



IBM Software Group | Rational software

# IBM Rational ClearCase UCM V 7

## Understanding nested UCM project VOBs

Rational software



@business on demand.

© 2009 IBM Corporation

Updated June 11, 2009

This module will cover the Nested UCM Project VOBs in IBM Rational® ClearCase® versions 7.0 and higher.

## Module objectives

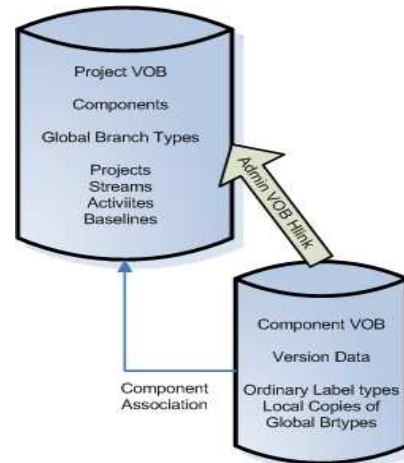
- The following topics are covered in this module:
  - ▶ Standard single UCM project VOB development
  - ▶ Creating a nested project VOB structure
    - Using an existing project VOB as an administrative VOB for child project VOB.
    - Using a Common administrative VOB for Multiple Project VOBs
  - ▶ Using ucutil link\_pvob
  - ▶ Considerations for multiSite environments
- When you complete this module, you will:
  - ▶ Understand how to safely configure a nested Pvob structure
  - ▶ Understand what nested structure is the best fit for your organization



This course covers four topics. Standard single UCM project VOB development, how to create a nested project VOB structure, how to the use ucutil link\_pvob and what considerations you need to keep in mind for a multisite environment. When you complete this module, you will have a good understanding of how to safely configure a nested Pvob structure and what nested structure is the best fit for your organization.

## Standard single UCM project VOB development

- Standard UCM development requires a connection between a project VOB and a component VOB



The Component VOB can contain one or more Component root storage locations depending on its configuration. Each VOB contains a variety of hyperlinks to ensure data is protected and catalogued in the necessary locations. Creating data in a VOB that is a member of an administrative VOB structure implies shared data. The shared data is created in either the administrative VOB or the Child VOB depending on the data type.

## Standard single UCM project VOB development (continued)

- Standard UCM Development implies that file and directory element data are generated in the component VOB
- Each branch is associated with a UCM stream housed in the project VOB
- Projects are created in the project VOB and are used to organize and apply policy to a group of Streams
- The UCM Baseline is an object created in the project VOB. It is associated with an ordinary label type generated in the component VOB



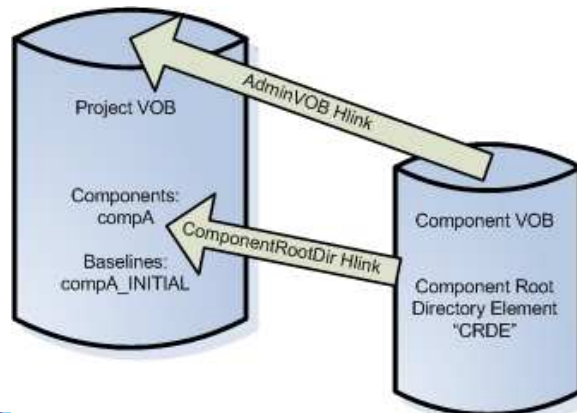
This slide reviews the details of standard UCM project VOB development. Each version generated in the Component VOB is associated with a UCM activity. The activity is housed in the project VOB. Every version generated must be associated with a ClearCase branch type. The Project VOB houses a global copy of every branch type needed during development. When a version is created, a local copy of the global branch is generated in the Component VOB. Review the diagram in slide three showing the location of all the data being described.

## Standard single UCM project VOB development (continued)

- **Component creation example:**

- ▶ **Example:**

```
M:\UCM_TRAIN\comp_root>cleartool mkcomp -root .@@ compA@\pvobA
```



5

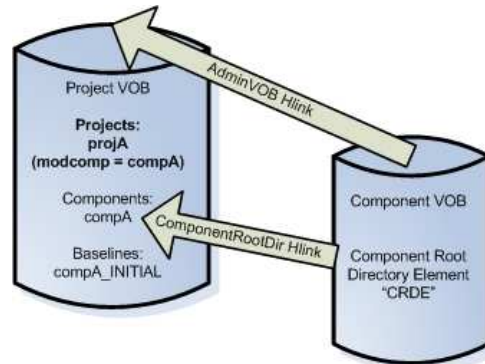
The diagram in this slide is demonstrating component creation. The administrative VOB hlink is created from the component VOB to the project VOB. ComponentRootDir hlink is then created from the component root directory element to the component object. An INITIAL baseline is also generated. This baseline will represent the Main\0 version of the component root directory element (the start of component development).

## Standard single UCM project VOB development (continued)

- **Project creation example:**

- ▶ **Example:**

```
M:\UCM_TRAIN\comp_root>cleartool mkproj -nc -modcomp compA@\pvobA -in RootFolder  
projA@\pvobA
```



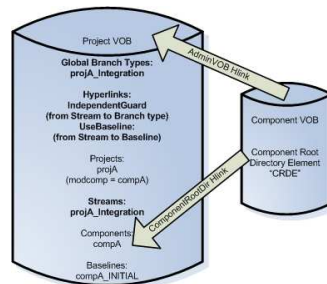
The diagram in this slide is demonstrating project creation. The project VOB is the only VOB that contains any knowledge of the project (No changes are made in the Component VOB).

## Standard single UCM project VOB development (continued)

- Stream creation example:

- Example:

```
M:\UCM_TRAIN\comp_root>cleartool mkstream -integration -in projA@\pvobA -nc -baseline
compA_INITIAL@\pvobA projA_Integration@\pvobA
```



The diagram in this slide is demonstrating stream creation. Stream creation will generate a stream object and a global branch type in the project VOB. Two hyperlink objects will also be created. A UseBaseline hlink joining the stream to a selected baseline and an IndependentGuard hlink joining the stream to the global branch type. (Notice, no change is made to the Component VOB)

## Standard single UCM project VOB development (continued)

### ▪ Create version example

#### ▶ Example:

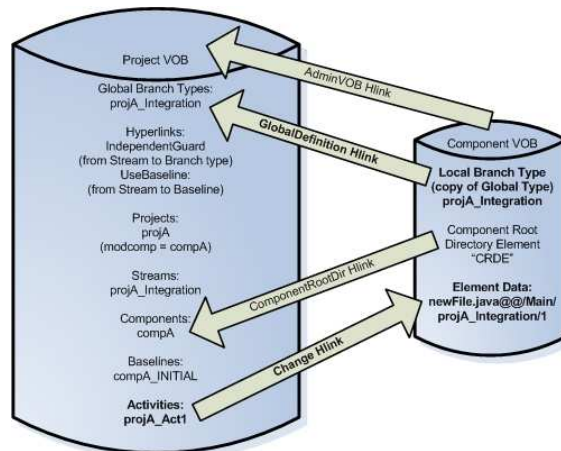
```
M:\UCM_TRAIN\comp_root>cleartool mkview -tag admin_projA_Int -stream
stream:projA_Integration@\pvobA -stgloc -auto
M:\admin_projA_Int\comp_root>cleartool mkact projA_Act1
M:\admin_projA_Int\comp_root>cleartool checkout .
M:\admin_projA_Int\comp_root>cleartool mkelem newfile.java
M:\admin_projA_Int\comp_root>cleartool checkin -ident -cact
```

The example in this slide is demonstrating version creation (check-out/check-in). The diagram can be viewed on the next slide. Creating a version on a UCM stream will require a UCM activity. The activity object is created in the project VOB and associated to the stream. The component VOB will generate a local copy of the global branch type associated with the stream. The version will then be associated with the UCM activity set during check-in of the new version.



## Standard single UCM project VOB development (continued)

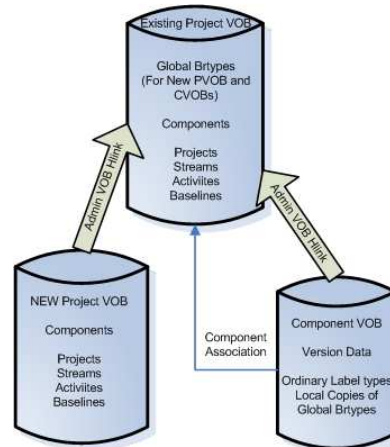
- Create version example



The diagram in this slide is demonstrating version creation (check-out/check-in).

## Creating a nested project VOB structure

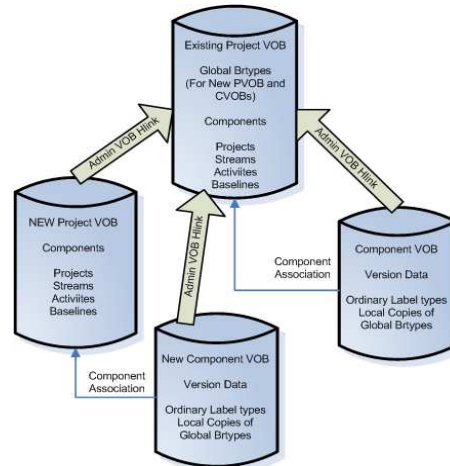
- Using an existing project VOB as an administrative VOB in a nested project VOB structure
  - ▶ Diagram demonstrating the metadata configuration



This diagram demonstrates a few things. How to use an existing project VOB as an administrative VOB for a separate UCM project VOB structure. It also shows project VOB and component VOB association and project VOB and component association. It is assumed that the new project VOB contains no component VOB association and that the new project VOB has no existing project structures containing streams. If the assumptions are incorrect, the “ucmutil” tool is needed to create the association between the project VOBs. The use of the ucmutil tool is discussed later in this module.

## Creating a nested project VOB structure

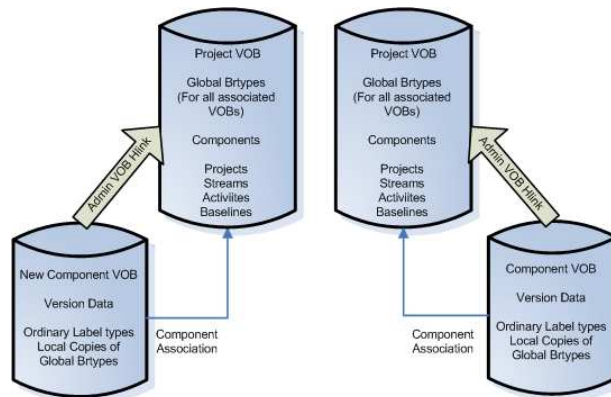
- **Using an existing project VOB as an administrative VOB in a nested project VOB structure.**
  - ▶ Diagram demonstrating the metadata configuration:



This diagram demonstrates utilizing an existing project VOB as an administrative VOB for a separate UCM project VOB structure. It shows a project VOB and component VOB association (For Existing and New VOBs). It also shows a project VOB and component association (For Existing and New VOBs). It is assumed that the new project VOB was associated with the existing project VOB before component creation and that the new project VOB was associated with the Existing project VOB before project creation. If the assumptions are incorrect, the “ucmutil” tool is needed to create the association between the project VOBs. Again, the use of the ucmutil tool will be discussed later in this module.

## Creating a nested Project VOB structure

- **Using a common administrative VOB for multiple project VOBs**
  - ▶ Diagram demonstrating the metadata configuration for the separate project VOB structures: (before creating the nested structure)

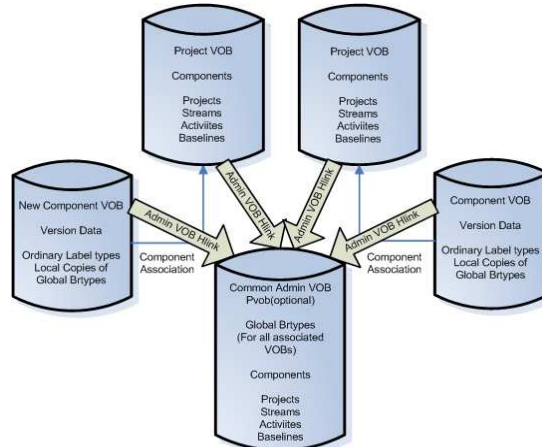


This diagram demonstrates independent project VOB structures before the association with a common administrative VOB. It is assumed that the child project VOBs contain the existing UCM projects and streams and that the child project VOBs were associated with one or more UCM component VOBs. In this configuration the “ucmutil” tool should be the only option to associate a child project VOB with a higher level administrative VOB.

## Creating a nested project VOB structure

- Using a common administrative VOB for multiple project VOBs

- Diagram demonstrating the metadata configuration after association to a common administrator VOB (Pvob optional)



13

This diagram demonstrates utilizing a new or existing administrative VOB (Pvob) as a common administrative VOB to child UCM project VOBs. It is assumed that the child project VOBs contain existing UCM projects and streams and the child project VOBs were associated with one or more UCM component VOBs. The “ucmutil” tool was used to associate each child project VOB to the common administrative VOB. In this configuration the “ucmutil” tool should be the only option to associate a child project VOB with a higher level administrative VOB.

## Using “ucmutil” to link project VOB structures

- **When do you need the ucmutil tool**
  - ▶ Child Project VOBs contain existing UCM Projects and Streams
  - ▶ Child Project VOBs are associated with one or more UCM Component VOBs
- **What is the purpose of the ucmutil tool**
  - ▶ Ucmutil has a sub command called “link\_povb”
  - ▶ “link\_povb” is used to associate Project VOBs
- **What data is being affected by the ucmutil tool**
  - ▶ All data is affected by the tool if used improperly
  - ▶ The tool will Migrate Global Brtypes from a Child Pvob to the new Target administrator VOB
  - ▶ The tool will regenerate Stream association to the Migrated Brtypes Types
  - ▶ The tool will regenerate Component VOB administrative VOB association to the new Target administrative VOB.

The ucmutil tool is found in Windows® at c:\program files\rational\clearcase\etc\utils\ucmutil.exe . In UNIX® this is found at /opt/rational/clearcase/etc/utils. The ucmutil tool is required when migrating existing project metadata. It requires a source VOB and a target VOB. The source VOB cannot be associated to a current administrative VOB. The source VOB, target VOB and preferably all associated Component VOBs must be locked –user (for the user\_running\_ucmutil).

## Using “ucmutil” to link project VOB structures (continued)

### ▪ Next Steps

- ▶ Lock each vob associated with the source and target VOBs with the `-nuser` switch
- ▶ Run `ucmutil link_pvob -verbose` and redirect the output to a log file for historical purposes. (It can also be helpful for troubleshooting)
- ▶ Ensure all branch types listed in preview mode log file are mastered locally at the site running the operation.
- ▶ Once the check is complete, you can run the `Link_pvob` subcommand `live` to associate the VOBs

Create the administrative VOB `hlink` from the source VOB to the target administrative VOB. Push all existing global branch type definitions in the existing project VOB to the target administrative VOB. Re-associate all existing streams with the global branch type definitions moved to the target administrative VOB. Redirect the administrative VOB link of any associated component VOBs to the new target administrative VOB. This is a result of running the “`ucmutil link_pvob`” utility. You should also note that the component association remains with the original project VOB.

## Considerations for multisite environments

- If branches are mastered at other sites, you will need to push mastership of all branches to one site before running the utility.
- The tool will fail to modify branch types that are not mastered locally.
- The tool can only be run once. If it fails to modify some branch types it can not be resolved by rerunning the tool.
- Rerunning the utility will fail since the administrative VOB hlink will already be in place. You will need to manually fix each failed branch type.

If the VOBs are multisited, IBM Rational support has a procedure that can be used to resolve branch types that need manual migration.



## Summary

- Nested Project VOB structures can take many different shapes and sizes.
- They enable UCM environments to be scalable and specialized.
- They can reduce the amount of data stored in any one repository.
- Each environment should consider the best fit for their needs
- The information supplied above can be further understood by reading the published whitepaper on this topic:



This module provided you with an understanding of nested UCM project VOBs. Now you are familiar with how to safely configure a nested Pvob structure and what nested structure is the best fit for your organization.

## Additional resources

- **Additional resources on developerWorks®**  
<http://www-01.ibm.com/support/docview.wss?uid=swg27015347>
- **Additional resources on ibm.com**  
<http://www-01.ibm.com/support/docview.wss?uid=swg27014131>



Additional resources are located on ibm.com and developerWorks.

## 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\\_NestedPvoblEAModule.ppt](mailto:iea@us.ibm.com?subject=Feedback_about_NestedPvoblEAModule.ppt)

This module is also available in PDF format at: [../NestledPvoblEAModule.pdf](http://NestledPvoblEAModule.pdf)



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



## Trademarks, copyrights, and disclaimers

IBM, the IBM logo, ibm.com, and the following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

ClearCase developerWorks Rational

If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. 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>

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

Windows, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

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

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.

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 NONINFRINGEMENT. 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 2009. 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.

