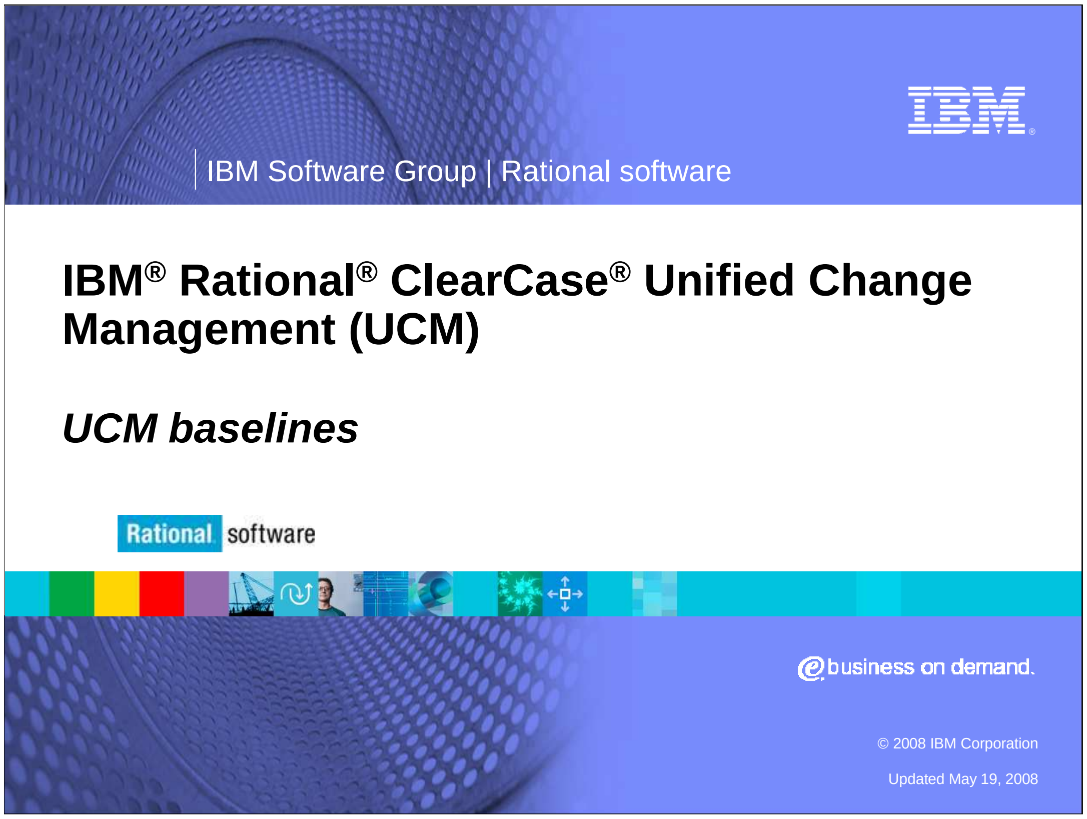


IBM® Rational® ClearCase® Unified Change Management (UCM)

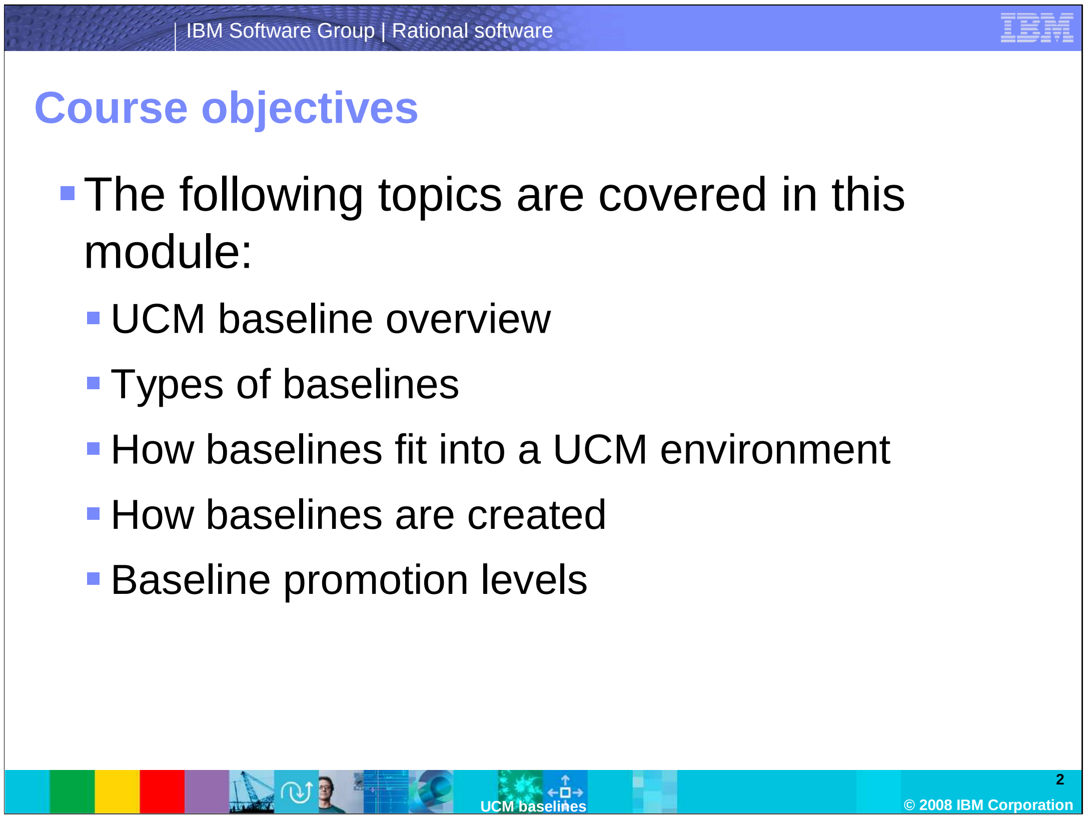
UCM baselines

Rational software

@business on demand.

Updated May 19, 2008

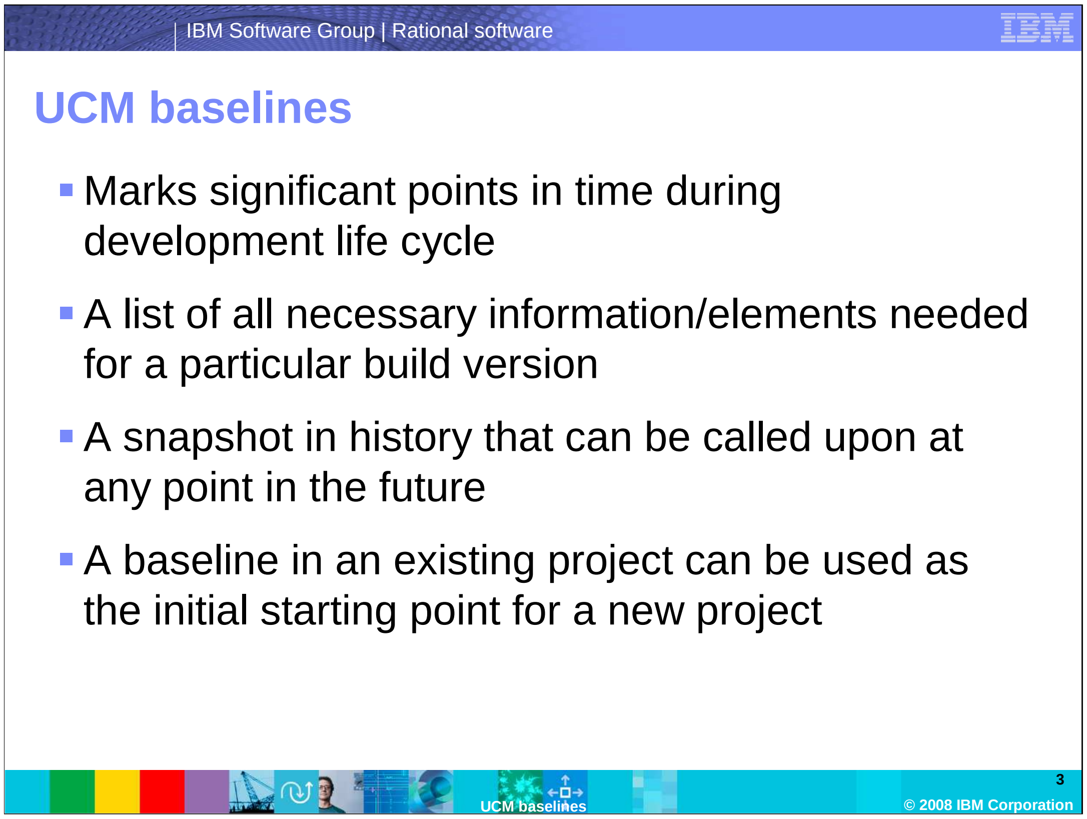
This module will cover UCM baseline basics for IBM Rational ClearCase Unified Change Management.



Course objectives

- The following topics are covered in this module:
  - UCM baseline overview
  - Types of baselines
  - How baselines fit into a UCM environment
  - How baselines are created
  - Baseline promotion levels

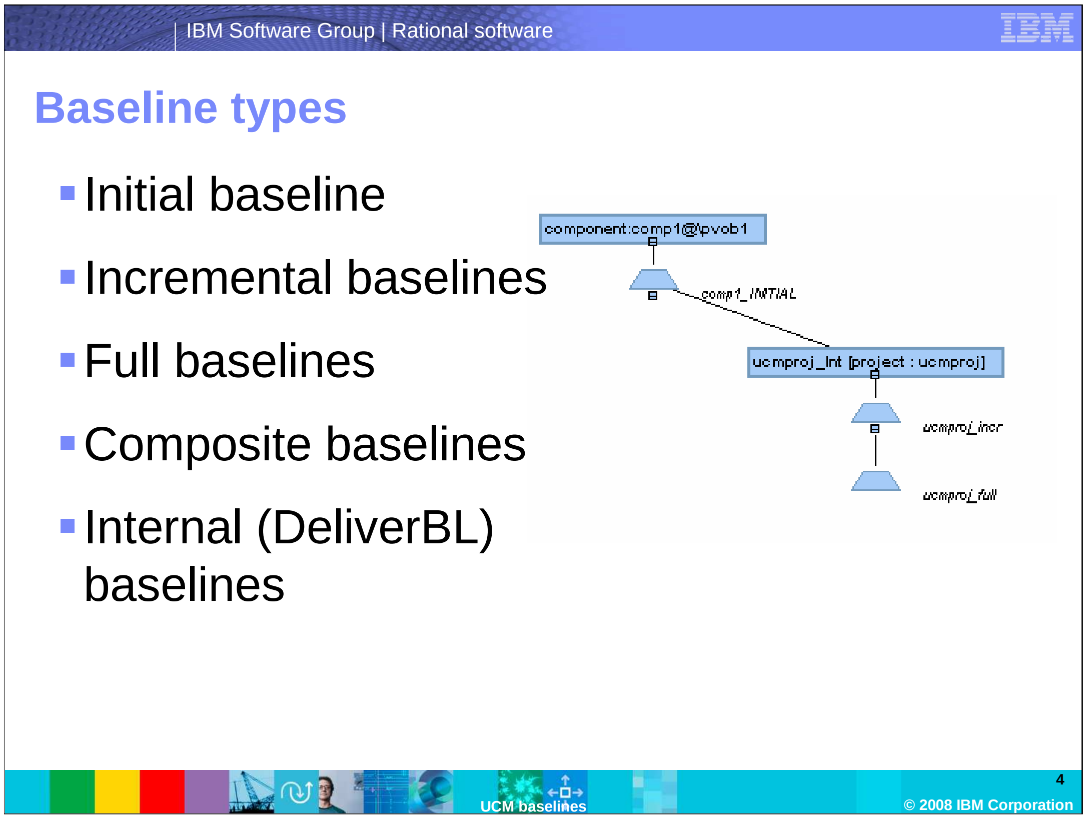
Several topics will be covered in this module. It will provide an overview of UCM baselines and how they apply within the UCM environment. It also will explain different types of baselines and what they are used for, how baselines are created and modified, and how baselines can be promoted.



UCM baselines

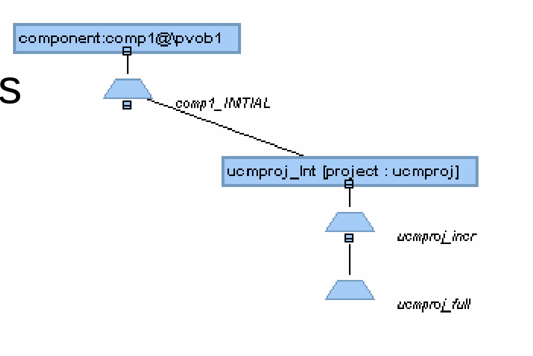
- Marks significant points in time during development life cycle
- A list of all necessary information/elements needed for a particular build version
- A snapshot in history that can be called upon at any point in the future
- A baseline in an existing project can be used as the initial starting point for a new project

There are many stages in the development life cycle such as build, test, QA and release. Baselines mark these stages during the development life cycle and are crucial to the project's overall success. Knowing and understanding what objects are being tested, built, or released is essential in day to day operations. The UCM baseline enables developers and software specialists to brand each pivotal point with an immutable uniquely named object. The object (the baseline) will become a list of all necessary information needed at the time of creation. The baseline can be called upon at any point in the future to reveal selected information at that point of the software life cycle; essentially, a snapshot of its history. A baseline in an existing project can be used as the initial starting point for a new project.



Baseline types

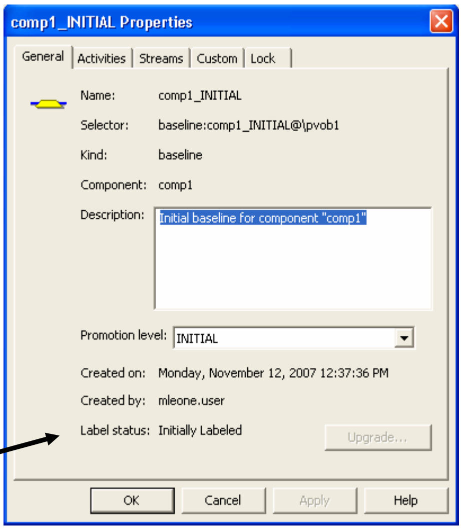
- Initial baseline
- Incremental baselines
- Full baselines
- Composite baselines
- Internal (DeliverBL) baselines



UCM baselines are broken up into five distinct types: initial baseline, incremental baselines, full baselines, composite baselines, and internal baselines. Four of these types are user-facing. The fifth type, the internal baseline, is used by UCM for specific actions. The next few slides will explain each type of baseline and its functionality.

Initial baselines

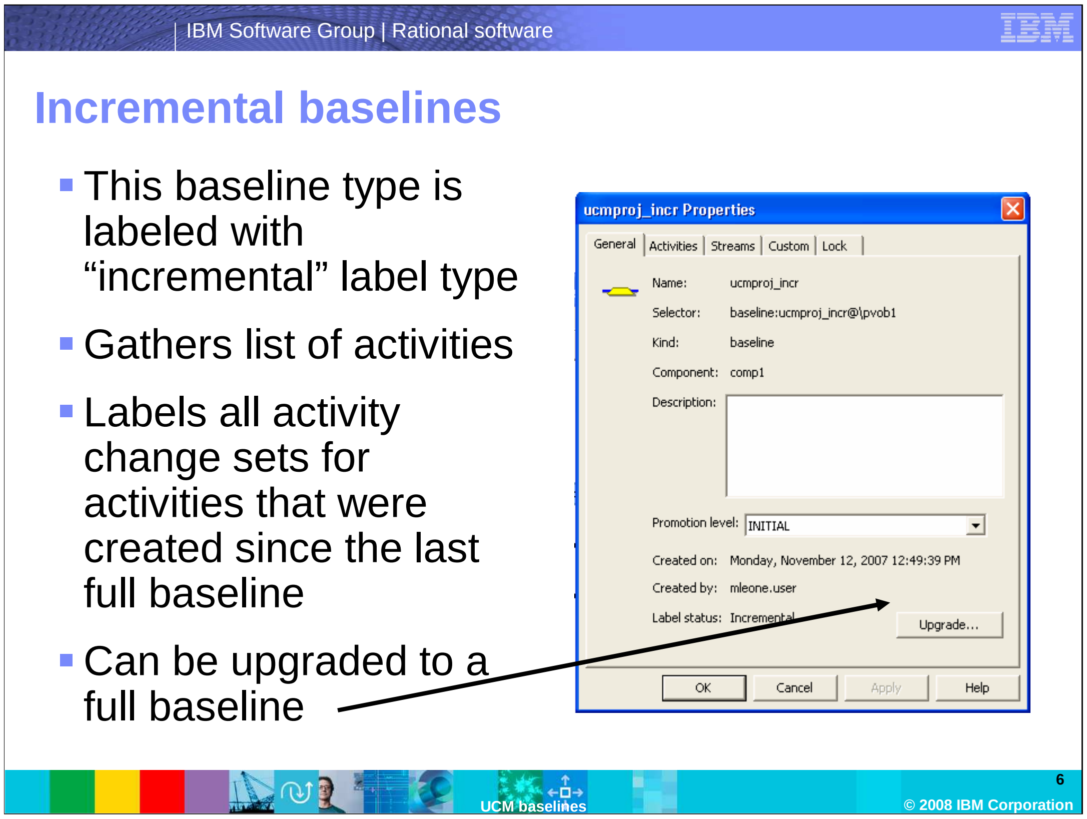
- The first baseline for any UCM component
- A representation for all elements main/0 version in that component
- The only baseline in a component that can hold the label status: Initially Labeled



comp1_INITIAL Properties

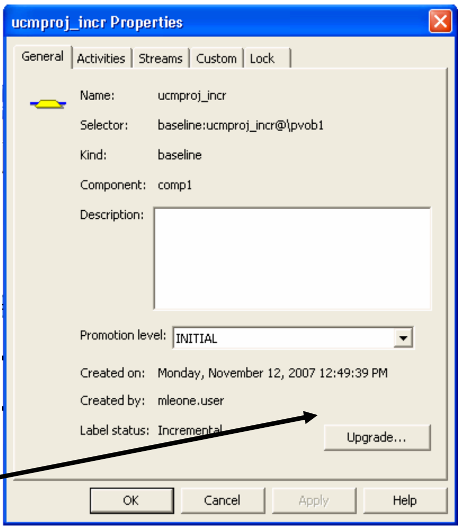
Name: comp1_INITIAL
Selector: baseline:comp1_INITIAL@\pvob1
Kind: baseline
Component: comp1
Description: Initial baseline for component "comp1"
Promotion level: INITIAL
Created on: Monday, November 12, 2007 12:37:36 PM
Created by: mleone.user
Label status: Initially Labeled

The initial baseline is the first baseline created for every UCM component. The initial baseline represents Main branch-version 0, for all UCM components. This becomes the starting point for development for each component. A long describe (or properties from the GUI) of the initial baseline will reveal that it is Initially Labeled, and no other baseline beyond this point can be Initially labeled.



Incremental baselines

- This baseline type is labeled with “incremental” label type
- Gathers list of activities
- Labels all activity change sets for activities that were created since the last full baseline
- Can be upgraded to a full baseline

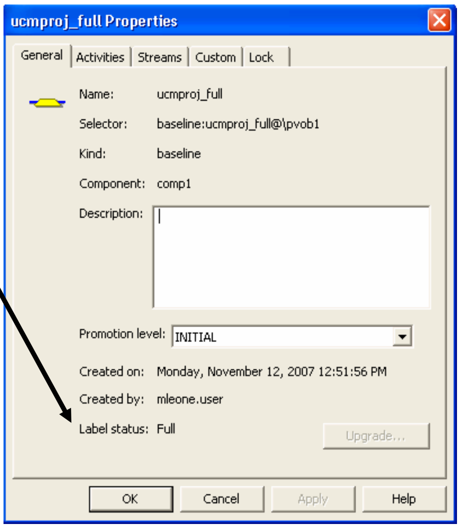


Name:	ucmproj_incr
Selector:	baseline:ucmproj_incr@\pvob1
Kind:	baseline
Component:	comp1
Description:	
Promotion level:	INITIAL
Created on:	Monday, November 12, 2007 12:49:39 PM
Created by:	mleone.user
Label status:	Incremental

The Incremental type of baseline is a baseline that has a corresponding label type. This means the baseline will be a gathered list of activities. The label type will be applied to the latest version of each changed element in the list of activities since the last full baseline. You can see this type of baseline by looking at the label status in a long describe or from the Windows® GUI as in the pictured example. Incremental baselines can be upgraded to full baselines to create a working backstop at that point.

Full baselines

- This baseline type is labeled with the “full” label type
- Gathers list of all activities
- Labels latest versions of all elements in the list of activities
- A full baseline is used as a backstop for future incremental baselines

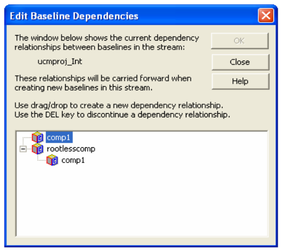


Name:	ucmproj_full
Selector:	baseline:ucmproj_full@\pvob1
Kind:	baseline
Component:	comp1
Description:	
Promotion level:	INITIAL
Created on:	Monday, November 12, 2007 12:51:56 PM
Created by:	mleone.user
Label status:	Full

This next type of baseline is a baseline that has a corresponding label type. The difference is that the label type is a full label type. This will make the baseline a new backstop for any new incremental baselines that will get created. The baseline will be a gathered list of activities. The label type will be applied to the latest version of each element in the list of activities. You can see this type of baseline by looking at the label status in a long describe or as pictured, in the GUI. An exception to this label type is an imported label as a baseline. This type of full baseline will be a label applied to all latest versions in the view that created the imported label.

Composite baseline

- A composite baseline is a gathered list of dependent baselines
- Composite baselines do not receive a corresponding label type
- Normally associated with a rootless component (A UCM component that does not have a physical root based in a VOB)



Edit Baseline Dependencies

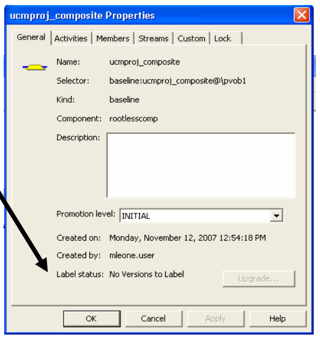
The window below shows the current dependency relationships between baselines in the stream:

ucmproj_Int

These relationships will be carried forward when creating new baselines in this stream.

Use drag/drop to create a new dependency relationship.
Use the DEL key to discontinue a dependency relationship.

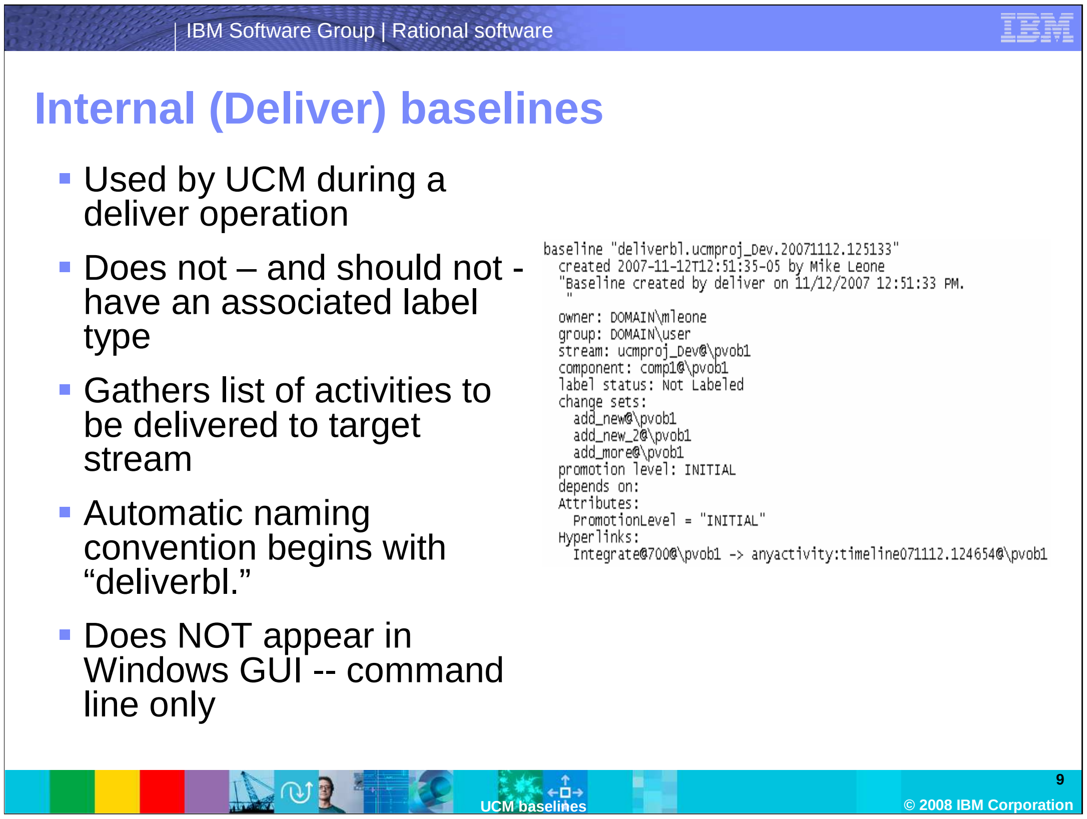
```
comp1
rootlesscomp
    comp1
```



ucmproj_composite Properties

Name:	ucmproj_composite
Selector:	baseline:ucmproj_composite@\pvob1
Kind:	baseline
Component:	rootlesscomp
Description:	
Promotion level:	INITIAL
Created on:	Monday, November 12, 2007 12:54:18 PM
Created by:	mleone.user
Label status:	No Versions to Label

A composite baseline simply described is a gathered list of dependent baselines (as seen in the baseline dependencies window on the top-right). This type of baseline will not have a corresponding label type. You can see this in a long describe in the label status or GUI. It will read “No Versions to Label.” This means the baseline is a list of dependent component baselines rather than activities. Note that composite baselines will be discussed again later in the module.


IBM Software Group | Rational software

Internal (Deliver) baselines

- Used by UCM during a deliver operation
- Does not – and should not - have an associated label type
- Gathers list of activities to be delivered to target stream
- Automatic naming convention begins with “deliverbl.”
- Does NOT appear in Windows GUI -- command line only

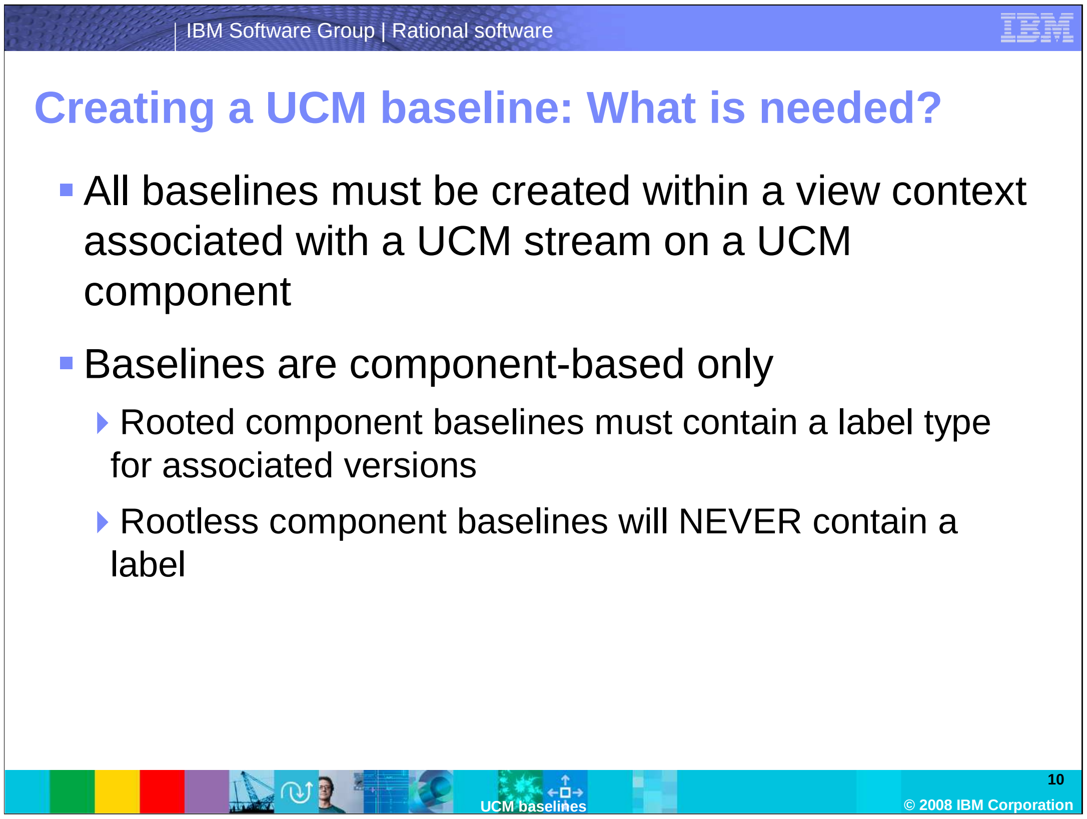
```
baseline "deliverbl.ucmproj_Dev.20071112.125133"
  created 2007-11-12T12:51:35-05 by Mike Leone
  "Baseline created by deliver on 11/12/2007 12:51:33 PM.
   "

  owner: DOMAIN\mleone
  group: DOMAIN\user
  stream: ucmproj_Dev@\pvob1
  component: comp1@\pvob1
  label status: Not Labeled
  change sets:
    add_new@\pvob1
    add_new_2@\pvob1
    add_more@\pvob1
  promotion level: INITIAL
  depends on:
  Attributes:
    PromotionLevel = "INITIAL"
  Hyperlinks:
    Integrate@700@\pvob1 -> anyactivity:timeline071112.124654@\pvob1
```

UCM baselines

9

© 2008 IBM Corporation

Internal baselines are also known as deliver baselines or “deliver-BL’s”. These baselines are not user-facing, though they can be seen by using the command line through such commands as “cleartool lsbl” when run on a stream that has been the source for UCM delivers. As the name suggests, these baselines are created as the result of a UCM deliver. When a deliver is initiated on the source stream, it goes and gathers a list of activities with changes that need to be delivered to the target. These activities are added to an internal deliver baseline and sent to the target stream. A deliver baseline will have a name generated automatically by UCM and always begin with the format “deliverbl” as seen in the accompanying screen capture. Also note that a deliver baseline will never receive a label, nor should it ever be labeled for consideration to be recommended. A hyperlink for this baseline is drawn not to a label type, but to an internal timeline entry for that component.


IBM Software Group | Rational software

Creating a UCM baseline: What is needed?

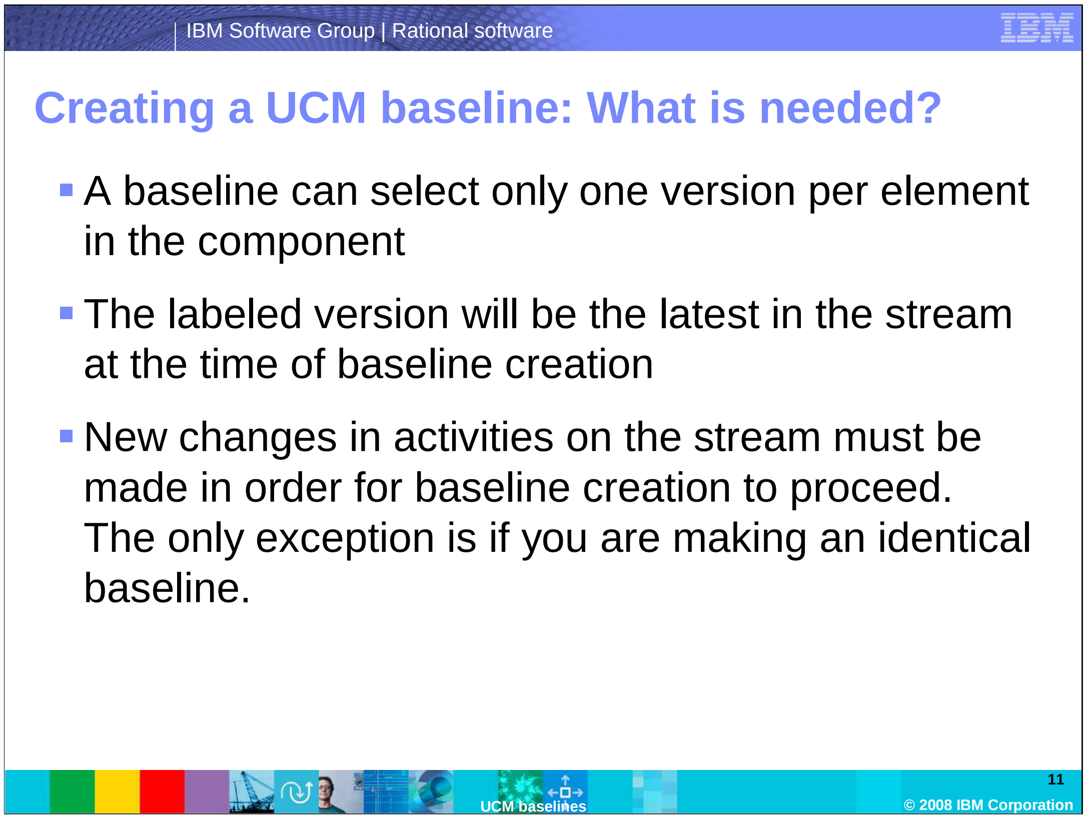
- All baselines must be created within a view context associated with a UCM stream on a UCM component
- Baselines are component-based only
  - Rooted component baselines must contain a label type for associated versions
  - Rootless component baselines will NEVER contain a label

UCM baselines

10

© 2008 IBM Corporation

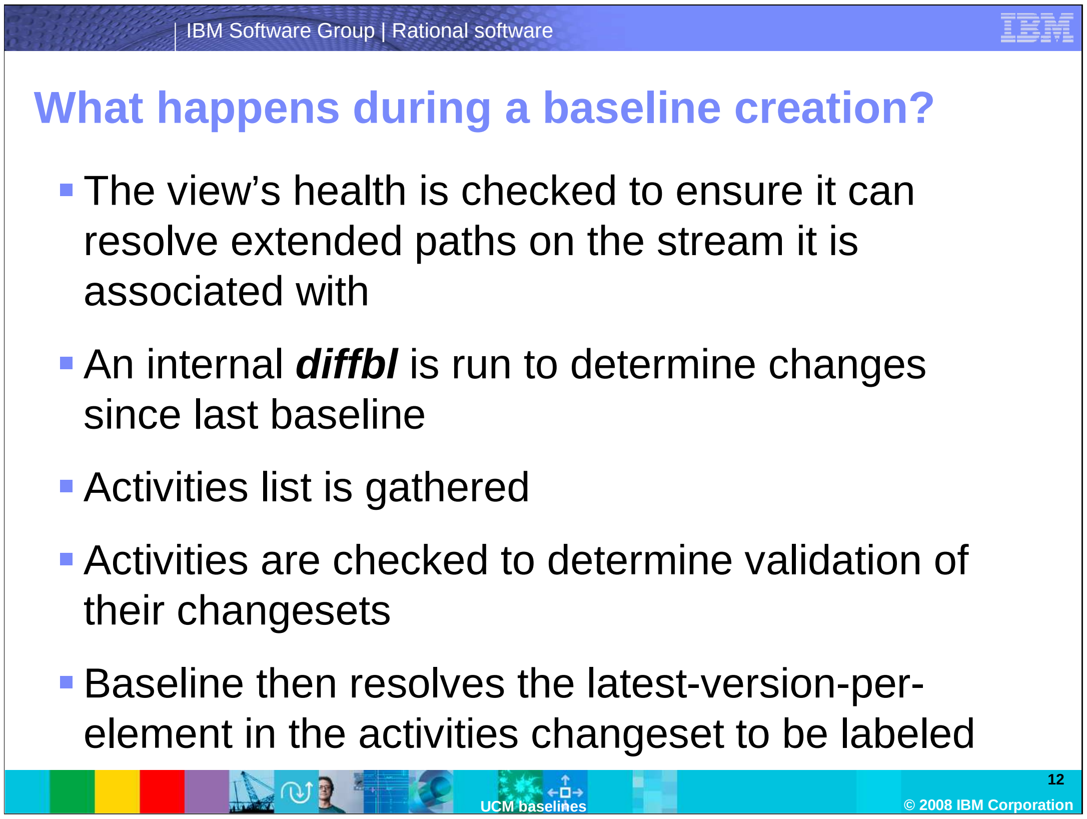
All UCM baselines are related to a single component. A baseline must be created in a view context associated with a UCM stream, though Initial baselines and imported baselines are the only exceptions to this rule. Outside the exceptions to the rule, a baseline requires at minimum: a stream, component, and view associated with that stream. Baseline rules differ between rooted and rootless components. Rooted component baselines must contain a label type for versions associated with the baseline, while rootless component baselines will never contain a label type since there are no versions to label.



Creating a UCM baseline: What is needed?

- A baseline can select only one version per element in the component
- The labeled version will be the latest in the stream at the time of baseline creation
- New changes in activities on the stream must be made in order for baseline creation to proceed. The only exception is if you are making an identical baseline.

Baselines only can select one version per element. The version labeled should be the latest version in the stream at the time of baseline creation. Unless you are making an identical baseline to the previous one made, new changes to activities in the stream must be made to allow for a new baseline creation.


IBM Software Group | Rational software

What happens during a baseline creation?

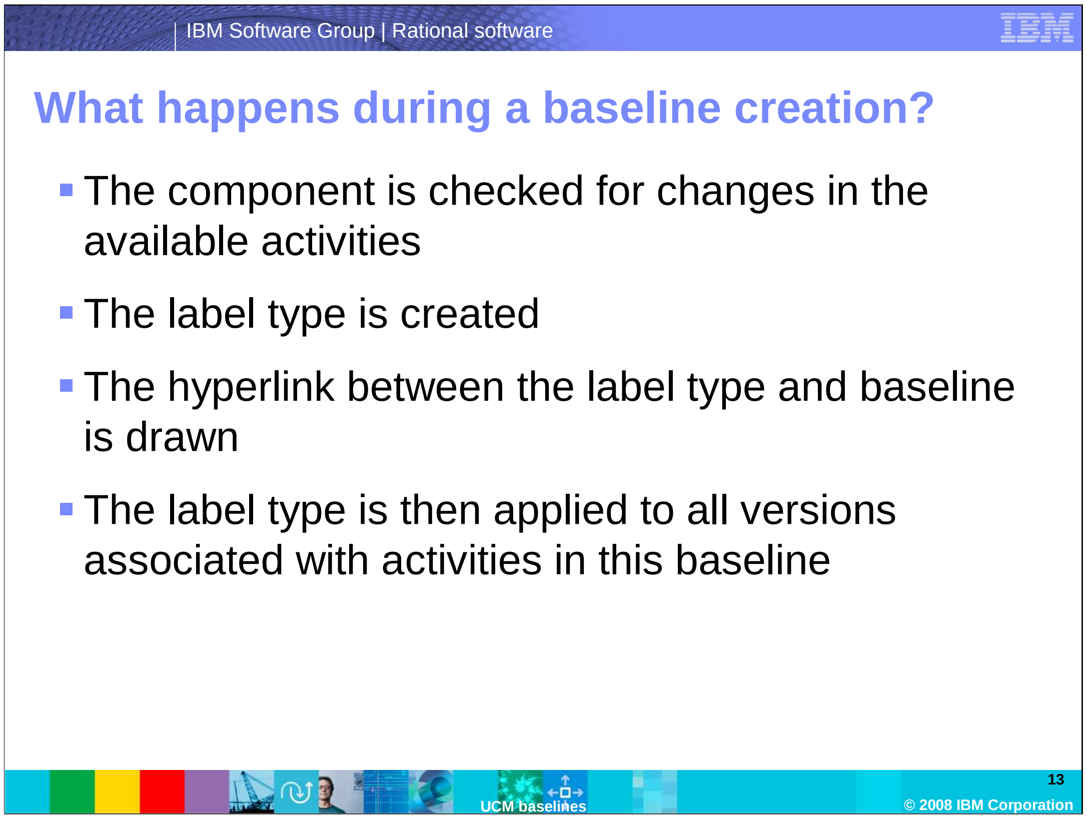
- The view's health is checked to ensure it can resolve extended paths on the stream it is associated with
- An internal ***diffbl*** is run to determine changes since last baseline
- Activities list is gathered
- Activities are checked to determine validation of their changesets
- Baseline then resolves the latest-version-per-element in the activities changeset to be labeled

UCM baselines

12

© 2008 IBM Corporation

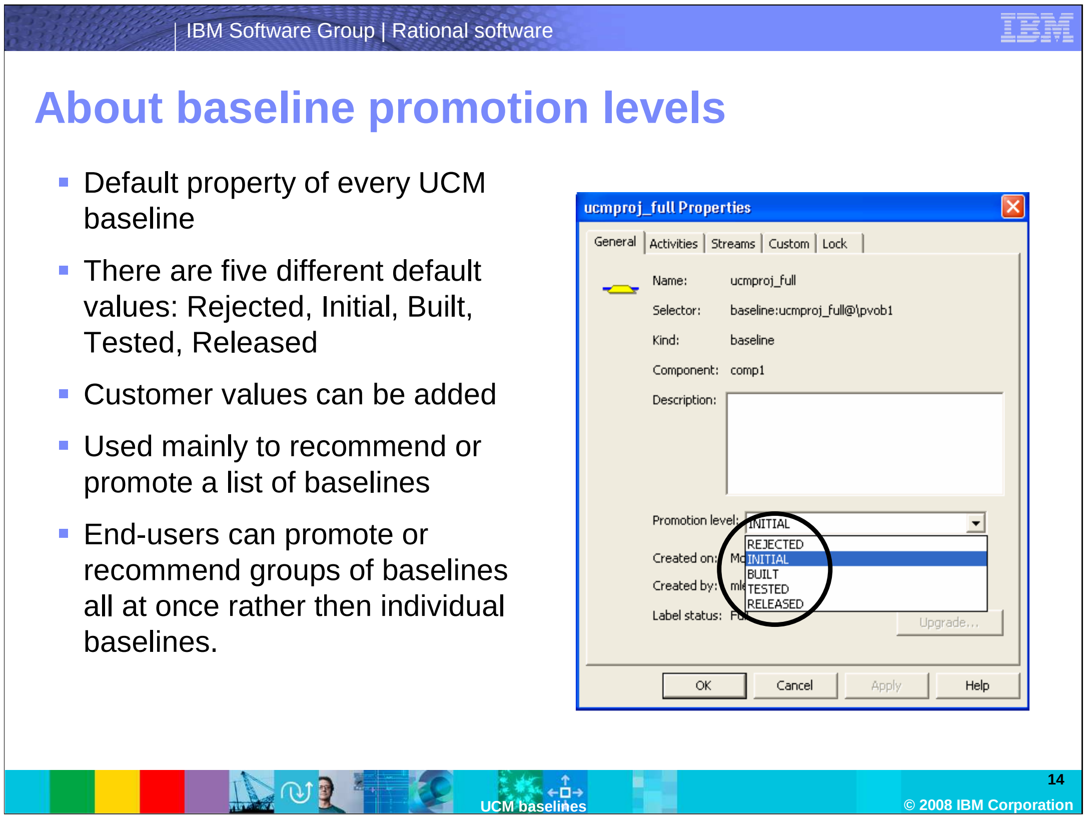
When the baseline creation is initiated, the first action is a health check of the view being used. UCM needs to assure that view is available to resolve version extended paths on the stream that the view is associated with. Next, a *diffbl* (diff BL) action is run to determine any new or changed activities since the last baseline was laid. The list of activities is then gathered to analyze versions that require an association with the baseline, then a health check is performed. That health check on the activities is necessary to ensure all change sets can be validated to actual versions under source control that require the new label. Next, the baseline resolves one single "latest version" for each element in each activity change set to be labeled.



What happens during a baseline creation?

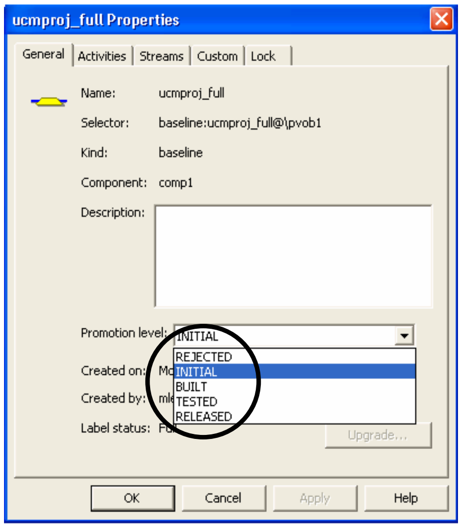
- The component is checked for changes in the available activities
- The label type is created
- The hyperlink between the label type and baseline is drawn
- The label type is then applied to all versions associated with activities in this baseline

At this point, each component is checked for changes within the activities available. A label type is created along with the hyperlink that connects the baseline to the label type. This hyperlink is known as a “BaselineLbType” hyperlink and is drawn between the baseline in the PVOB to the label type in the component VOB.



About baseline promotion levels

- Default property of every UCM baseline
- There are five different default values: Rejected, Initial, Built, Tested, Released
- Customer values can be added
- Used mainly to recommend or promote a list of baselines
- End-users can promote or recommend groups of baselines all at once rather then individual baselines.



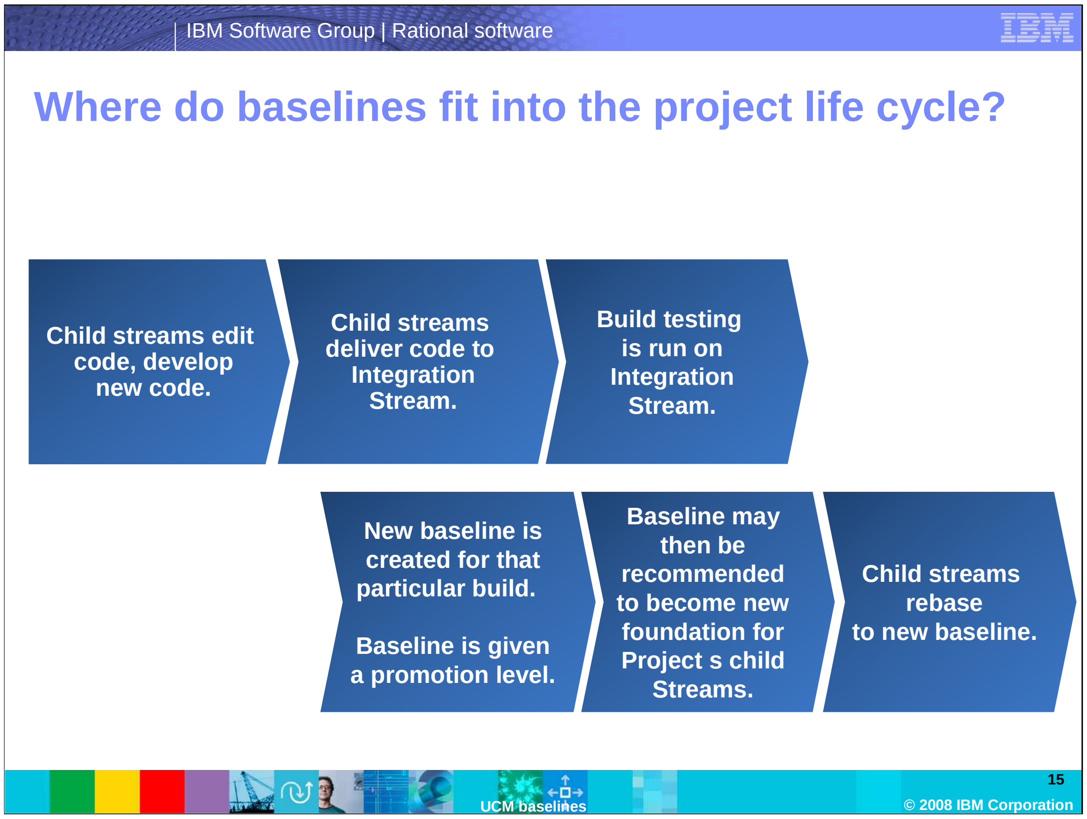
ucmproj_full Properties

Name:	ucmproj_full
Selector:	baseline:ucmproj_full@\pvob1
Kind:	baseline
Component:	comp1
Description:	
Promotion level:	INITIAL
Created on:	Mo
Created by:	ml
Label status:	Fu

REJECTED
INITIAL
BUILT
TESTED
RELEASED

Baselines can be set to different levels of promotion. The baseline promotion level is a default attribute of every UCM baseline. The promotion level allows a relationship to be established by means of a common value shared between baselines. For a default ClearCase UCM installation, there are five values for the baseline promotion level. They are: Rejected, Initial, Built, Tested, and Released. The tool is not limited to these five, and custom values can be added to this list if need be.

These promotion levels are mainly used for promoting or recommending (or both) a list or group of baselines. The promotion level allows you to recommend all baselines that reside at a specific value, rather than having to take on the cumbersome task of recommending each individual baseline separately.

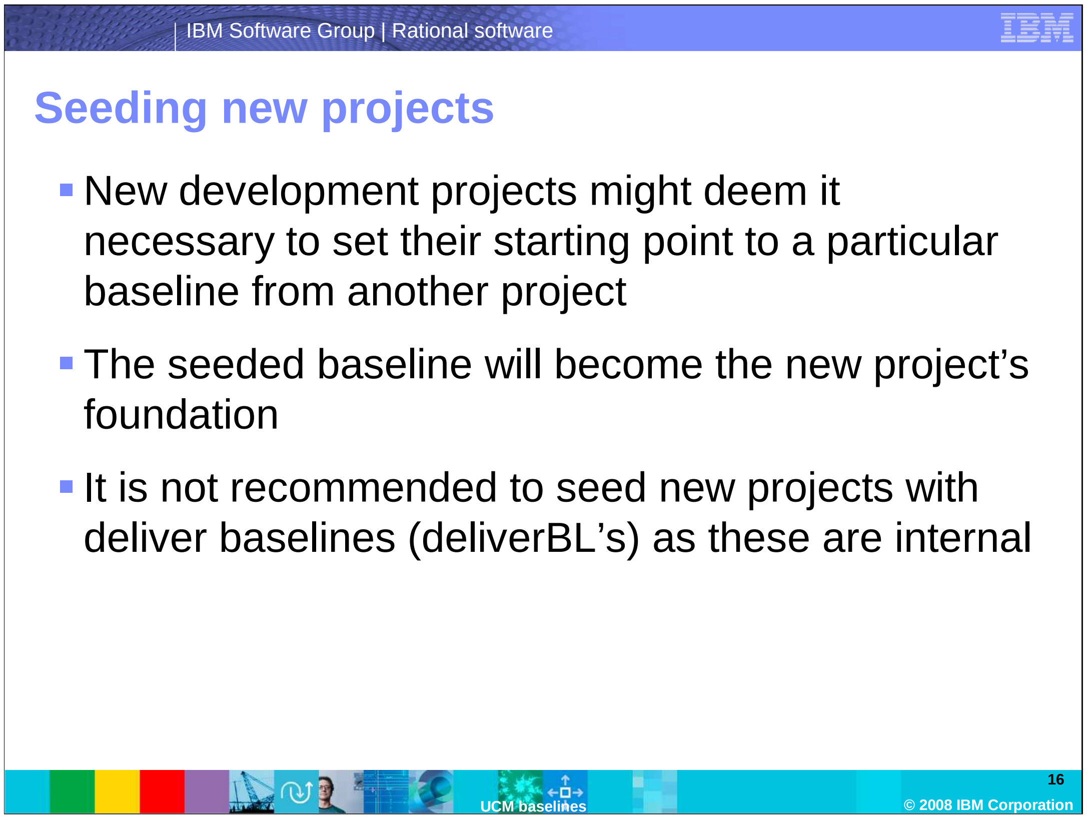

IBM Software Group | Rational software

Where do baselines fit into the project life cycle?

Child streams edit code, develop new code.
Child streams deliver code to Integration Stream.
Build testing is run on Integration Stream.
New baseline is created for that particular build.
Baseline is given a promotion level.
Baseline may then be recommended to become new foundation for Project s child Streams.
Child streams rebase to new baseline.

UCM baselines 15 © 2008 IBM Corporation

So where do these baselines actually fit into the UCM Project life cycle? At component creation, the Initial baseline is created, though this baseline will always be empty, sitting as a placeholder for main/0. The environment might or might not call for a label to be imported from base ClearCase as a new UCM baseline. At this point, child streams are created from either the initial or imported baseline and development begins. Over time, child stream will deliver their code to the Integration stream, and builds and testing are run on that stream. Once the particular build is validated by the development group, they might want to lay a new baseline to mark this point in time for this particular set of project code. The baseline then is created as either a full or incremental baseline, and a promotion level is set. Once the baseline is successfully labeled, it can be set as the recommended baseline for this component on this stream, allowing child streams to rebase to this new baseline, which will become part of the child streams' foundation. The cycle then begins again.



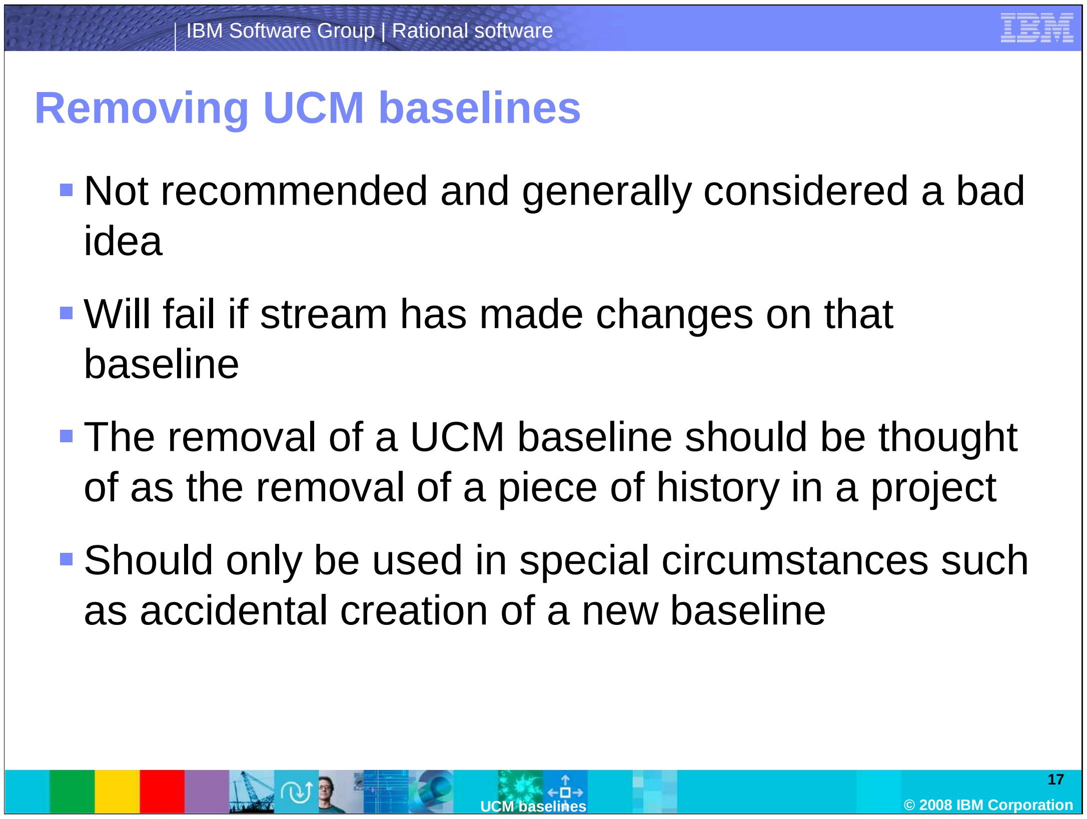
Seeding new projects

- New development projects might deem it necessary to set their starting point to a particular baseline from another project
- The seeded baseline will become the new project's foundation
- It is not recommended to seed new projects with deliver baselines (deliverBL's) as these are internal

Baselines are also useful when it comes to capturing specific points in time for a project environment. At times, new development projects might find it necessary to set their starting point to a particular recommended baseline on another project's Integration stream. This is known as baseline seeding of a project. The seeded baseline will now become the new project's foundation baseline, and development will start from that snapshot in time and move forward with no effect on the original project.

Alternately, a UCM development team may at some point find it necessary to view the contents of a baseline that was laid at some past point in the project's history. For this purpose, where no additional work needs to be performed on that older baseline, a read-only child stream may be created within the project and be seeded with that baseline of interest. Once the information has been viewed, the stream can be removed with zero effect on the project.

With respect to "deliver baselines," it is never recommended to seed new projects with deliver baselines, as they are meant for internal use only by UCM.



IBM Software Group | Rational software

Removing UCM baselines

- Not recommended and generally considered a bad idea
- Will fail if stream has made changes on that baseline
- The removal of a UCM baseline should be thought of as the removal of a piece of history in a project
- Should only be used in special circumstances such as accidental creation of a new baseline

UCM baselines

17

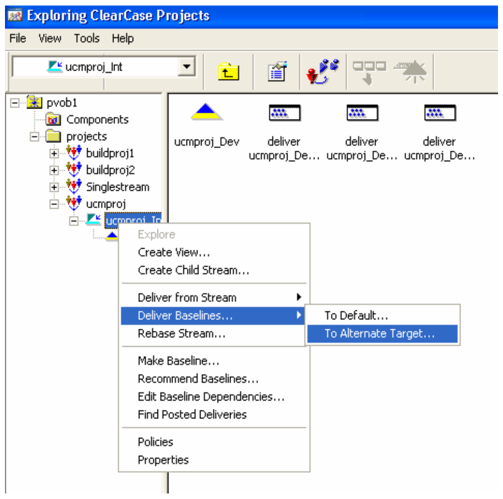
© 2008 IBM Corporation

Baselines are essentially captures of unique moments on a project. For this reason, it is not recommended, and generally is considered a bad idea, to remove a UCM baseline. In fact, removing a baseline will fail if a stream has made any changes to the versions associated with that baseline.

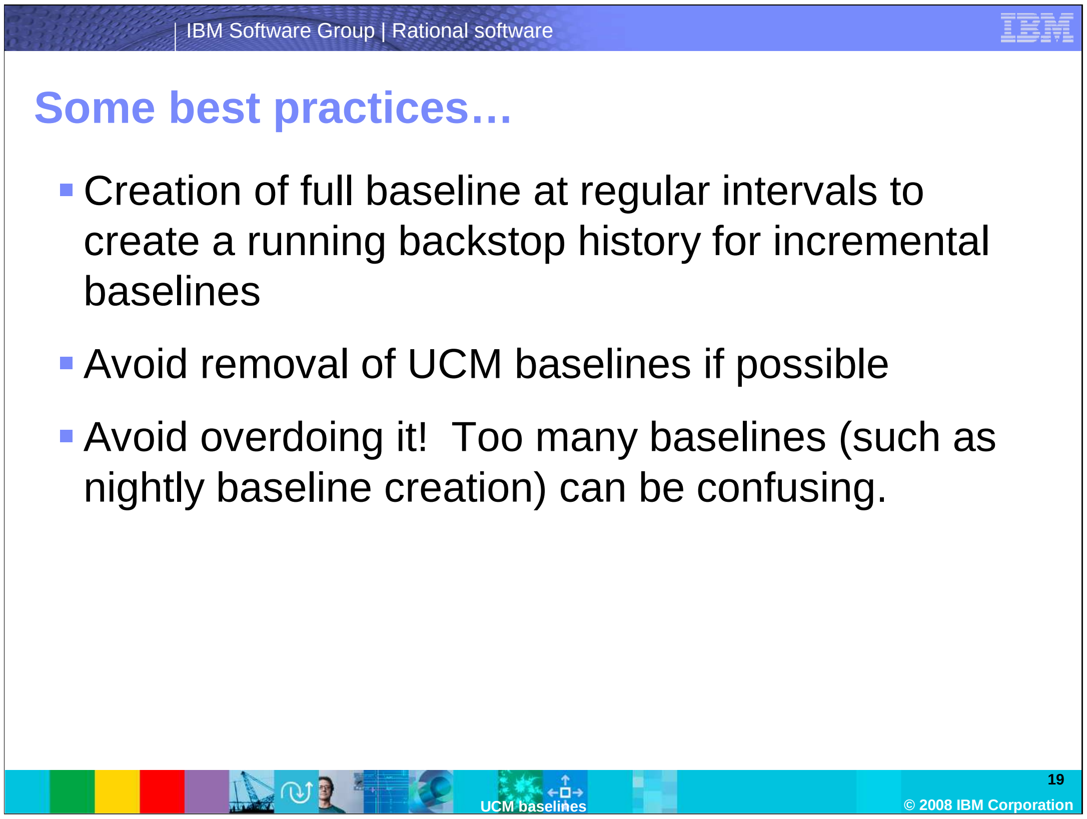
The removal of a UCM baseline should be equated with the removal of a piece of the project’s history and should only be used under special circumstances. Often, the cleartool rmbl command will be run immediately after an accidental baseline creation, before that baseline is rebased to and edited by any streams.

Delivering baselines

- UCM allows users to deliver actual baselines between different projects
- Project policies must be edited to allow for deliveries of baselines
- The deliver will attempt to merge all changesets within the baseline with the versions on the target stream's view.



UCM introduces the functionality to deliver entire baselines between streams on a project, and even between two different projects. The deliver of a UCM baseline will attempt to merge all change sets contained in the activities within the baseline with the versions contained on the target stream's view. For baseline deliveries between projects, the project policies must be edited to allow for deliveries of baselines.



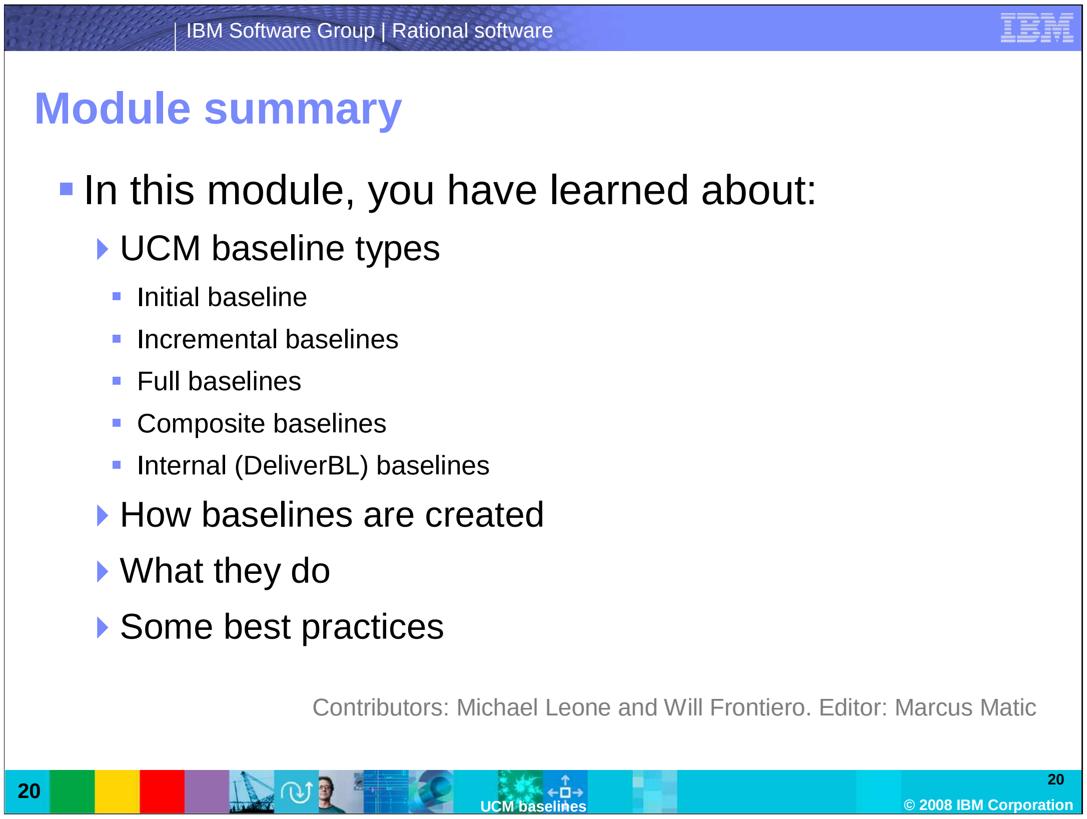
Some best practices…

- Creation of full baseline at regular intervals to create a running backstop history for incremental baselines
- Avoid removal of UCM baselines if possible
- Avoid overdoing it! Too many baselines (such as nightly baseline creation) can be confusing.

Some general best practices to keep in mind with respect to baselines include creation of full baselines at regular intervals. This allows a regular creation of a backstop for the related component. A backstop is used by incremental baseline to map version back to the last full baseline that was laid. If full baselines are created every so often, an incremental baseline will not have to travel as far back to find the last backstop that was created.

As discussed in the previous slide, removal of UCM baselines should be avoided wherever possible.

Finally, avoid the temptation to create too many baselines. Remember, UCM baselines exist to create a unique label on element versions that are associated with a special event, such as a release. For example, a nightly baseline creation will do little more than start to clutter up your component over time. This practice will make it more difficult for users in the environment to locate baselines associated with special events in a sea of baselines.


IBM Software Group | Rational software

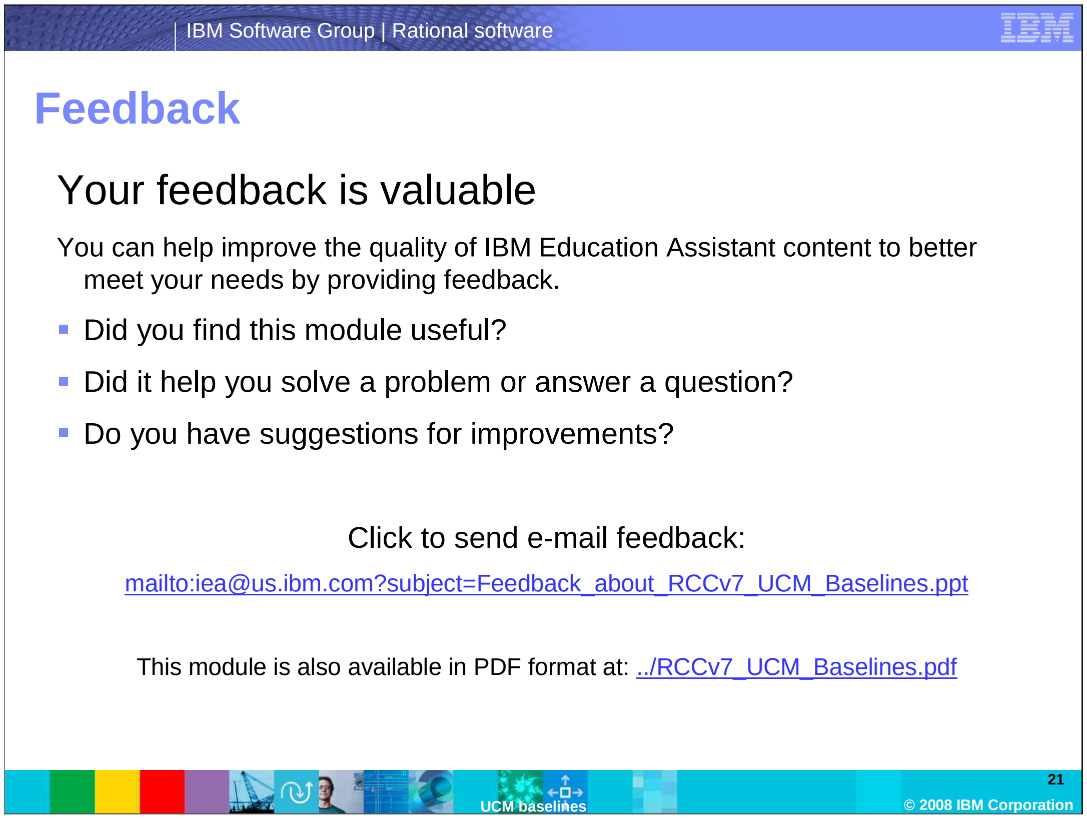
Module summary

- In this module, you have learned about:
  - UCM baseline types
    - Initial baseline
    - Incremental baselines
    - Full baselines
    - Composite baselines
    - Internal (DeliverBL) baselines
  - How baselines are created
  - What they do
  - Some best practices

Contributors: Michael Leone and Will Frontiero. Editor: Marcus Matic

20 UCM baselines 20 © 2008 IBM Corporation

In this module, you have learned about the types of UCM baselines and what purpose they serve. You've learned about baseline creation, and what takes place behind the scenes during that process, and some baseline best practices.



IBM Software Group | Rational software

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_Baselines.ppt

This module is also available in PDF format at: ../RCCv7_UCM_Baselines.pdf

UCM baselines

21

© 2008 IBM Corporation

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

Trademarks, copyrights, and disclaimers