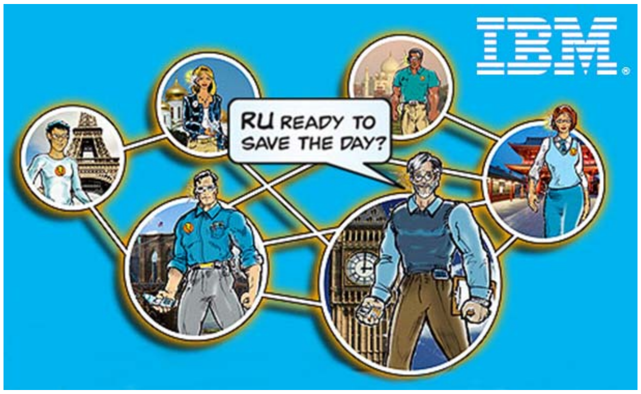
Introduction to Application Lifecycle Management on ClearQuest

Lab # 3: System-wide Settings



2008 Carolyn Pampino

Table of Contents

1 Overview 3

2 ALM Admin 4

3 Security Policy 5

4 Category & Category Type Label 7

5 Type & Type Labels 10

6 Labels 12

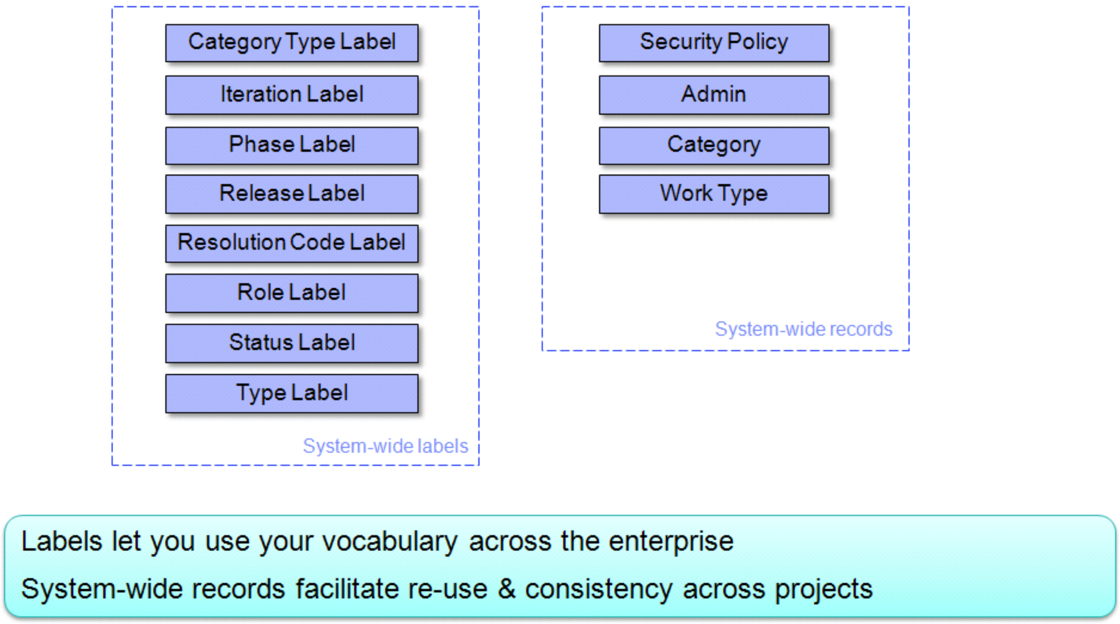
7 Using Role Labels as Templates 14

8 Congratulations! 17

Lab #3 System-wide Settings

1 Overview

The ALM solution comes with a set of system-wide settings that provide the power to modify the solution to fit your enterprise without impacting the underlying the schema. These settings allow you to use your vocabulary for defining objects within the system, and because they are system-wide, all users can share the same vocabulary.



1. Overview of system-wide records

2 ALM Admin

Anyone added to the ALMAdmin record will have the ability to modify all records in the ALM solution.

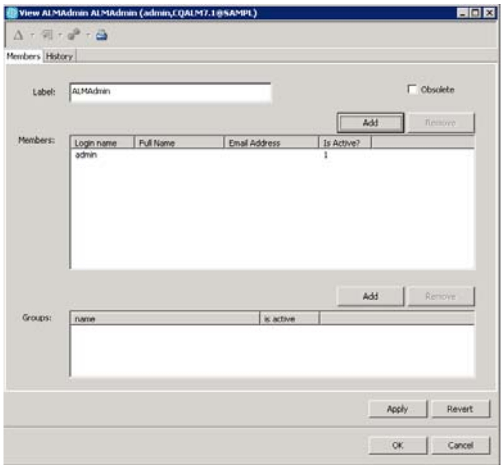
Add an ALMAdmin

___1. Login as ‘admin’ if you have not already.

___2. Run the Query for “**All ALMAdmins**”. (**Public Queries > ALM > General > All ALMAdmins**

___3. View the record in the results list.

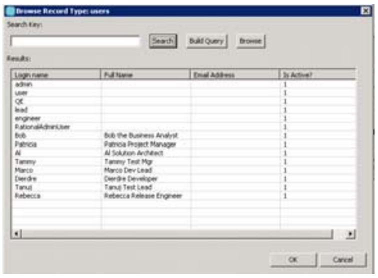
___4. Click **Modify**. 



___5. Add a member by clicking on the **Add** button next to the **Members**: list.

___a The Browse Record dialog opens.

___b Click the **Search** button.



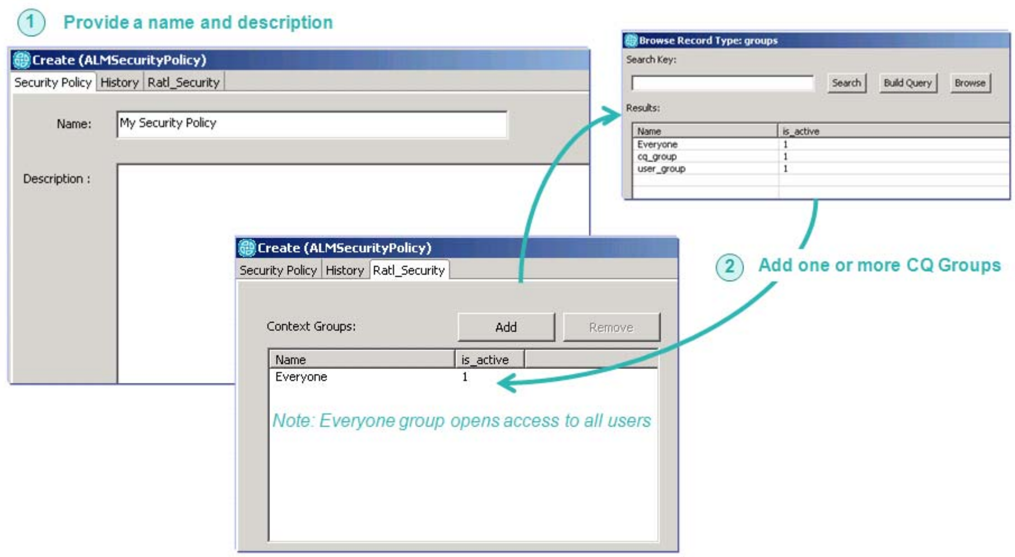
___c Select Patricia from the list of users.

___d Click **OK**.

Patricia now has access to modify the records in the ALM Package.

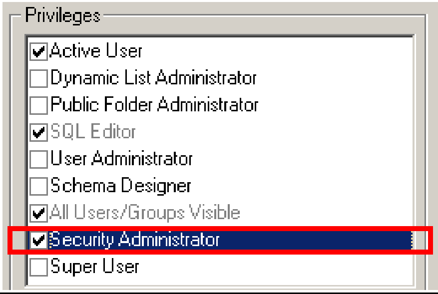
3 Security Policy

Security policies are defined by adding one or more ClearQuest groups to an **ALMSecurityPolicy** record. Once set, project managers are free to create new projects and choose the security policy needed for that project. Administrators need not get involved, unless a new CQ Group is needed. If a new policy is needed, it's simply a matter of defining a new security policy.



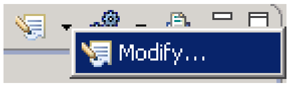
Modify a Security Policy

> *You need to be logged in as a user with CQ Security Administrator privileges to modify a Security Policy. The 'admin' user is a super user that has this privilege.*

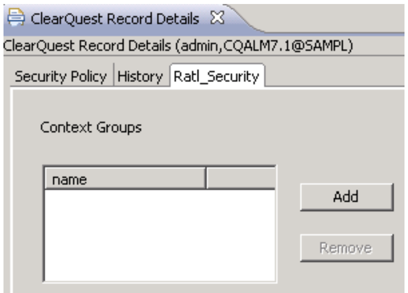


___1. Locate existing Security Policy records by running the " **Public Queries > ALM > General > All Security Policy**" query 

___a Select the security policy named '**Everyone**.'

___b Click **Modify**. 

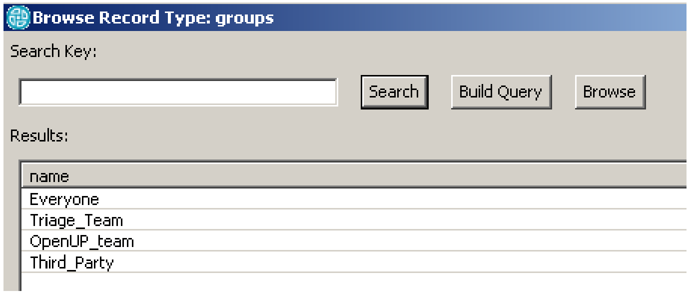
___c Click on the **Ratl_Security** tab



___d Click on the **Add** button next to the **Context Groups** field.

___e The Browse Record Type dialog appears.

___f Click **Search** button.



___g Select **ALL** groups **but** the 'Everyone' group

___h Click **OK**.

___i Note the new groups are added to the list of Context Groups.

___j Click the **Apply** button. 

You have now created a Security Policy that is open to all users of the system. This is convenient for situations where the security is lax or when you are creating a prototype.

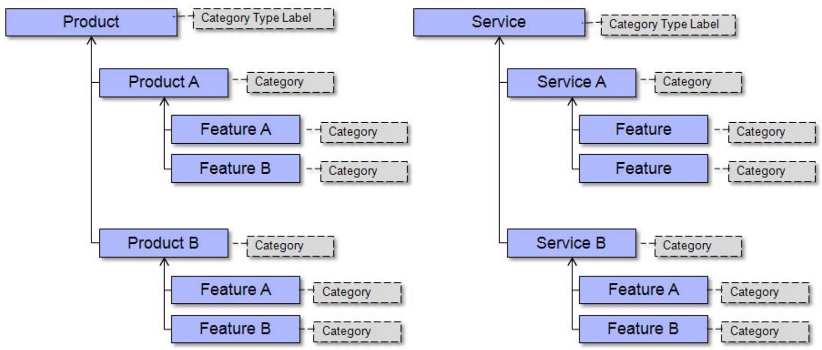
> *In CQ 7.1 the 'Everyone' group serves as catch-all group for all users in the system. In CQ 7.0.1 there's defect with using the 'everyone' group. The exercise above is a work-around for creating a security policy that is open to all users of the system.*

Explore the tabs and the fields of this record to become familiar with it. To create a Security Policy choose **New->ALMSecurityPolicy**.

4 Category & Category Type Label

While we discussed categories in the Project section, it is important to note that CategoryTypeLabels are available system wide.

Categories and Category Types allow you to model your classification system for projects. Categories can be hierarchical enabling you to decompose large systems into smaller more manageable units. This allows for re-use and consistent 'classification' across multiple projects.

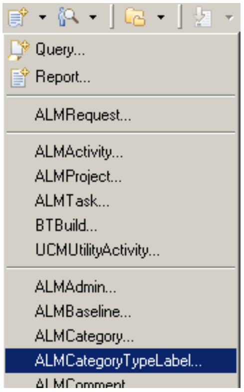


2. Two category trees: One for Products, one for Services

Create a Category Type Label

The sample database provides three category type labels: Component, Product and Solution. In this exercise you will create a new CategoryTypeLabel called "Service"

___1. File > New > CategoryTypeLabel



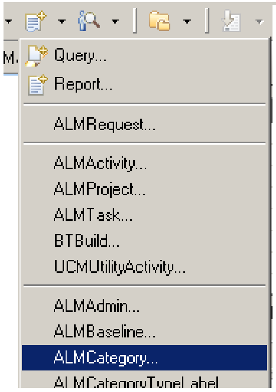
___2. Type `Service` in the **Name** field

___3. Optionally provide a **Description**. "`Use for organizing services`"

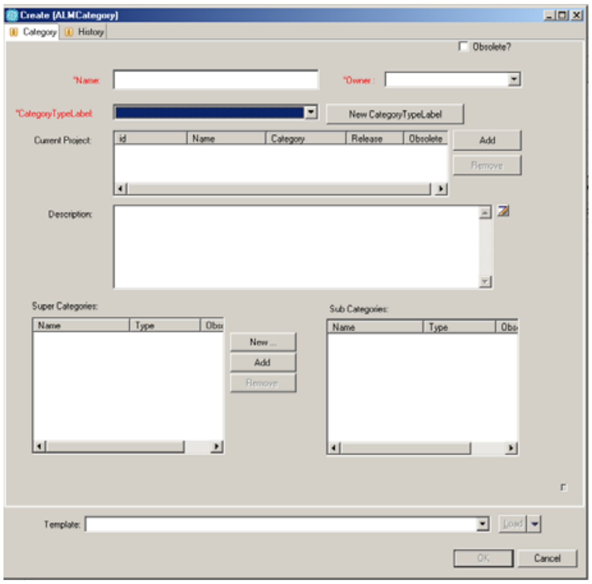
___4. Click **OK**

Create a Category

___1. File > New >ALMCategory



The Create ALMCategory window opens.



___2. Provide a **Name** for your category: `Shopping cart`.

___3. Set the **Owner** field to `admin`.

___4. Set **CategoryTypeLabel** to `Service`.

___5. Click on the **History** tab

___6. Set the **SecurityPolicy** to Everyone

___7. Click **OK**

Test your changes

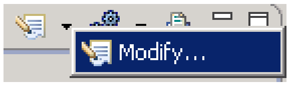
___1. New **ALMRequest**

___2. Click the **Category** drop-down. Your new Category should appear on the list.

Modify an Existing Category & Set the Current Project

___1. Locate existing Category records by running **Public Queries > ALM > General > All Category**.

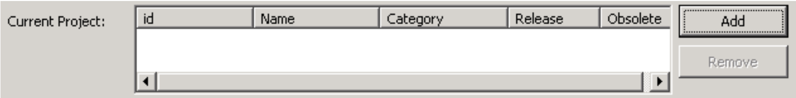
___2. View the category named 'Our Triage Product'.

___3. Click **Modify**. 

___4. Set the **Owner** (if it is not already set). Choose 'admin'.

___5. Set the **Current Project**

___a Click the **Add** button for the Current Project field.



___b The Browse Record type for ALMProject appears.

___c Click the **Search** button

___d Choose the project you created in the previous lab.

___e Click **OK**

___f Click **Apply**

___g From now on, when you create a Request record, and choose this category, the project field will default to the one you selected.

___6. Test your changes

___a New > ALMRequest

___b Set the **Category** to 'Our Triage Product'. Your project should be set by default.

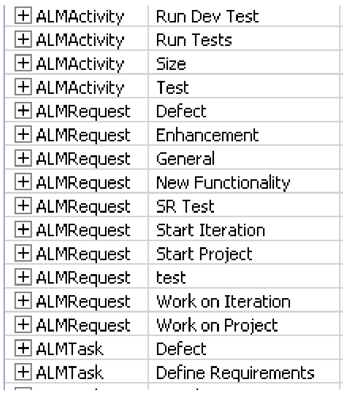
5 Type & Type Labels

Types are used to identify the ‘nature’ of the work. Types apply to Request, Task, and Activity and Build. You set the types system-wide. Project teams then configure which types to use by creating a Work Configuration. Some examples of types include but are not limited to: Enhancement, Defect, New Feature and so forth.

Inspect existing Types

___1. Locate existing Type records by running **Public Queries > ALM > General > All Type**

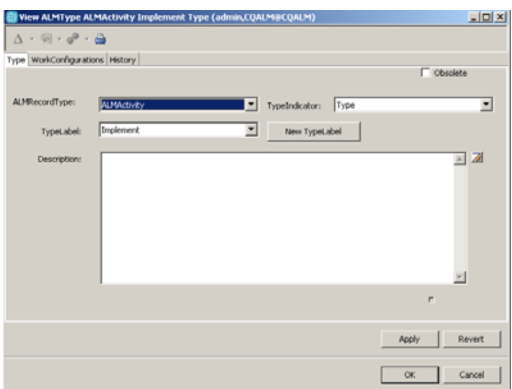
___2. Examine the result set, notice the Record Type, Type and TypeLabel fields. The records consist of ALMActivity, ALMRequest, and ALMTask. Each of these record types have one or more TypeIndicators and TypeLabels, such as “Run Tests” or “Define Requirements”.



ALMActivity	Run Dev Test
ALMActivity	Run Tests
ALMActivity	Size
ALMActivity	Test
ALMRequest	Defect
ALMRequest	Enhancement
ALMRequest	General
ALMRequest	New Functionality
ALMRequest	SR Test
ALMRequest	Start Iteration
ALMRequest	Start Project
ALMRequest	test
ALMRequest	Work on Iteration
ALMRequest	Work on Project
ALMTask	Defect
ALMTask	Define Requirements

A sample set of types is provided with the SAMPL database. These types are derived from OpenUP.

___3. View the **ALMActivity** with the **TypeIndicator** ‘Type’ and the **TypeLabel** ‘Implement’.



The record consists of choosing a RecordType, a TypeIndicator and TypeLabel. Description is optional.

The **WorkConfiguration** tab lists all WorkConfigurations that reference this type.

The **History** tab provides an audit trail for the record,

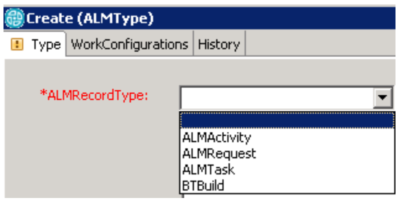
Create a Type

The team is under pressure to produce and is working around the clock. The project manager for this team is a bit unconventional and decides to create an activity to “Get Dinner” that can be assigned to someone when the project hits a critical point. Your job is to create that special Activity type.

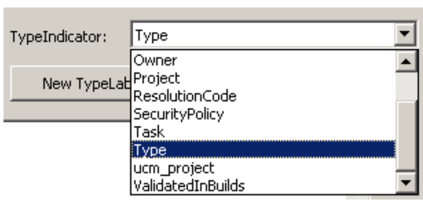
___1. Run the All Type query (**Public Queries->ALM->General-All Type**) . Inspect all types to confirm that one does not exist for “Get Dinner”.

___2. It doesn’t, so you need to create one. **File->New->ALMType** opens the ALMType dialog

___3. First set the **ALMRecordType** to `ALMActivity`

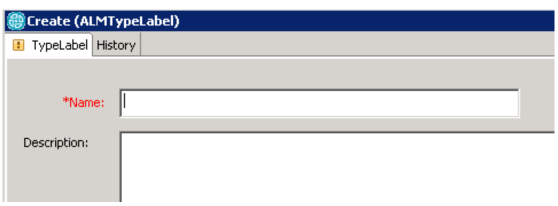


___4. Set the **TypeIndicator** to “Type”



___5. Next create a Type Label by clicking the **New TypeLabel** button

___a Provide a **Name** (`Get Dinner`) and optionally a description.



___b Click **OK**.

___6. Optionally provide a **Description** for the ALMType

___7. Click **OK** to submit the ALMType

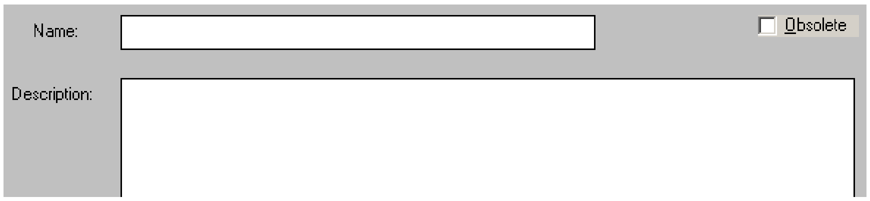
___8. Run the All Type query (**Public Queries->ALM->General-All Type**). Inspect all types to confirm that “Get Dinner” is now in the list.

6 Labels

About Labels

There are many instances where standardization across the enterprise is needed to help organize and govern projects, and rarely do we have tools to help enforce such consistency. With the introduction of "label" records, you now have the opportunity to define a set of names which appear in the user interface, most often in the form of drop-down lists on records. This section discusses each of the new label types and offers some suggestions for how you might use them.

Most 'Labels' simply consist of a Name and a Description.



3. User Interface for "Label" records

The following labels are set and shared across the database instance:

- Category Type Label - Categories provide a powerful means of classification projects, yet you are likely to find yourself wishing you had more than one 'tree' of categories. By defining more than one category type label, you can now create hierarchies of categories belonging to one category type or another. An example is to define category types such as: Solution, Product, Service, Re-Usable Component, Business Unit name and so forth.
- Phase and Iteration Labels - Many processes, including the Rational Unified Process recommend dividing projects into Phases, where each phase can have one or more iterations. Doing so helps to divide a project into more manageable units. For example, the Rational Unified Process suggests for Phase labels: Inception, Elaboration, Construction, and Transition. An iteration is a planned, time-boxed interval typically measured in weeks. Iterations focus the team on delivering incremental value to stakeholders in a predictable manner. By using Phase and Iteration labels you can ensure consistent adoption of terminology across your organization.
- Release Label - Releases are used to identify the 'version' of the software being developed. Use this field to identify a release label that will be used by others in the organization. For example, IBM has standardized on a four digit release numbering scheme such as: '7.1.0.0'.
- Resolution Code Label - When units of work are completed, a resolution code is set to provide a history and context for the type of resolution. For example, not all work is completed in a project. Sometimes there are 'duplicate' requests, or a reported problem cannot be reproduced or it works as designed. You have the opportunity to define a set of resolution codes to be used by your organization.
- Role Label - Role labels help to ensure a consistent use of roles across the organization such as Analyst, Architect, Project Manager, Developer, Tester are examples of role names.
- Status Label - The status label is used to report the status or 'health.' This appears on the Project, Phase, and Iteration records. Some examples include: Healthy, Suspect, Critical. Some organizations use color coding such as "Green" (for healthy), "Yellow" (for caution), and "Red" (for critical).
- Type Label- Requests, Tasks and Activities vary from organization to organization, therefore each of these records has a drop-down list called Type. The names that appear on that list come from the Type record. You start by defining the Type Label. This is simply the name of the type, such as "Passed." You then use the Type record to combine the Record type, a TypeIndicator and the Type Label.

Create a Label & Use it.

___1. **File->New-> Phase Label** (or any "...Label". This exercise uses "Phase Label")

___a Provide a **Name.**

___b Click **OK**.

___**2. File->New->Phase**

___3. In the dialog, click the **PhaseLabel** dropdown – Notice your new label now appears as a choice.

7 Using Role Labels as Templates

Role labels differ from the other labels. The Role Label dialog has an additional tab called “Approved Actions”. This allows you to define a base set of approved actions for the Role label. Any project that uses that role label will automatically inherit the approved actions. These can be over-ridden in the project if needed. However, this provides a convenience by having the RoleLabel act like a template for project teams.

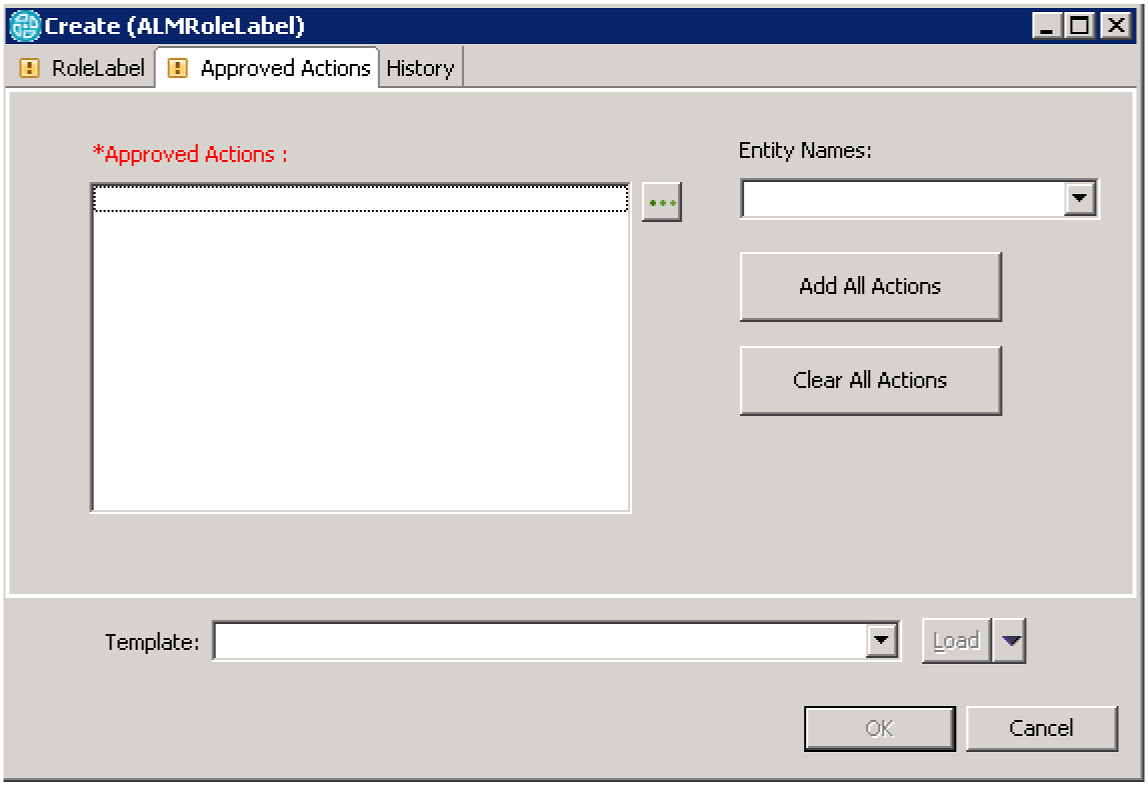
Create a Role Label

___**1. File > New > ALMRole Label**

___a Provide a **Name** for the role. To continue on the “Get Dinner” activity, create a roll such as: “`Dinner Duty`.”

___b Optionally provide a **Description**.

___2. Click the **Approved Actions** tab.



You can take a couple of approaches to setting the Approved Actions. One is to slowly build up the actions. Use this approach when specific role-based Approved Actions are an important aspect to your design.

The other approach is to click the **Add All Actions** button and remove any excess. There are many actions and it may be easy to overlook some. Use this approach when role-based action restriction is not as important.

This lab will teach you how to build up the actions.

___1. Select the small button to the right of the Approved Actions list

___2. The Editor appears.

___a Take a moment to look at the list of Approved Actions. You can set actions for Activity, Task, Request, Project, Baselines, Builds, Phases, Iterations, Role, Work Configuration

___b This is a simple role definition, where the person in charge of getting dinner is assigned an activity. The other actions do not apply. In this silly example, the person with 'Dinner Duty' can work with ALMActivity, but they cannot work with anything else.

___c Select `ALMActivity::Submit`

___d Click the **Add To** button

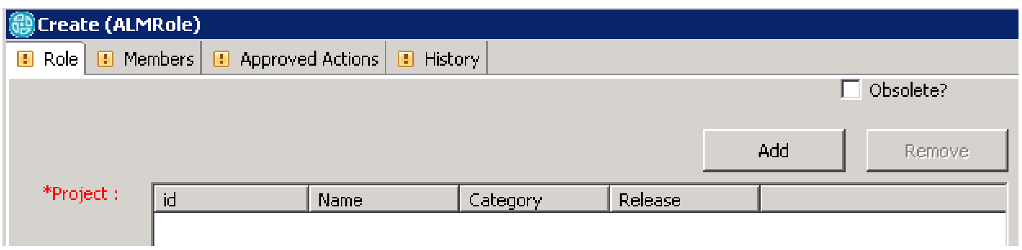
___e Repeat until the actions for the ALMActivity are added to the **Selections** list: Open, Activate, and Complete, QuestionOrComment.

___3. Click **OK**.

Create a Role that Uses your Role Label

___1. File->New->ALMRole

___2. Notice all tabs have required fields.



___3. First set the **Project** context that will use this role by clicking the **Add** button.

> *The Role record has a reference to the Project record. When working on the Project record, the list of roles is presented, however they cannot be changed from the project record. Always create the Role record and reference the Project.*

___f In the Browse Records dialog, click **Search**.

___g In the results, choose the *Agile* project

___h Click **OK**

___4. Note the **History** tab no longer has a required field. This is because the Security Policy is inherited from the Project selected.

___5. Set the **Role Label** by clicking on the drop-down list and choosing the role label you just created. (Dinner Duty) ___a Notice the **Approved Actions** Tab no longer has required fields.

___b Click on the **Approved Actions** Tab. The actions that you defined in the previous exercise are all listed. This is how the Role Label serves as a template to pre-populate the role for the project. Modifications to the Approved Actions at the Role record will only apply to the project this role references.

___6. Click on the **Members** tab.

___a Either User(s) or Group(s) must be added to perform this role. Setting one releases the 'required' nature of the other. Since anyone on the team can be assigned a "Get Dinner" activity, add all Groups to this membership.

___b On the **Members** tab of the Role dialog, set the **Primary** by choosing a name from the drop-down list. The Primary is used as the default by the system when a list of users is presented in the user interface.

___c All required fields should be completed at this point. Click the **OK** button to save your Role.

Create an ALMWorkConfiguration

___1. New ALMWorkConfiguration

___2. Bind your "Get Dinner" activity type to the *Agile* project.

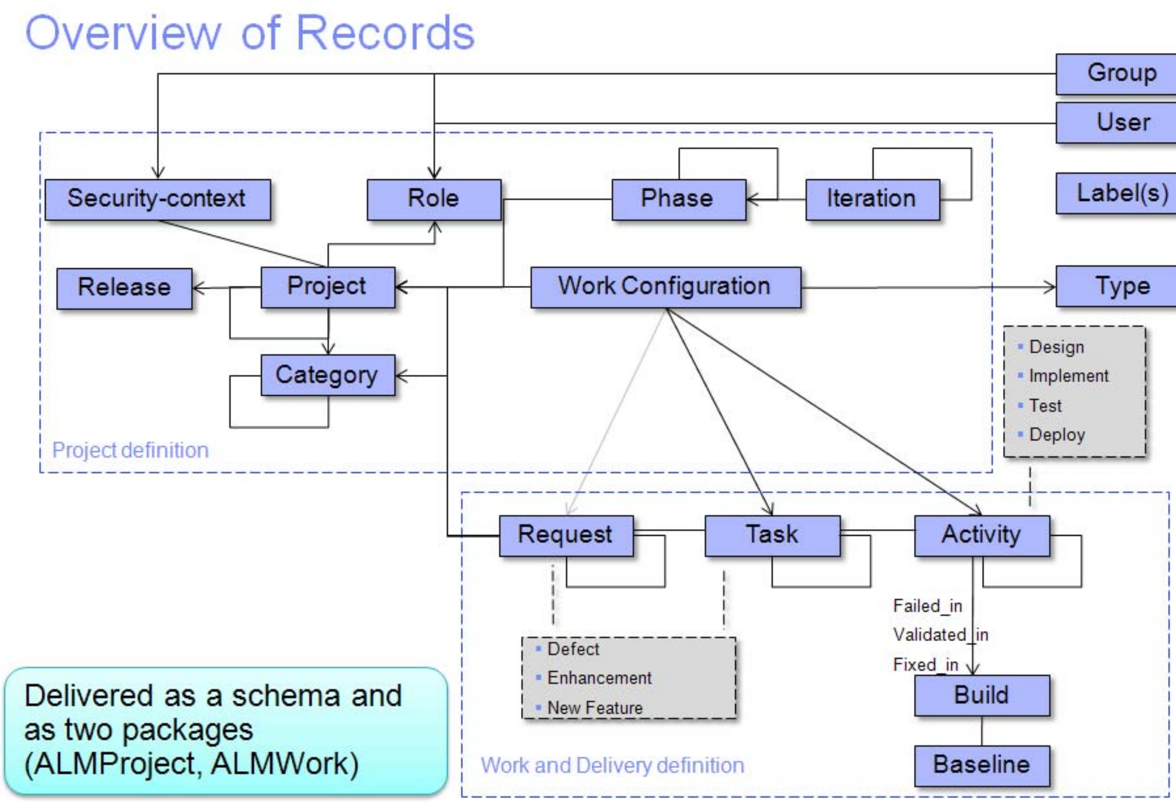
___3. Set the Role to the new role that you just created.

___4. Test your work by creating a new **ALMActivity** for the Agile project. The **Type** drop-down should include the *Get Dinner* choice.

8 Congratulations!

You have completed the first three labs for working with the ClearQuest ALM Schema.

A summary of the records used in this lab is provided in Figure 4.



4. Overview of records used in this session