



**IBM Rational ClearQuest
Application Lifecycle Management Package
Version 1.1
Release Notes and User Guide**



TABLE OF CONTENTS

1	INTRODUCTION.....	4
2	NEW FEATURES IN THIS RELEASE.....	4
2.1	NEW ALM WORKFLOW AUTOMATION POLICIES.....	4
2.1.1	ALMProject RECORD POLICIES.....	4
2.1.2	ALMWorkConfiguration RECORD POLICIES.....	7
2.1.3	ALM WORKFLOW AUTOMATION POLICIES INTEGRATOR SCRIPT: ALM_INTEGRATOR.PL.....	9
2.2	ALM PROJECT SYNCHRONIZATION CAPABILITY.....	11
2.3	NEW FIELDS ADDED TO ALMActivity TO SUPPORT THE AGILE PROCESS.....	12
2.4	PREVENTION OF CHECKOUT.....	13
2.5	NEW COMMAND LINE CAPABILITY FOR ALM COPY PROJECT.....	14
2.6	ADDED ALMWorkConfiguration-> PRIMARYRole.....	15
2.7	ALMActivity IS ASSOCIATED WITH THE ALMProject.....	15
2.8	REASSIGNING A TASK THAT INCLUDES ACTIVITIES.....	16
2.9	CONFIGURING ROLE FOR REQUESTS IN ALM SCHEMA.....	16
2.10	SECURITYPolicy ON BTBuild IS SET TO SUPPORT NON-ALM BTBuild USERS.....	16
2.11	TAKING THE CURRENT PROJECT FROM THE LOWEST LEVEL OF CATEGORY RATHER THAN THE HIGHEST.....	16
2.12	SUBMITDate AND SUBMITTER FIELDS TO SELECT ALM RECORD TYPES.....	17
2.13	ALM SCHEMA FIELDS SUPPORTING THE AGILE PROCESS.....	17
2.14	ROLE SUPPORT.....	17
2.15	ALM COMMENTING FEATURE REDESIGNED.....	17
2.16	FIELD ON THE ALMProject RECORD THAT LISTS THE RESERVED RECORD FOR THE ALL FUNCTIONALITY.....	18

2.17 TYPE RECORDS ARE PROJECT DEPENDENT.....19

3 FIXES IN THIS RELEASE.....20

4 UPGRADE CONSIDERATIONS.....22

1 Introduction

The application lifecycle management (ALM) packages provide a solution for ensuring ALM best practices in your Rational® ClearQuest® change management system. Application Lifecycle Management is a key enabler for streamlining a team's ability to produce a release of software. It coordinates software development activities and assets to produce and manage software applications throughout their life cycles.

The ClearQuest ALM packages provide a role-based process and security model that offers optimal performance with a collection of tightly integrated record types that create a foundation for managing software development projects.

This release of ClearQuest ClearQuest ALM includes new features and fixes to problems found in the previous release.

2 New features in this release

2.1 *New ALM Workflow Automation policies*

This release introduces a new set of policies that enable the automation of steps in the workflow associated with ALMRequest, ALMTask and ALMActivity records. These policies can be configured on the ALMProject and ALMWorkConfiguration records. The policies can then be enforced using a script called `alm_integrator.pl`. This script should be run as a scheduled job at a single site, or at multiple sites if using ClearQuest Multisite.

2.1.1 **ALMProject Record Policies**

A new tab called “Workflow Automation” (See Figure 1) has been added to the ALMProject record. The policies are grouped according to the record type to which they apply, as follows:

P1) Accept ALMRequest when all Tasks are completed.

This policy accepts the ALMRequest when all its associated tasks have been completed with the specified resolution code. If any of the associated tasks have been completed with a different resolution code, the policy is not triggered.

A typical use of this policy is to close the request <when?> its associated tasks have been completed successfully, thereby saving the request owner from having to do so manually.

P2) Create ALMTask set using WorkConfiguration record when ALMRequest is created.

This policy creates the primary tasks for a request at the time when the request is created.

This policy simplifies the role of triaging the request for those projects where typically testers are raising defects and it is known which projects should work on completing the tasks.

P3) Activate an ALMTask when the Primary Children Activities are created (Unassigned priority will be set).

This policy activates the task when the primary children activities are created. Used in conjunction with policies P2 and P5 this policy automates the triaging of a request, building the request-task-activity hierarchy without manual intervention.

Note: When this policy is enforced, the integrator script activates the task even if the priority has not been specified. In such cases the script sets the priority to the value **Unassigned**, provided that such a priority has been defined for task records before the script is run.

To define the “Unassigned” priority:

In ClearQuest click the “New” button, select ALMType, and specify the following details:

Record Type: ALMTask

Type Indicator: Priority

Type Label*: Unassigned

*If the “Unassigned” type label is not provided in the dropdown, click the NEW LABEL button to create one.

If the “Unassigned” priority has not been set before the integrator script is run and the task has a blank priority, the policy is still enforced and a warning message is printed to the log file of the integrator script. In such a case the “Unassigned” priority should be defined as shown above. Any tasks with blank priorities that have been activated by the script should subsequently be updated to set their priority to a valid value.

P4) Complete Tasks with a specified Resolution Code when all Activities are completed.

This policy completes the task with a given resolution code when all its associated children activities have been completed with the specified resolution code.

A typical use of this policy is to complete the task when all the associated activities have been completed successfully.

P5) Create Primary Activities when the task is created and set the activity owner to

- a. Primary Contact (Default)
- b. Task Owner
- c. Unassigned

This policy creates the primary activities associated with the task when the task is created. Each activity owner is then set to either the values of a, b, or c above. The primary contact is defined from the primary role on the activity's ALMWorkConfiguration record and is the default value.

Note 1: A ClearQuest userid of **Unassigned** must exist in order for the activity owner id to be set to **Unassigned**. If such an id neither exists nor is a valid id for the project associated with the activity, the integrator script overrides this option and sets the activity owner to the Primary Contact.

Note 2: If the activity owner option selected is to assign the "Task Owner" and the task owner is not valid for the activity, the integrator script overrides this option and sets the activity owner to the Primary Contact.

The screenshot displays the 'Workflow Automation' tab within the ALMProject interface. The tab is divided into three main sections: Request, Task, and Activity, each with its own set of configuration options.

- Request Section:** Labeled 'P1', it contains a checkbox 'Accept when all tasks complete with one of the resolution code(s):'. To the right of this checkbox is a list box showing resolution codes: Complete, Duplicate, Fixed, Rejected, and Unreproducible.
- Task Section:** Labeled 'P2', 'P3', and 'P4', it contains three checkboxes: 'Create when request is submitted', 'Activate when primary activities are created.', and 'Complete task with resolution code:'. To the right of the third checkbox is a 'Resolution Code' dropdown menu. Below these checkboxes is a label 'When associated ALMActivities have Resolution Code(s) from:' followed by another list box showing the same resolution codes. A 'Refresh' button is located below the list box.
- Activity Section:** Labeled 'P5', it contains a checkbox 'Create primary activities and move to Opened state when task is created'. Below this checkbox is a label 'Set Owner To:' followed by three radio buttons: 'Primary Contact (Default)', 'Task Owner', and 'Unassigned'.

Figure 1 – The ALMProject "Workflow Automation" tab

2.1.2 ALMWorkConfiguration Record Policies

A new tab called “Workflow Automation” (See Figure 2) has been added to the ALMWorkConfiguration record containing the policies associated with the record.

P6) Supports definition of child predecessors and child successors associated with a record type.

This policy provides the definition of workflow automation within a record type (e.g. ALMTask) by allowing the configuration of predecessor/successor lists based upon the type labels (e.g. “Develop, Fix”, “Review Fix”, “Test”) of the child record type (e.g. ALMAActivity) that is associated with the record type.

Therefore, for an ALMRequest work configuration, this policy allows the definition of task predecessors/ successors. Likewise, for an ALMTask work configuration this policy allows the definition of activity predecessors/ successors. Because ALMAActivities have no child record type, this policy is not available for ALMAActivity work configurations.

The setting of predecessors and successors defines a workflow within the child record type that you can use with the policies that are described below to automate certain actions; for example, to activate an activity when all its predecessors have been completed.

P7) Supports Opening of successor record(s) when all predecessor(s) have attained a specifiable state.

This policy opens the successor record(s) defined by the ALMWorkConfiguration after all its predecessors have reached the specified state. For example, you can specify that a “Test” activity should be opened when all its predecessor activities (e.g. “Develop Fix”, “Review Fix”) have been completed.

P8) Supports activating of successor record(s) when all predecessors have attained a specifiable state.

This is similar to policy P7 but the policy activates the record(s) instead of opening it.

P9) Supports completing successor record(s) with a specifiable Resolution Code when all predecessors have attained a specifiable state.

This is similar to policies P7 and P8 above but instead the policy completes the successor record(s) with the given resolution code when the predecessor(s) have reached a specified state. Where the state specified for the predecessor is “Completed” then one or more resolution codes must also be provided to enforce the policy.

P10) Relates Primary children in the WorkConfiguration to each other.

This policy creates a relationship between the primary children associated with the current ALMWorkConfiguration. The policy is only available for ALMRequest and ALM

Task work configurations whose primary children are task and activity records, respectively. Once created, the relationships are shown on the “Related Records” tab of each child record.

For example, for an ALMTask work configuration, three primary children (activities) of types “Develop Fix”, “Review Fix” and “Test Fix” are defined. When this policy is selected, for each activity a link to the other two activities is provided in the “Related Records” tab. This allows easy access to all activity details without having to manually create relationships between the activities.

Note: This policy works only for tasks and activities created after upgrading to ALM1.1.

The refresh button on both Workflow Automation tabs is used to refresh the dropdown fields when accessing ALM from the web.

Work Configuration Workflow Automation History

Refresh

Predecessors:

P6 Implement

Successors:

Build
Create Test Case

Workflow Automation

P7 ☐ Open successors, when all predecessors have reached this state: State:

P8 ☐ Activate successors, when all predecessors have reached this state: State:

P9 ☐ Complete successors, with a resolution code: Resolution Code
State:

When all predecessors have reached this state: State:

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

☒ None

Relate Primary Children

P10 ☐ Relate Primary children in WorkConfiguration to each other using the Related Records tab

Figure 2 – The ALMWorkConfiguration “Workflow Automation” tab

2.1.3 ALM Workflow Automation Policies Integrator Script: alm_integrator.pl

The integrator script enforces the workflow automation policies defined on the ALMProject and ALMWorkConfiguration records. The script is located in the directory `.\ALMWork\1.1\pkg_util\`.

Run **ratlperl alm_integrator.pl –help** to display all options:

`-user <user>` The login id of the ClearQuest user.
`-pw <password>` The password of the ClearQuest user.

ClearQuest ALM Release Notes

Version 1.1

- dbname* <*dbname*> The schema repository connection to the database.
- dbset* <*dbset*> The user database connection.
- CreateTask* Allow this script to run ALM CreateTask Record script alias. Only one site should enforce this policy in a replicated environment.
- CreateActivity* Allow this script to run ALM CreateActivity Record script alias. Only one site should enforce this policy in a replicated environment.
- logfile* <*logfile*> Log output to the specified file in addition to printing the output to the console.
- preview* Display the user input in the log file.
- help* Print this message and exit.

Example 1:

In this example no policies are enabled so the script simply returns, not having updated any records.

```
>ratlperl alm_integrator.pl -user admin -pw "" -dbname ClearQuestALM -dbset ALM1.1 -logfile c:\temp\integrator.log
00000:13.21.48:MESSAGE : alm_integrator: main(): Entering main().
00001:13.21.48:SUCCESS : Logger: log_prv_open_file(): Successfully opened log file: c:\temp\integrator.log
00002:13.21.48:MESSAGE : alm_integrator: main(): Logging started at 090921_132148
00003:13.21.50:SUCCESS : alm_integrator: main(): Login to ClearQuestALM by admin succeeded.
00004:13.21.50:MESSAGE : alm_integrator: Policy_Complete_ALMTasks(): Entering Policy_Complete_ALMTasks()
00005:13.21.50:MESSAGE : alm_integrator: Policy_Complete_ALMTasks(): Exiting Policy_Complete_ALMTasks()
00006:13.21.50:MESSAGE : alm_integrator: Policy_Activate_ALMTasks(): Entering Policy_Activate_ALMTasks()
00007:13.21.50:MESSAGE : alm_integrator: Policy_Activate_ALMTasks(): Exiting Policy_Activate_ALMTasks()
00008:13.21.50:MESSAGE : alm_integrator: Policy_Open_ALMAactivities(): Entering Policy_Open_ALMAactivities()
00009:13.21.50:MESSAGE : alm_integrator: Policy_Open_ALMAactivities(): Exiting Policy_Open_ALMAactivities()
00010:13.21.50:MESSAGE : alm_integrator: Policy_Accept_ALMRequests(): Entering Policy_Accept_ALMRequests()
00011:13.21.50:MESSAGE : alm_integrator: Policy_Accept_ALMRequests(): Exiting Policy_Accept_ALMRequests()
00012:13.21.50:MESSAGE : alm_integrator: wfa_Relate_Activities(): Entering wfa_Relate_Activities()
00013:13.21.50:MESSAGE : alm_integrator: wfa_Relate_Activities(): Exiting wfa_Relate_Activities()
00014:13.21.50:MESSAGE : alm_integrator: wfa_Relate_Tasks(): Entering wfa_Relate_Tasks()
00015:13.21.50:MESSAGE : alm_integrator: wfa_Relate_Tasks(): Exiting wfa_Relate_Tasks()
00016:13.21.50:MESSAGE : alm_integrator: wfa_Sequence_Activities(): Entering wfa_Sequence_Activities()
00017:13.21.50:MESSAGE : alm_integrator: wfa_Sequence_Activities(): Exiting wfa_Sequence_Activities()
00018:13.21.50:MESSAGE : alm_integrator: wfa_Sequence_Tasks(): Entering wfa_Sequence_Tasks()
00019:13.21.51:MESSAGE : alm_integrator: wfa_Sequence_Tasks(): Exiting wfa_Sequence_Tasks()
00020:13.21.51:MESSAGE : alm_integrator: main(): Exiting main().
```

Errors in the log file:

Warnings in the log file:

Example 2:

In this example, policy “P1” is turned on and the script is invoked with-CreateTask. When a new request is submitted, the integrator script creates the primary task(s) for that request:

ClearQuest ALM Release Notes

Version 1.1



```
ratlperl alm_integrator.pl -user admin -pw "" -dbname ClearQuestALM -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 ClearQuestALM 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:ClearQuestALM00000022
13.29.30:MESSAGE : alm_integrator: Policy_Create_ALMTasks(): Successfully called CreateTask for
ALMRequest:ClearQuestALM00000022
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:

Note1: Use Windows “Scheduled Task” to create a task that runs periodically and executes alm_integrator.pl at each site.

Note2: The log file overwrites any existing log file each time the script runs. If you need to maintain one log only you should redirect the standard output and error to your logfile, for example:

```
ratlperl alm_integrator.pl -user admin -pw "" -dbname ClearQuestALM -dbset ALM1.1 -
CreateTask > alm_integrator.log 2>&1
```

Monitor this log file for errors and warning messages, which are highlighted in the file. This file can get quite large and should therefore be maintained regularly, either after it has been checked or after the contents have been archived.

2.2 ALM Project synchronization capability

The Copy Project utility in ALM minimizes the effort and time required to copy an ALM project. However in ALM 1.0, new ALMPhase, ALMRole, and ALMWorkConfigurations that are added to the source project after the initial copy cannot be copied to an existing target project.

This release introduces the alm_syncprojects.pl script to copy these additional records.

This script can copy new ALMPhase, ALMRole, and ALMWorkConfigurations from one source ALM project to a target ALM project.

The script is located in the directory .ALMWork\1.1\pkg_util\.

To display the options:

```
ratlperl alm_syncprojects.pl -help
-projadmin    The login id of the ProjectAdmin.
-pw           The password of the ProjectAdmin.
-dbname       The name of the user database.
-dbset        The name of the database set or connection that contains the database.
-fromprojid   ALMProject Id copying from.
-toprojid     ALMProject Id copying to.
-help         This is an optional flag for the help.
Example:
>ratlperl alm_syncprojects.pl -projadmin ProjectAdmin -pw "" -dbname ClearQuestALM -dbset TestSchema
-fromprojid ClearQuestALM00000001 -toprojid ClearQuestALM00000019
```

The following ALMPhase(s) will be copied from Project:ClearQuestALM00000001 to Project:ClearQuestALM00000019

```
ClearQuestALM00000001 Test2
Calling CopyEntity
Number of entities created in the CopyEntity Array = 1
Entity DefName is: ALMPhase
DisplayName is: ClearQuestALM00000001 Test2
Call ValidateEntities
Call CommitEntities
CopyEntity was successful.
```

```
ALMPhase is successfully copied from ClearQuestALM00000001 to ClearQuestALM00000019.
Found no new ALMRole(s) to copy from ClearQuestALM00000001 to ClearQuestALM00000019.
Found no new ALMWorkConfiguration(s) to copy from ClearQuestALM00000001 to ClearQuestALM00000019.
```

2.3 New fields added to ALMActivity to support the Agile process

New fields have been added to the ALM Activity tab. (See Figure 3)

The screenshot shows the 'Activity' tab in the ClearQuest ALM interface. At the top, there is a navigation bar with tabs: Activity, Related Records, Resolution, Baseline and Build, History, Comments, and Attachments. Below this is a '*Headline:' field. A 'Project:' label is followed by a 'Get Project from Task' button. Below the project section is a table with columns: ID, Name, Category, Release, and Obsolete. The main section is titled 'Activity's Fixed-in Context' and contains several fields: 'Phase Assigned to:' (a dropdown menu), 'Iteration Assigned to:' (a dropdown menu), 'Estimated Effort:' (a text input), 'Actual Effort:' (a text input), and 'Due Date:' (a date input). A yellow rectangular box highlights the 'Phase Assigned to:', 'Iteration Assigned to:', 'Estimated Effort:', and 'Actual Effort:' fields. To the right of the 'Due Date:' field are two small icons: a green double arrow and a document icon.

Figure 3 – New ALMActivity fields to support the Agile process

The new fields are:

Phase Assigned to: This is similar to **Phase assigned to** on a task and specifies the phase in which the activity should be completed in.

Iteration Assigned to: This is similar to **Iteration assigned to** on a task and specifies the Iteration in which the activity should be completed in.

Estimated Effort: Estimated time required to complete the activity.

Actual Effort: Actual time that was required to complete the activity

Due Date: The date by which the activity should be completed

Priority: Specifies the priority for completion of the activity(not shown).

2.4 Prevention of checkout

When a user tries to check out a file in ClearCase, this policy prevent the checkout from succeeding if the UCM project on the ALMActivity and the UCM project on the ALMProject do not match.

To enable this policy:

1. Important: Test the following schema changes and only push the changes to production after validation.
2. Locate the following UCM global hooks:

In UCM_ChkBeforeWorkOn (VB) global hook add:

Add the following lines to the beginning of the hook after the header section using ClearQuest designer:

```
UCM_ChkBeforeWorkOn = ALM_UCM_ChkBeforeWorkOn(entity_type, entity_id)
if UCM_ChkBeforeWorkOn <> "" then
    exit function
end if
```

In UCU_ChkBeforeWorkOn (PERL) global hook add:

Add the following lines to the beginning of the hook but after variable declaration using ClearQuest designer:

```
$result = ALM_UCU_ChkBeforeWorkOn(entity_type, entity_id);
if ($result ne ""){
    return $result;
}
```

3. Check in the schema and upgrade your test database.
4. Test the changes.

In a ClearCase view, check out a file using an ALMAActivity whose UCM project does not match the UCM project on the ALMProject for the same activity.

The checkout should fail.

Now change the UCM project to match for both ALMAActivity and ALMProject.

The check out should succeed.

A check out can sometimes succeed even if the UCM project associated with the ALMAActivity and ALMProject are not identical. This could be the case if the ALMAActivity had checkouts before the policy was implemented and the upgrade was performed.

If you have customized the above UCM global scripts before this change, also retest your customization after applying the changes in your test environment.

2.5 New command line capability for ALM Copy Project.

The script alm_copyproject.pl is located in the \ALMWork\1.1\pkg_util\ directory.

This script copies ALMProject and its ALMPhase, ALMRole, and ALMWorkConfigurations from one source ALM project to a target ALM project. To display the options:

```
ratlperl alm_copyproject.pl -help
alm_copyproject.pl
{-projadmin <projadmin>}
{-pw <password>}
{-dbname <database name>}
{-dbset <database set>}
{-projectid <projectid>}
[-clonedprojname [<clonedprojname>]]
[-category [<category>]]
[-release [<release>]]
[-help]
```

Example:

```
>ratlperl alm_copyproject.pl -projadmin admin -pw "" -dbname ClearQuestALM -dbset ALM1.1 -projectid
ClearQuestALM00000001 -clonedprojname "Test Copy Project"
```

```
-Category "Our OpenUP Solution" -release "OpenUP Release 2"
```

The copy was successful!. The new project ID is: ClearQuestALM000000021

2.6 Added ALMWorkConfiguration-> PrimaryRole

Currently the owner of a task is determined using the first RoleLabel in the alphabetically sorted sequence of the ALMRole->RoleLabels associated with the ALMWorkConfiguration. This amounts to a random determination of the default task owner.

The new PrimaryRole field specifies which Role from the set of ALMWorkConfiguration-Roles is to be used as the Role whose Role->Primary will be set as default owner of a Task.

If there is no ALMWorkConfiguration->PrimaryRole value, the existing ALMWorkConfiguration->ALMRole-Primary mechanism will be used as the fallback <fallback=default?>.

Important Note: The PrimaryRole is a new mandatory field in this release. When copying ALMProjects using copy project function, set this field on all ALMWorkConfigurations record of Record type ALMTask and ALMActivity that were previously created. If this is not done prior to a copy, the copy project will fail and only, create a partial copy of the source project.

2.7 ALMActivity is associated with the ALMProject

This new feature associates an ALMActivity with an ALMProject when adding to source or checking out a file in a ClearQuest-Enabled UCM project. For this feature to work, the following conditions have to be met:

- There must be only one ALMProject associated with the UCM Project in which you are performing a check out or add to source operation
- ALMProject must have a DefaultRequest and a DefaultTask.

2.8 Reassigning a Task that includes Activities

If the **Activity->Project** is different from the **Activity->Task->Project**, then a click of a new **Activity->Sync from Task** button will copy the task's project onto the activity

If the **Activity->Project->UCM_Project** is nonnull and is different from the **Task->Project->UCM_Project**, an error is returned advising the user to ensure that the appropriate UCM_Project is set on the ALMProject before it is referenced on the current activity.

2.9 Configuring role for requests in ALM schema

If a newly OPTIONAL ALMRole is set on an ALMWorkConfiguration for an ALMRequest, non-Owners of ALMRequests would have to be ALMRole->Members or in ALMRole->Groups with ALMRole->ApprovedActions in order to initiate certain actions on the referenced ALMRequest

Note: Now for all ALMWorkConfiguration records of TypeIndicator='Type' and record_Type='ALMRequest' you will need one (or more) Roles with Approved Actions. For example; you will need to include the ApprovedAction of 'ALMRequest::CreateTask' for the Triage Role in the ToDo Practice.

2.10 SecurityPolicy on BTBuild is set to support non-ALM BTBuild users

The "Everyone" security policy is now required to exist in the ClearQuest database in order to submit/modify a **BTBuild**. This is now the constant default value for **BTBuild->SecurityPolicy**.

2.11 Taking the Current Project from the lowest level of Category rather than the highest

The **Category->CurrentProject** is used to set the **ALMRequest->Project**, regardless of whether a **Category** has any **SuperCategories**. The advantage of this approach is that the more specific the **Category**, that **Category->CurrentProject** is used (if existing) and therefore the most specific, smallest set of **Members** and **ALMRole->Primary Member** values is set. This provides more support for Component Based development.

2.12 SubmitDate and Submitter fields to select ALM record types

These fields have been added to **ALMRequest**, **ALMTask** and **ALMActivity** record types.

2.13 ALM schema fields supporting the Agile process

The fields **ALMRequest: Priority**, **EstimatedEffort (int)**, and **ActualEffort (int)** have been added to the ALM schema to support the Agile process

A key feature of the Agile process is to record the estimated effort and the actual effort required. Additionally the **Priority** field on the Activity and Request records supports the Upper and Lower Tier 1-record models which are ClearQuest ALM Agile models.

2.14 Role support

In this release, permissions can be configured on the **ALMRequest** record type. By setting up the **ALMRole** record type, non-owners of **ALMRequests** can be approved to initiate certain actions on **ALMRequests**.

2.15 ALM commenting feature redesigned

ALM's commenting feature has been redesigned to be more intuitive – just like any messaging functionality. This design (see the **ALMComment** figure below) also enables better accessibility of referenced records; messages responded to are at one's fingertips
<this is bizarre, but I don't understand it enough to suggest how to revise>

Date Submitted	Need Response?	Author	Responses
2009-07-15 20:30:17 GMT	No	admin	And Another
2009-07-15 22:27:54 GMT	No	admin	A Response

Figure 4 – ALMComment record redesigned

2.16 Field on the ALMProject record that lists the Reserved record for the ALL functionality

Rejecting ALMRequests requires an ALMProject, to which the rejected requests are triaged. This release makes designating such ALMProjects easier. Figure 5 shows an added option on the ALMProject record that allows project administrators choose which project to specify as the target project for all rejected requests. The figure below highlights this configuration on the ALMProject record form.

The screenshot shows the 'Project' tab in the ClearQuest ALM interface. The form contains the following fields and controls:

- Project ID:** Text field with value 'ALMt100000008'.
- Name:** Text field with value 'Projekt x'.
- Category:** Dropdown menu with value 'x' and a 'New' button.
- Release:** Dropdown menu with value '1.x' and a 'New' button.
- Owner:** Dropdown menu with value 'admin'.
- Current Phase:** Dropdown menu.
- State:** Text field with value 'Pending'.
- Status:** Dropdown menu.
- Obsolete:** Checkbox, currently unchecked.
- Assign to all Rejected Requests?:** Checkbox, currently checked. This checkbox is highlighted with a yellow rectangle.
- Description:** Large text area.
- Start Date:** Text field with a calendar icon.
- End Date:** Text field with a calendar icon.
- Report Tags:** Section with 'New', 'Add', and 'Remove' buttons, and a table with a 'Name' header and an empty row.

Figure 5 – New field to specify a target project for all rejected requests

2.17 Type records are *Project dependent*

The Request's severity and Task's priority choice-lists are dependent on their corresponding ALMProjects by default. The choice-list values are configurable on the ALMProject record (see Figure 6). The functionality gives users a more finite <constrained?> set of choices for the ALMRequest's severity and ALMTask's priority field values.

Project Plans Team Members Related Projects ALMWork Configurations Workflow Automation History

UCM_Project: [Dropdown]

Default Parents :

Default Request:

Project ID	Request ID	Type	State	Owner	Headline

Default Task:

Project ID	Task ID	Type	State	Owner	Headline

Work Configurations:

Record Type	Type Label	Obsolete
ALMActivity	Defect	NO
ALMRequest	Defect	NO
ALMTask	Defect	NO
ALMTask	Test	YES

Choice Lists Configurations

Requests Severity: [New]

- ALMRequest High Severity
- ALMRequest Low Severity

Tasks Priority: [New]

- ALMTask High Priority
- ALMTask Medium Priority

Figure 6 – Request's severity and Task's priority choice-lists

3 Fixes in this release

This release includes fixes for problems found in the previous release. Some of the more significant fixes are:

- Users can no longer drill down below the list of available Category values on the Request. **ALMRequest Drill Down** button now stops at the bottom of the tree instead of drilling past the bottom as before. The bottom-most Category will also be added to the Category Path controls for querying purposes as well as displaying in the **ALMRequest->Category** form control. There is also a **Drill Back** button that drills back up.

- ClearQuest ALM now respects the **Obsolete** setting when retrieving **ALMWorkConfiguration** records. We also keep track of all childtypes that have been created during the execution.
- ALMComment form now displays the unique name of the comment
- BTBuild no longer checks Approval if it isn't associated with an ALMProject
- ALM schema circular Web dependencies have been fixed.
- Fixed error handling to help customers configure correctly. The fix now looks for errors in the array and displays them.
- Fixed error where SetEntityFieldVals Global Hook returned the wrong entityDefName.
- Fixed CM Server error when deleting two stateless records with same name.
- ALMProject::SetDefault Action is now allowed for all users like <like what?>. Setting this Action on every Role is no longer needed. Every user can do ALMProject->SetDefault Action at any time.
- ALMProject is now a mandatory field in a BTBuild record. With this fix, when a value is set on the baseline field, the project field is mandatory on the BTBuild.
- When creating an ALMActivity using ClearCase, the owner is already set when the Activity form opens. Now if the user sets the Activity Type, the owner clears. The fix will not clear the owner if the owner is a valid owner in the owner's choice list.
- CopyProject no longer copies Tasks associated with Requests.
- ALMProject::SetDefault Action is now allowed for all users. A user no longer needs to have ALMProject::SetDefault added to the approved actions for the role.
- ALMWorkConfiguration->TypeLabelChoice list is now sorted
- When completing a task, the security policy is now set on the comments generated against the activities.
- Request->CategoryTypeLabel is no longer red labeled when clicking Category Drop Down first.

4 Upgrade considerations

Consider the following when upgrading to this release:

1. ALMRequest::CreateTask must now be specifically added to ALMRole->ApprovedActions for roles that will need to perform this action. In ClearQuest ALM 1.0 that was not required.
2. ALMProject::SetDefault is no longer needed in any ALMRole. Every role has the permission to set default project. There is no need to remove this approved action from an ALMRole->ApprovedActions list, but there is no need to add it any more either.
3. All ALMWorkConfigurations must have a PrimaryRole specified before a Copy Project is performed. The copy project will fail if this field is blank for ALMWorkConfigurations with Record type of ALMTask and ALMActivity.
4. In ALM 1.1, at least one ALMProject must be specified with its "AssignedToAllRejectedRequests" checked.
5. You may not use Copy Project (from Related Projects tab) on a project with AssignedToAllRejectedRequests checked without first unchecking this checkbox, performing Copy Project, and clicking revert on the source ALMProject to get your original settings back.
6. If copying a project using alm_copyproject.pl script or the project wizard, you'll need to save the project with the selection above unchecked and restore the original setting after copy project is completed.
7. If you have marked your 'ALL' Project Obsolete (used without option for Rejected Requests in ClearQuest ALM 1.0), you will need to uncheck the Category->Obsolete checkbox before modifying your ALMProject or you will not see the 'ALL' Category as an option on that Project and therefore will not be able to set the "AssignedToAllRejectedRequests" checkbox. This is because Obsolete settings now works properly where it did not work properly in ClearQuest ALM 1.0. Once you have updated your 'ALL' Project to be the "AssignedToAllRejectedRequests" Project, you can once again Modify the ALMCategory->Obsolete check box so that the 'ALL' Category is not a nuisance choice to your users. The 'ALL' Project will work properly so long as you do not Modify or State transition it. If you do need to change the 'ALL' Project, repeat this process
8. In order to see Priority choices on the ALMRequest and ALMActivity (a new ClearQuest ALM 1.1 feature), you need to add new ALMType records with

- Record Type set to ALMRequest and ALMActivity record type and Type Indicator set to Priority.
9. There is a special procedure for applying the ALMWork, Attachments, and EmailPlus packages from the DCF download site when using UTF-8 code page on Rational ClearQuest V7.1.1. The guidance on how to do this is available from the Rational Support site at
http://www-01.ibm.com/support/docview.wss?rs=0&q1=1414422&uid=swg21414422&loc=en_US&cs=utf-8&cc=us&lang=all
 10. A failed hook execution issue has been identified when using ClearQuest ALM packages (downloaded from the DCF download site), with the ClearQuest schema repository and user database managed with Microsoft SQL Server 2005. We have isolated the issue to the following configurations:
 - Windows 2003 Simplified Chinese OS + ClearQuest v7.1.1 + LMWork 1.1.3 package + SQL Server 2005
 - Windows 2003 Japanese OS (with or without WindowsServer2003-KB945142-v2-x86-JPN.exe) + ClearQuest v7.1.1 + ALMWork 1.1.2 package + SQL Server 2005
 - Window 2003 Simplified Chinese OS + ClearQuest v7.1.1 + ALMWork 1.1 package + SQL Server 2005

When creating ALMRequest the user might encounter the following errors:
"Problem creating resource: Execution of a hook failed during the action Open. It was the FIELD_PERMISSION hook of the field AttachmentsErrorText, attached to the ALMRequest "SAMPL00000020". The reason for the failure was: Can't lcoate object method "FormatTraceInfo" via package "AXPAttahment" (perhaps you forgot to load "AXPAttahment" ?) at main::attachmentserrortext_Permission line 416."

or

"Error! It failed when executing the Hook of Open Action. The Hook is ACTION_ACCESS_CONTROL in ALMRequest
"AdEntity@0x03a5cd20.dbid=0.id='0'". The reason of Error is: Undefined subroutine &main::almrequest_AccessControl called"

At this time there is no solution or workaround to this issue. A support technote will be published when there is a resolution .

© Copyright IBM Corporation 2007, 2009
IBM Corporation
Software Group
Route 100
Somers, NY 10589
USA

IBM, the IBM logo, Rational, the Rational logo, Telelogic, the Telelogic logo and other IBM products and services are trademarks of the International Business Machines Corporation, in the United States, other countries or both.

Other company, product, or service names may be trademarks or service marks of others.

ALL RIGHTS RESERVED.

The Rational Software home page on the Internet can be found at **ibm.com/software/rational**

The IBM home page on the Internet can be found at **ibm.com**