



IBM Software Group | Rational software

**IBM® Rational® ClearCase® Unified Change Management (UCM)**

***Module 2: Working with UCM projects***

Rational software

@business on demand.

© 2007 IBM Corporation  
Updated July 18, 2007

The slide features a blue background with a circular pattern of dots. The IBM logo is in the top right corner. The title and subtitle are in bold black text. The Rational software logo is in a blue box. A horizontal bar with various icons is below the Rational logo. The @business on demand. logo is in the bottom right. Copyright and update information are at the very bottom.

This presentation covers working with UCM projects.

## Module objectives

- The following topics are covered in this section:
  - ▶ Creating a UCM project
  - ▶ Adding a component to an existing project
  - ▶ Extracting information about the project from the command line
  - ▶ Locking a project in order to stop development in that project
  
- Upon completion of this section, you will be able to:
  - ▶ Create a UCM project
  - ▶ Add a new component to an existing project



This module will help you figure out how to best work with UCM projects, including creating a UCM project, adding a component to an existing project, extracting information about the project from the command line, and locking a project in order to stop development in that project.

Upon completion of this section, you will be able to create a UCM project from the command line and from the GUI, and you will know how to add a new component to an existing project to make that component modifiable in that project.

## Working with UCM projects

- Create a UCM project Vob (cleartool mkvob “-ucmproject”)
- Create a project folder (cleartool mkfolder) (\*optional\*\*)
- Create a UCM project (cleartool mkproject)
- Join a UCM project (clearjoinproj “unix” / clearprojexp “win”)
- Understand project structure (cleartool chproject)
- Understand project policies (cleartool chproject)
- Add or remove a component in a project (cleartool rebase –baseline)
- Modifiable component list (cleartool describe –l <project>)
- Lock or obsolete a project (cleartool lock –obsolete <project>)
- Special project types
  - ▶ Bootstrap project (used to build composite baselines)
  - ▶ Main line project (cleartool mkproject –model SIMPLE)

3

Module 2: Working with UCM projects

© 2007 IBM Corporation

This module will cover the commands shown here; such as how to create a UCM project vob, joining a UCM project, modifying components, and working with special project types. Each command shown in parentheses will be covered in more detail in the coming slides. You can refer back to this slide for a quick snapshot of commands later.

(Pause 5 seconds)

## Create a UCM project vob

- To begin the creation of a UCM environment you must create a UCM project VOB (UCM PVOB)

```
Cleartool mkvob -tag \project_vob -ucmproject  
\svr\vobstg\project_vob.vbs
```

- ▶ The project vob will be the vob containing all metadata for future projects within the specified Pvob
  - ▶ The project vob RootFolder is visible in the project explorer
  - ▶ You will create your projects beneath the PVOB RootFolder
  - ▶ Project vobs are special vobs. You can not convert a base vob to a pvob.
- For a step-by-step guide to creating a UCM project VOB, refer to the *RCCv7\_UCM\_Doc1\_PVOB.pdf* document



To begin the creation of a UCM environment you must create a UCM project VOB. A PVOB contains UCM objects such as projects, streams, activities, and change sets that are used to manage the development artifacts in components.

From the command line, run command shown here.

Some important items to note about the project vob:

The project vob will be the vob containing all metadata for future projects within the specified Pvob. The project vob RootFolder is visible in the project explorer.

You will create your projects beneath the PVOB RootFolder.

Project vobs are special vobs, and you can not convert a base vob to a pvob.

## Create a project folder

- In the project explorer you have the *option* to create a folder to organize all of your projects
  - ▶ Note: You must run this command from a view
  - ▶ Note: You must be in the root of the pvob

### Cleartool mkfolder –in RootFolder training\_project

- **-in** is used to specify a Folder to create the project in
- **RootFolder** is a predefined Folder name for the Pvob root folder
- **training\_project** is the name of the new Folder created

5

Module 2: Working with UCM projects

© 2007 IBM Corporation

In the project explorer you have the *option* to create a folder to organize all of your projects. This is completely optional and is a matter of personal preference on how to manage your projects. Note that this command must be run from the command line interface while in any view and in the root directory of the pvob.

The command is shown here.

The “dash in” switch is used to specify a folder to create the project in, RootFolder is a predefined folder name for the Pvob root folder, and training\_project is the name of the new folder created.

You can now create projects within the newly created folder

## Create a project folder (cont.)

- Example:

```
M:\view\project_vob>cleartool mkfolder -in RootFolder training_project
```

- Result:

```
Created folder "training_project"
```

- You will now be able to create projects within the newly created folder



This example creates a new folder. The resulting folder “training project” is shown here. You will now be able to create projects within the newly created folder.

## Create a UCM project

- Project names can not be the same as existing projects, folders or components
- Project names should be short if possible
- Creating a project from the GUI will automatically create an integration stream
- Creating the project from the command line interface requires that you create the integration stream separately

For an instructional video on how to create a UCM project from the GUI, see:

<http://www.ibm.com/support/docview.wss?uid=swg21247816&aid=1>



Project names in a UCM project can not be the same as existing projects, folders or components. They should be short, if possible because you will have to remember them for other tasks.

Creating a project from the GUI will automatically create an integration stream, and creating the project from the command line interface requires that you create the integration stream separately.

To view a short demonstration on how to create a UCM project from the GUI, pause this presentation and follow the link shown here.

(Pause 5 seconds)

## Create a UCM project (cont.)

- Create a project command

```
cleartool mkproject -in RootFolder project:project_name@\project_vob
```

- Other options when running cleartool mkproject:

**-modcomp** = specify modifiable components in the project

**-policy** or **-npolicy** = enable or disable policy for project  
(disabled by default)

**-crmenable** = specify CQ user db to enable CQ integration

**-model SIMPLE** (single stream) or **DEFAULT** (parallel development)

To Create a project you must use the command shown here.

You also have other switch options when running cleartool mkproject.

The -modcomp switch specifies modifiable components in the project, the -policy and -npolicy switches enables and disable policy for a project – policy is disabled by default.

The -crmenable switch specifies CQ user db to enable CQ integration, and the -model SIMPLE specifies and single stream while a -model DEFAULT specifies parallel development.

## Join a UCM project

- From the command line interface, joining a project can only be done if you know the name of the project.
- Joining a project from the command line is done by:
  - ▶ Creating a new stream or
  - ▶ Creating a view into an existing stream:
    - 1) cleartool mkstream -in project:project\_name@\project\_vob  
stream:new\_stream@\project\_vob
    - 2) cleartool mkview -stream stream:existing\_stream@\project\_vob
- Instructions in ***RCCv7\_UCM\_Doc2\_mkprojwiz.pdf***

How to join a UCM Project from the GUI:

<http://www-1.ibm.com/support/docview.wss?uid=swg21247816>



You can join a UCM project from both the GUI and from the command line interface. You can get step-by-step instructions in the .pdf file listed here.

From the command line interface, joining a project can only be done if you know the name of the project. Joining a project from the command line interface is done by either creating a new stream or by creating a view into an existing stream using the commands shown here.

To view a short demonstration on how to join a UCM project from the GUI, pause this presentation and follow the link in this slide.

## UCM project structure

- UCM projects can only have ONE integration stream
  - ▶ The integration stream is the highest level stream of all streams in the project hierarchy
  - ▶ The integration stream should be used to generate the projects configuration
  - ▶ The integration stream's recommended baseline list will automatically seed child streams configurations
  - ▶ Child streams will automatically have the integration stream as their default deliver target for development purposes

10

Module 2: Working with UCM projects

© 2007 IBM Corporation

Because UCM Projects can only have ONE Integration stream, you should be aware of these things:

- The integration stream is the highest level stream of all streams in the project hierarchy
- The integration stream should be used to generate the projects configuration (or, determine components to be used in the project)
- The integration stream's recommended baseline list will automatically seed child streams configurations when they are created. When joining the project, a developer's stream will use the recommended baseline list to generate its configuration.
- The child streams will automatically have the integration stream as their default deliver target for development purposes – although you can have as many child streams as you need.

## UCM project policies

- Three basic mechanisms for defining policies:
  - ▶ UCM project policies and properties
    - How streams and projects interact
    - The integration and configuration of ClearQuest®
    - Access control over projects and streams
  - ▶ ClearCase triggers
    - Customer-specific customization of the process model
    - Set pre- and post-operation triggers
  - ▶ Project practices
    - Policy definitions and triggers enforce process rules
    - Set or unset formal and informal rules



UCM policies define configuration and operational rules for a project.

There are three basic mechanisms for defining policies: UCM project policies and properties, ClearCase triggers, and project practices.

UCM project policies and properties provide a set of tunable project rules. For instance, how streams and projects interact, integration and configuration of ClearQuest, and access control over projects and streams. These policies and properties tuning can be done from the Project Explorer interface.

ClearCase triggers allow customer-specific customization of the process model. Pre- and post-operation triggers can also be attached to many ClearCase and UCM operations. Just because you can does not mean you should. Make sure you have good business reasons for adding triggers to your system. It is not uncommon to find an overuse of triggers, thus causing performance degradation.

Finally, project practices policy definitions and triggers enforce process rules in your environment. Additionally there will be set of less formal rules which you define for your project team and document in your plan. For example, you can synchronize your development stream with the project integration stream at least weekly, or deliver activities as soon as they have passed unit testing, or check in work in progress on a nightly basis.

## UCM project policies (cont.)

- Project policies
  - ▶ Policies can be set at the project level only or per stream
  - ▶ You can enable a policy for the entire project or you can enable it per stream
  - ▶ Policies are used for access restrictions, allow or deny deliver options, or set default view types when developers join the project
- Policies can be enabled or disabled from the command line interface:
  - ▶ `cleartool chproject -amodcomp` = make component modifiable
  - ▶ `cleartool chproject -policy <policy keyword>` = enables a policy
  - ▶ `cleartool chproject -npolicy <policy keyword>` = disables a policy
- For a step-by-step guide to creating a UCM project VOB, refer to exercise 1 in the *RCCv7\_UCM\_Doc3\_projectPolicies.pdf* document

As you just learned, UCM project policies and properties provide a set of tunable project rules. Policies can be set at the project level only or per stream. You can also enable a policy for the entire project or you can enable it *per stream*. Policies are sometimes used for access restrictions; allow or deny deliver options, or set default view types when developers join the project.

Policies can be enabled or disabled from the command line interface using the *cleartool chproject* command. Some examples are shown in this slide.

For a step-by-step guide to creating a UCM project VOB, refer to the .pdf document listed here.

## Add or remove a component in a project

- Adding a component to a project from the GUI
  - ▶ For a step-by-step guide to creating a UCM project VOB, refer to the *RCCv7\_UCM\_Doc4\_Addcomp.pdf* document

For an Instructional video on how to add a project from the GUI, see:

<http://www-1.ibm.com/support/docview.wss?uid=swg21147087&aid=1>



To add a component to a project you will need to *rebase* the project's integration stream with the appropriate component foundation baseline. Once the component has been added to the integration stream's foundation, it will be available to the project.

To add a component to a project from the GUI, it is best to refer to the step-by-step document listed here. You can also view an instructional video on adding a new component to an existing project by pausing this presentation and following the link provided here.

## Add or remove a component in a project (cont.)

- Adding a component to a project from the command line interface
  - ▶ The name of the integration stream
  - ▶ The name of the desired baseline of the component being added
  - ▶ Have a view into the integration stream



You can also add a component to a project from the command line interface. First, you will need to know the name of the integration stream, and you will need to know the name of the baseline of the component being added, and last, you will need to have a view into the integration stream.

## Add or remove a component in a project (cont.)

- Removing a component from a project
  - ▶ Need to know the same information as adding a component
  - ▶ Component must have no foundation baselines in any of the projects streams
  - ▶ In most cases it is very difficult and at times not possible to remove the component from the projects streams
- To remove a component from a stream
  - ▶ No changes made to any foundation baselines in the projects streams
  - ▶ Must not be in use or modified on a child stream
- If the conditions above are satisfied the following command can be used to remove the component from all streams in the project:

```
cleartool rebase -dbaseline baseline:baseline_name@\project_vob -  
view view_name -stream stream:stream_name@\project_vob
```

To remove a component you will need to know the same information as adding a component. To remove a component from a project, it must have no foundation baselines in any of the projects streams, and be aware that in most cases it is very difficult and at times not possible to remove the component from the projects streams.

To remove a component from a stream, it must have no changes made to any foundation baselines in the projects streams, and it must not be in use or modified on a child stream.

If the conditions above are satisfied the following command can be used to remove the component from all streams in the project. Use the cleartool rebase dash dbaseline command shown here. The dash dbaseline specifies one or more baselines to remove from the stream's configuration. Files in those baseline's components are subsequently no longer visible or modifiable in the stream.

## Modifiable component list

- A component can only be made modifiable within a project
- When a component is added to a PVOB it is read-only by default
- When creating a project you can specify the modifiable component list
- The project's modifiable component list determines what components can be modified by its streams
- Once a component is made modifiable, it can never be changed back to non-modifiable within a single project
- You can view the modifiable component list in the project explorer > project policy > component tab or from the command line interface command:  
`cleartool describe -l project:project_name@\project_vob`
- For a step-by-step guide to creating a UCM project VOB, refer to the *RCCv7\_UCM\_Doc4\_Addcomp.pdf* document

When adding a component, understand that it can only be made modifiable within a project. When a component is added to a PVOB it is read-only by default. When creating a project you can specify the modifiable component list. The project's modifiable component list determines what components can be modified by its streams.

Keep in mind that once a component is made modifiable, it can never be changed back to non-modifiable within a single project. You can view the modifiable component list in the project explorer > project policy > component tab or from the command line interface with the command shown here:

For a step-by-step guide to creating a UCM project VOB, refer to the .pdf document listed here.

## Lock or obsolete a project

- *Locking* a project restricts the project from further development
- *Obsolete* a project to make the project locked and invisible from the project explorer and command line interface
- Example:

```
cleartool lock -obsolete project:project_name@\project_vob
```

Locking a project restricts the project from further development. If a project is at the end of its life cycle you can lock the project from further production.

Obsolete a project to make the project locked and invisible from the project explorer and command line interface. This is the recommended solution for removing a project from future production. Removing a project is a very cumbersome process which can cause further damage to future development, so it is often better that the project be locked obsolete for future development to in order to retire the project. Use the command shown here to obsolete the project.

## Special project types

- Two types of projects which can be used to enhance UCM development:
  - ▶ Bootstrap projects
    - Used to build and implement composite baselines
  - ▶ Main line projects
    - Seed all other projects with common foundations
    - Allows multiple projects to stem from a single main line project for simple hierarchical structure
    - Helps keep naming conventions short for ease of use and clarity
- Creating a bootstrap project:
  - ▶ You can use a single stream or parallel development project
  - ▶ For a step-by-step guide to creating a bootstrap project, refer to the *rccv7\_ucm\_doc5\_specialprojects.Pdf* document

Contributors: Marcus Matic, Alex Grillakis, and Will Frontiero

There are two types of projects which can be used to enhance UCM development: bootstrap and mainline projects.

The bootstrap project is used to build and implement composite baselines. The mainline project is used to seed all other projects with common foundations; mainline projects will allow multiple projects to stem from a single mainline project for simple hierarchical structure. The project also helps keep naming conventions short for ease of use and clarity.

When creating a bootstrap project, you can use a single stream or parallel development project.

For a step-by-step guide to creating a bootstrap project, refer to the .pdf document listed here.

## 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 e-mail feedback:

[mailto:iea@us.ibm.com?subject= Feedback about  
RCCv7 UCM Module2 WorkingWithUCMProjects.ppt](mailto:iea@us.ibm.com?subject=Feedback%20about%20RCCv7%20UCM%20Module2%20WorkingWithUCMProjects.ppt)



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

## Trademarks, copyrights, and disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

ClearCase ClearQuest IBM Rational

Rational is a trademark of International Business Machines Corporation and Rational Software Corporation in the United States, Other Countries, or both.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2007. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.