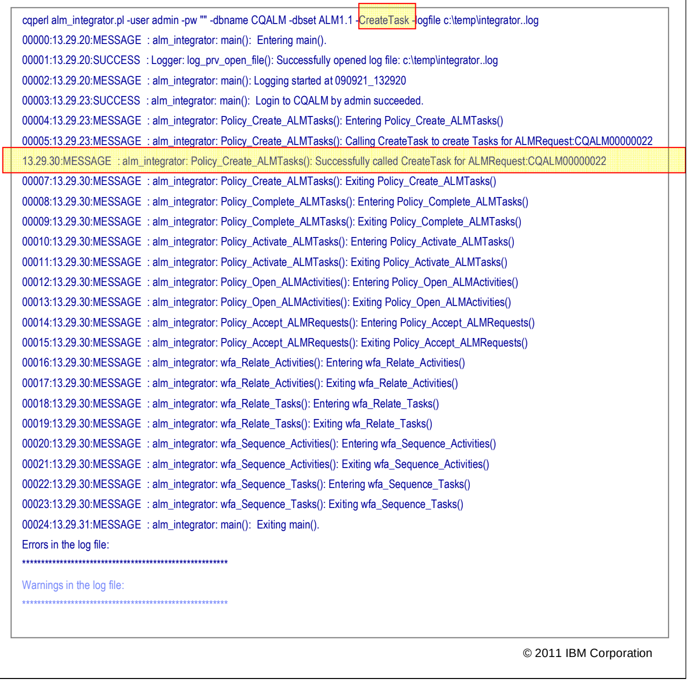
Policy 10 enables the relation of Primary children in the WorkConfiguration to each other in the Related Records Tab of those records. This makes finding records based on relationship easier. For example, you will not have to open the owning record and drill back down to find out what the related activities are. This will automatically relate any primary children; Activities if the WorkConfiguration is related to a task, or tasks if the WorkConfiguration is related to a request. This is only available with relationships created after upgrading to an ALM package that supports Agility Through Automation.

Integrator script – alm_integrator.pl

- Enforces the policies set on WorkFlow Automation tabs
- Located in .\packages\ALMWork\1.1\pkg_util
- cqperl alm_integrator.pl
  -user <user>
  -pw <pw>
  -dbname <dbname>
  -dbset <dbset>
  -CreateTask*
  -CreateActivity*
  -logfile <logfile>
  -preview
  -help

*This option must only be specified at one site

- alm_integrator.pl must be run as a scheduled job at each site
  – If not, workflow automation policies will not be enforced



```
cqperl alm_integrator.pl -user admin -pw "" -dbname CQALM -dbset ALM1.1 -CreateTask -logfile c:\temp\integrator..log
00000:13.29.20:MESSAGE  : alm_integrator: main():  Entering main().
00001:13.29.20:SUCCESS  : Logger: log_prv_open_file(): Successfully opened log file: c:\temp\integrator..log
00002:13.29.20:MESSAGE  : alm_integrator: main(): Logging started at 090921_132920
00003:13.29.23:SUCCESS  : alm_integrator: main():  Login to CQALM by admin succeeded.
00004:13.29.23:MESSAGE  : alm_integrator: Policy_Create_ALMTasks(): Entering Policy_Create_ALMTasks()
00005:13.29.23:MESSAGE  : alm_integrator: Policy_Create_ALMTasks(): Calling CreateTask to create Tasks for ALMRequest:CQALM00000022
13.29.30:MESSAGE  : alm_integrator: Policy_Create_ALMTasks(): Successfully called CreateTask for ALMRequest:CQALM00000022
00007:13.29.30:MESSAGE  : alm_integrator: Policy_Create_ALMTasks(): Exiting Policy_Create_ALMTasks()
00008:13.29.30:MESSAGE  : alm_integrator: Policy_Complete_ALMTasks(): Entering Policy_Complete_ALMTasks()
00009:13.29.30:MESSAGE  : alm_integrator: Policy_Complete_ALMTasks(): Exiting Policy_Complete_ALMTasks()
00010:13.29.30:MESSAGE  : alm_integrator: Policy_Activate_ALMTasks(): Entering Policy_Activate_ALMTasks()
00011:13.29.30:MESSAGE  : alm_integrator: Policy_Activate_ALMTasks(): Exiting Policy_Activate_ALMTasks()
00012:13.29.30:MESSAGE  : alm_integrator: Policy_Open_ALMActivities(): Entering Policy_Open_ALMActivities()
00013:13.29.30:MESSAGE  : alm_integrator: Policy_Open_ALMActivities(): Exiting Policy_Open_ALMActivities()
00014:13.29.30:MESSAGE  : alm_integrator: Policy_Accept_ALMRequests(): Entering Policy_Accept_ALMRequests()
00015:13.29.30:MESSAGE  : alm_integrator: Policy_Accept_ALMRequests(): Exiting Policy_Accept_ALMRequests()
00016:13.29.30:MESSAGE  : alm_integrator: wfa_Relate_Activities(): Entering wfa_Relate_Activities()
00017:13.29.30:MESSAGE  : alm_integrator: wfa_Relate_Activities(): Exiting wfa_Relate_Activities()
00018:13.29.30:MESSAGE  : alm_integrator: wfa_Relate_Tasks(): Entering wfa_Relate_Tasks()
00019:13.29.30:MESSAGE  : alm_integrator: wfa_Relate_Tasks(): Exiting wfa_Relate_Tasks()
00020:13.29.30:MESSAGE  : alm_integrator: wfa_Sequence_Activities(): Entering wfa_Sequence_Activities()
00021:13.29.30:MESSAGE  : alm_integrator: wfa_Sequence_Activities(): Exiting wfa_Sequence_Activities()
00022:13.29.30:MESSAGE  : alm_integrator: wfa_Sequence_Tasks(): Entering wfa_Sequence_Tasks()
00023:13.29.30:MESSAGE  : alm_integrator: wfa_Sequence_Tasks(): Exiting wfa_Sequence_Tasks()
00024:13.29.31:MESSAGE  : alm_integrator: main():  Exiting main().
Errors in the log file:
********************************************************
Warnings in the log file:
********************************************************
```

6 Agility through automation © 2011 IBM Corporation

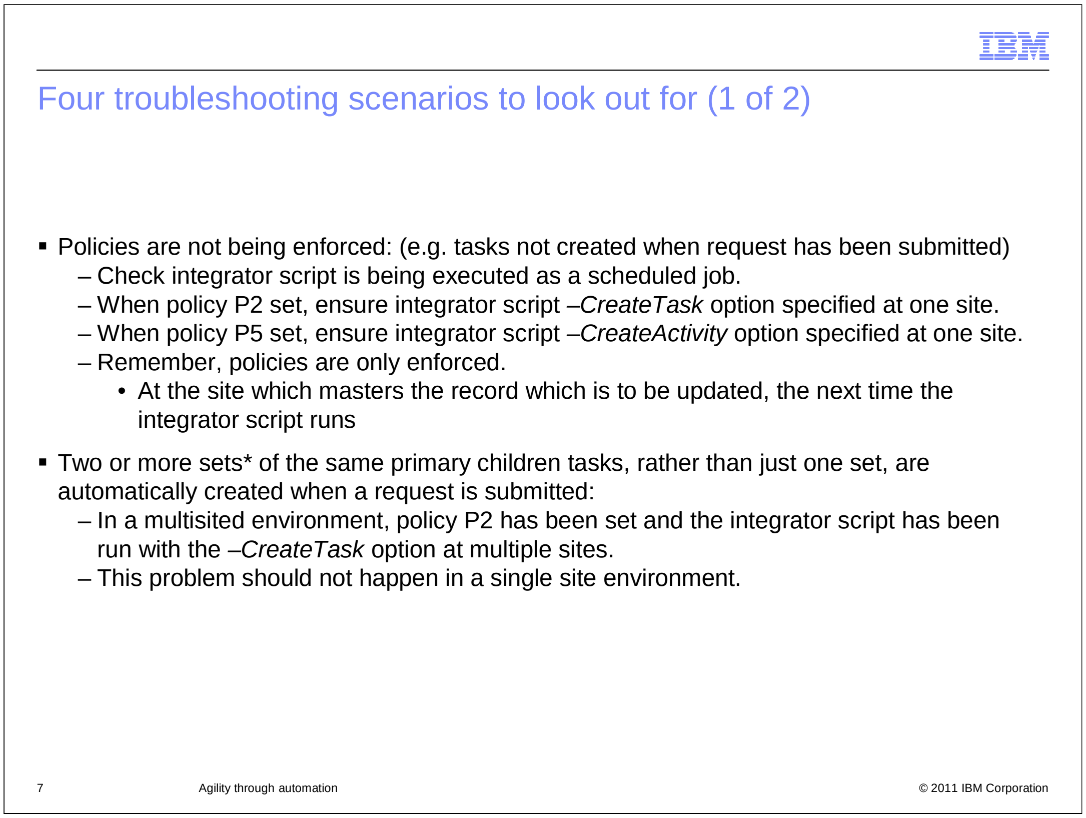
As previously stated, the configurations defined in the Workflow Automation Tabs are enforced by the Perl Integrator Script, the third component of Agility through Automation. The script is located in the packages directory of your ClearQuest installation location. It is run using the cqperl (“CQ Perl”) command line tool provided with your installation.

A ClearQuest username and password must be provided as parameters when using the script. Parameters are also available to specify a log file and its location. When using the preview parameter, all of the parameters and values passed to the script are recorded in a log file for reference or troubleshooting purposes. There is also a help parameter which provides information relating to help with the script.

The CreateTask parameter is used to run the CreateTask utility for any policies in place to automatically create tasks when a request is submitted. Likewise, the CreateActivity parameter is used to run the CreateActivity utility for any policies in place to automatically create activities when a task is submitted. These parameters must only be specified at one site in the ClearQuest environment. You should only use these parameters at the site where you want the task or activity to be mastered. This prevents the concurrent creation of two or more sets of the primary tasks and activities from different replicas.

As mentioned previously in this presentation, the Perl Integrator Script must be defined as a scheduled job at each site of your ClearQuest environment. If this does not occur, workplace automation policies will not be enforced.

There are two highlights in the screenshot example. The first is to indicate that the CreateTask parameter has been passed to the script. The second highlight is a line from the command output, showing that a task creation policy is active and has resulted in the creation of a new task.



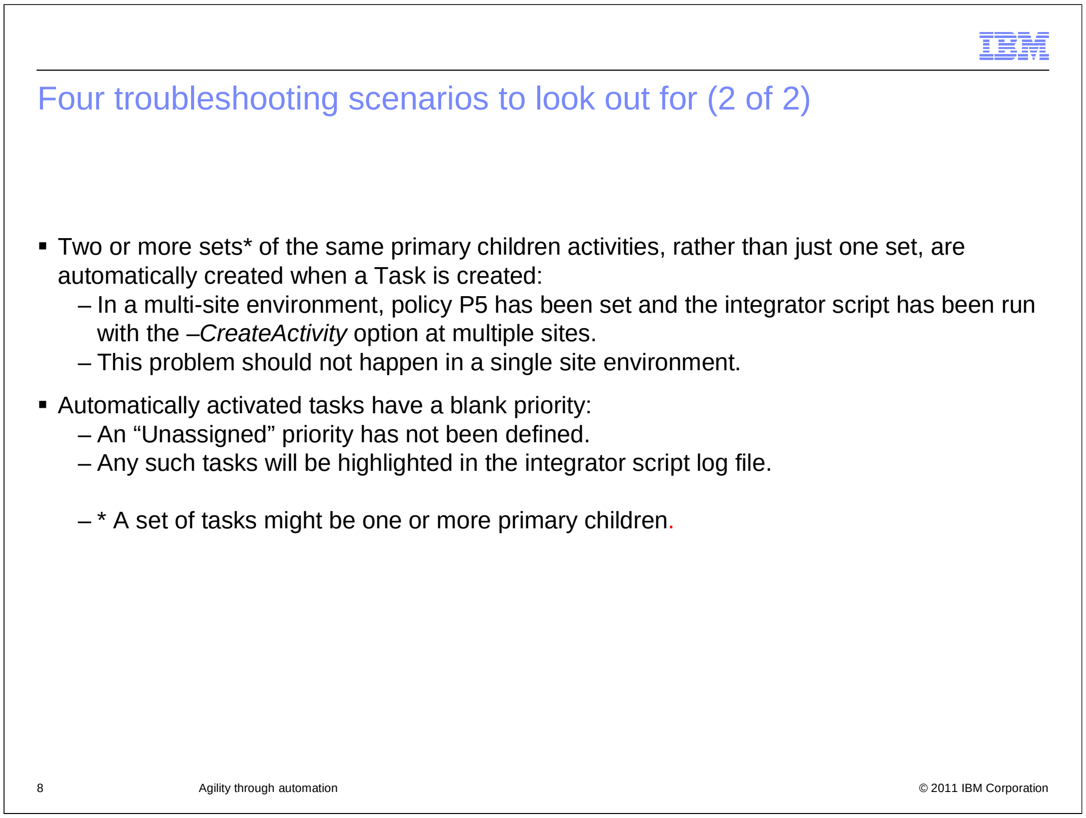
Four troubleshooting scenarios to look out for (1 of 2)

- Policies are not being enforced: (e.g. tasks not created when request has been submitted)
  - – Check integrator script is being executed as a scheduled job.
  - – When policy P2 set, ensure integrator script –*CreateTask* option specified at one site.
  - – When policy P5 set, ensure integrator script –*CreateActivity* option specified at one site.
  - – Remember, policies are only enforced.
    - At the site which masters the record which is to be updated, the next time the integrator script runs
- Two or more sets* of the same primary children tasks, rather than just one set, are automatically created when a request is submitted:
  - – In a multisited environment, policy P2 has been set and the integrator script has been run with the –*CreateTask* option at multiple sites.
  - – This problem should not happen in a single site environment.

There are four possible problems to look out for when using Agility Through Automation.

First problem is when policies are not being enforced. For example, tasks are not being created when requests are submitted. If this is happening, make sure that the Perl Integrator Script is being run as a scheduled job at all sites in the ClearQuest environment. Additionally, when using policies 2 and 5, make sure that the integration script parameters CreateTask and CreateActivity are defined only at one site. Remember that policies are only enforced at the site which masters the record which is to be updated, and are only enforced the next time the Perl Integrator script runs.

Second problem is when multiple tasks, rather than just one, are automatically created upon the submission of a request. If this happens, you are in a ClearQuest Multisite environment, and Policy 2 is defined, make sure that the parameter CreateTask is not defined for the Perl Integrator Script at multiple sites. This problem should not happen in a single-site environment.



Four troubleshooting scenarios to look out for (2 of 2)

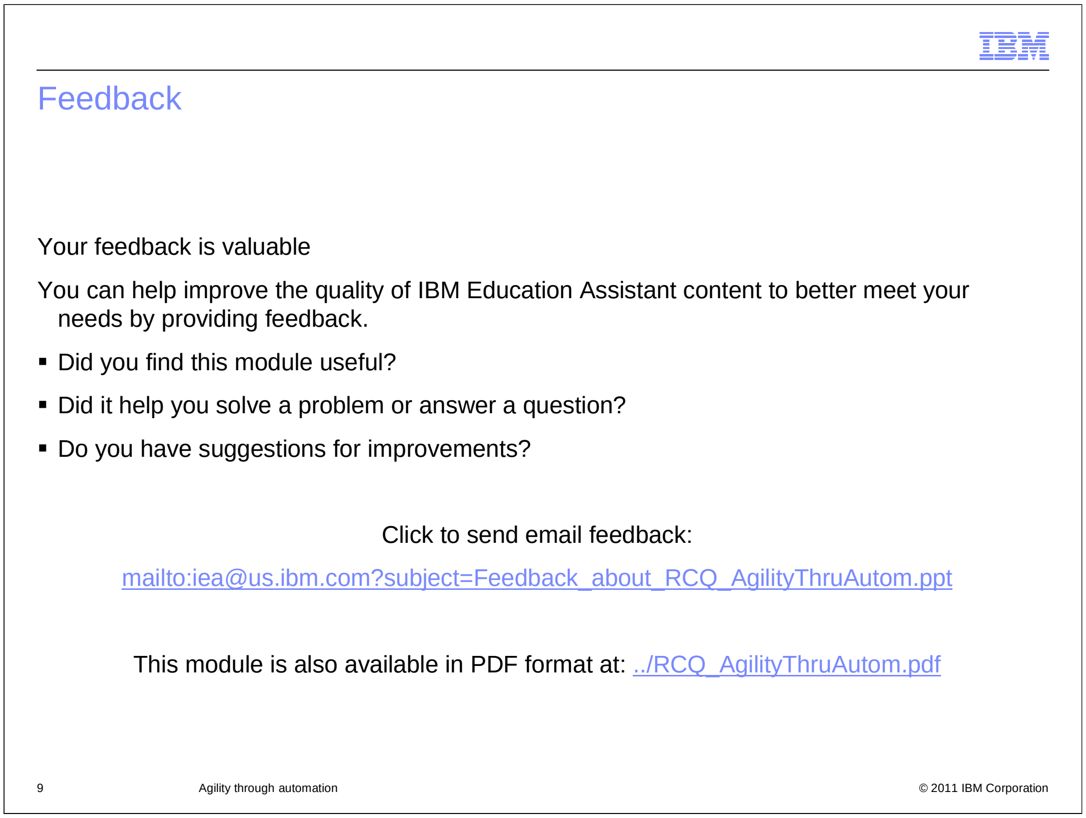
- Two or more sets* of the same primary children activities, rather than just one set, are automatically created when a Task is created:
  - In a multi-site environment, policy P5 has been set and the integrator script has been run with the –*CreateActivity* option at multiple sites.
  - This problem should not happen in a single site environment.
- Automatically activated tasks have a blank priority:
  - An “Unassigned” priority has not been defined.
  - Any such tasks will be highlighted in the integrator script log file.

  - * A set of tasks might be one or more primary children.

8 Agility through automation © 2011 IBM Corporation

The third problem is when secondary activities are automatically created when a task is opened. Similar to the previous problem, this can only happen in a Multisite environment. Make sure that if policy 5 is in place, that the CreateActivity parameter is not defined in the Perl Integrator Script for multiple sites.

The last possible problem to look out for is when automatically activated tasks have a blank priority. If this occurs, make sure that you are meeting the two “Unassigned” prerequisites mentioned earlier in this presentation. Any such tasks would be highlighted in the integrator script log file.



Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

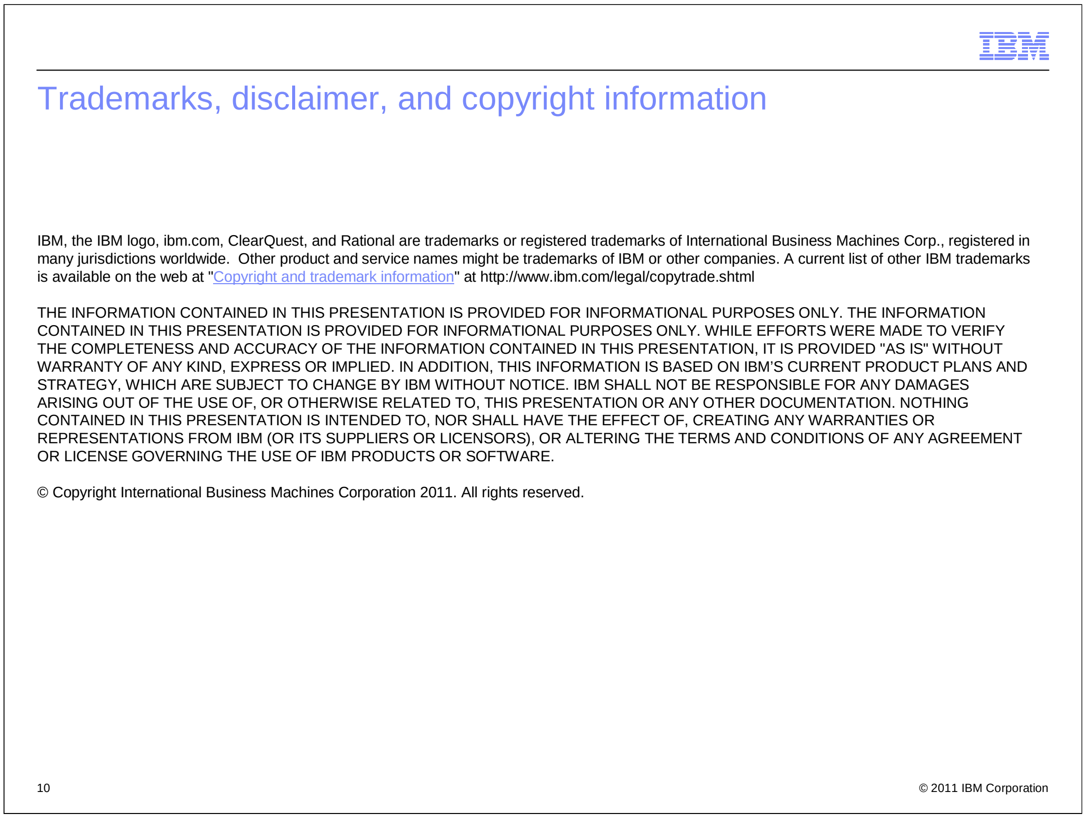
Click to send email feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_RCQ_AgilityThruAutom.ppt

This module is also available in PDF format at: ../RCQ_AgilityThruAutom.pdf

9 Agility through automation © 2011 IBM Corporation

You can help improve the quality of IBM Education Assistant content by providing feedback.



Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, ClearQuest, and Rational are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.