Babbage Harvest Difference Engine Help

Release 1.00

Contents

Chapter 1

Introduction

What Is Babbage

Babbage is an extension to CA Harvest Software Change Manager Workbench which adds the ability to compare the structure of individual project lifecycles. Lifecycles can be compared with one another, or with templates. For enterprises with multiple brokers, lifecycles can also be compared across brokers.

Babbage is named after *Charles Babbage*, 1791-1871, inventor of the Difference Engine.

Lifecycle Differencing

In Harvest, the project lifecycle is fully customizable, and is defined in terms of States, Views and Processes along with their associated Access Control. Lifecycles are defined on a per project basis, and although they can be created from templates, changes to the templates do not propogate to the projects created from them. In order to ensure that all projects are consistent with enterprise-wide standards, it is necessary to have some way of comparing them. This enables differences to be seen, reacted to and/or propogated as necessary.

Let us consider the uses for such a tool:

- Determining which lifecycle template was used to create a project
- Helping keep projects in sync with the templates from which they were created
- Enabling access control changes to be detected
- Finding processes which short-circuit the development process

We are all used to comparison tools that compare text files and even structured files like XML. Babbage works on the same principal, but rather than working on files, it directly compares the lifecycle definitions stored in the Harvest database.

Chapter

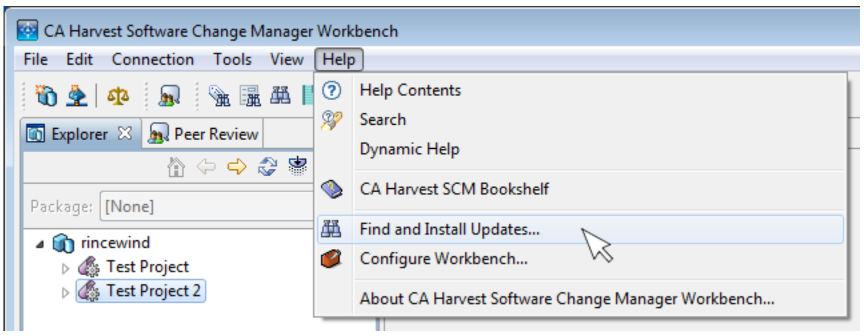
2

Installing Babbage

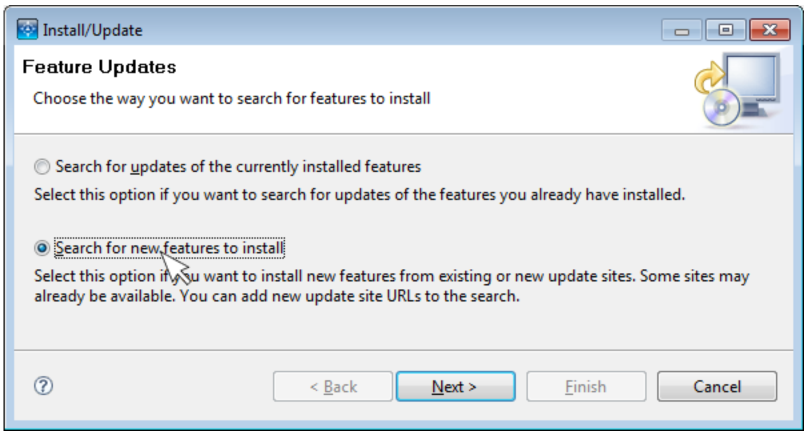
Installing Babbage

Babbage installs as a plug-in to CA Harvest Workbench.

From the main Workbench menu, select **Help** > **Find and Install Updates...**



In the dialog that opens select the **Search for new features to install** radio-button, and click **Next**.


Install/Update

Feature Updates

Choose the way you want to search for features to install

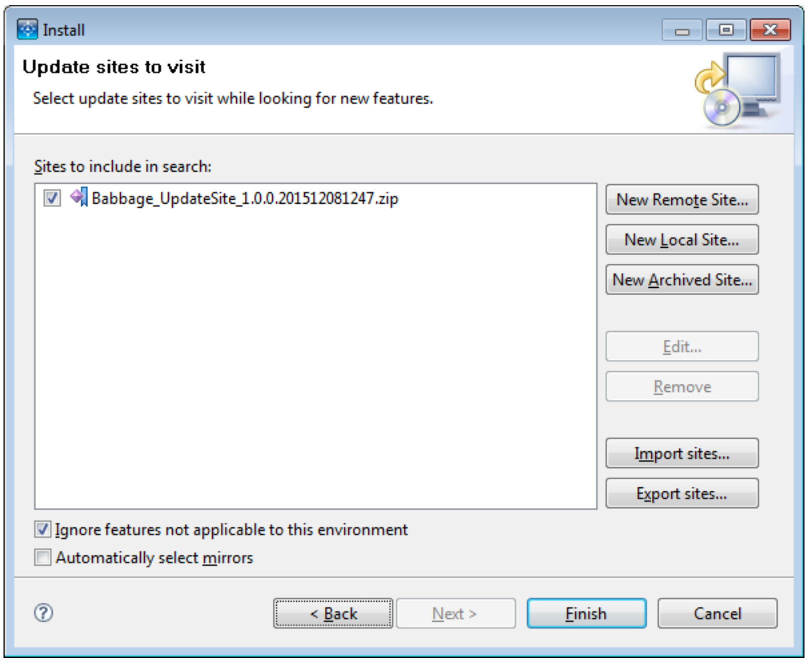
( ) Search for updates of the currently installed features

Select this option if you want to search for updates of the features you already have installed.

(•) Search for new features to install

Select this option if you want to install new features from existing or new update sites. Some sites may already be available. You can add new update site URLs to the search.

On the next page of the dialog, click on the **New Archived Site...** button and navigate to the folder where you have downloaded Babbage. Select the downloaded zip file and click **Open**. Click **OK** on the Edit Local Site dialog and the zip file should appear in the **Sites to include in search** list. Click the **Finish** button.


Install

Update sites to visit

Select update sites to visit while looking for new features.

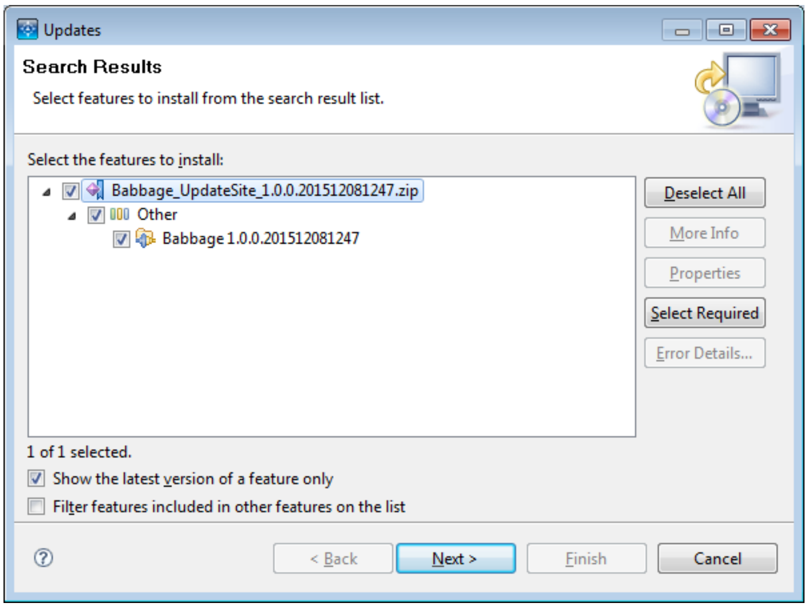
Sites to include in search:

[✓] Babbage_UpdateSite_1.0.0.201512081247.zip

[✓] Ignore features not applicable to this environment

[ ] Automatically select mirrors

Tick the checkbox next to the Babbage plugin and click **Next**.


Updates

Search Results

Select features to install from the search result list.

Select the features to install:

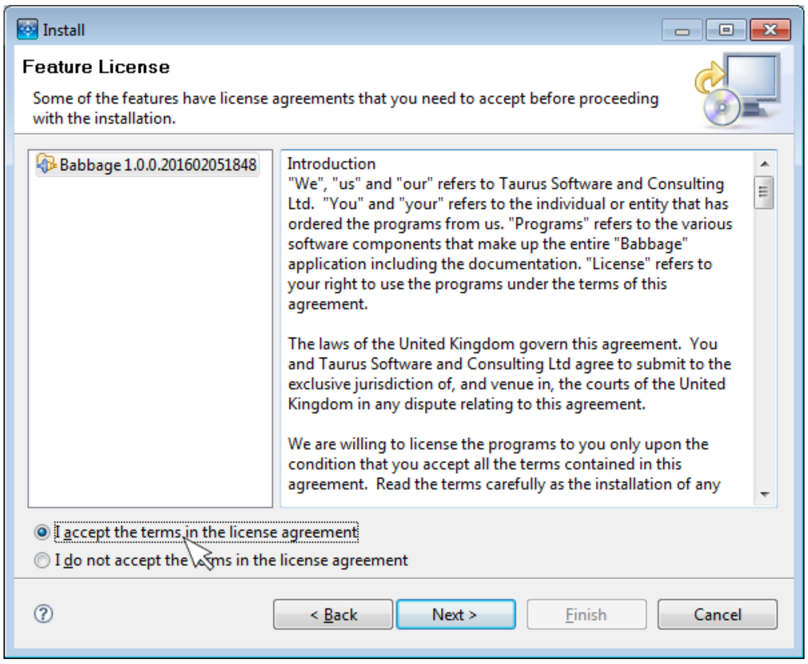
[✓] Babbage_UpdateSite_1.0.0.201512081247.zip
  [✓] Other
    [✓] Babbage 1.0.0.201512081247

1 of 1 selected.

[✓] Show the latest version of a feature only

[ ] Filter features included in other features on the list

Select the **I accept the terms in the license agreement** radio-button and click **Next**.


Install
Feature License
Some of the features have license agreements that you need to accept before proceeding with the installation.
Babbage 1.0.0.201602051848
I accept the terms in the license agreement
I do not accept the terms in the license agreement
< Back
Next >
Finish
Cancel

Introduction
"We", "us" and "our" refers to Taurus Software and Consulting Ltd. "You" and "your" refers to the individual or entity that has ordered the programs from us. "Programs" refers to the various software components that make up the entire "Babbage" application including the documentation. "License" refers to your right to use the programs under the terms of this agreement.

The laws of the United Kingdom govern this agreement. You and Taurus Software and Consulting Ltd agree to submit to the exclusive jurisdiction of, and venue in, the courts of the United Kingdom in any dispute relating to this agreement.

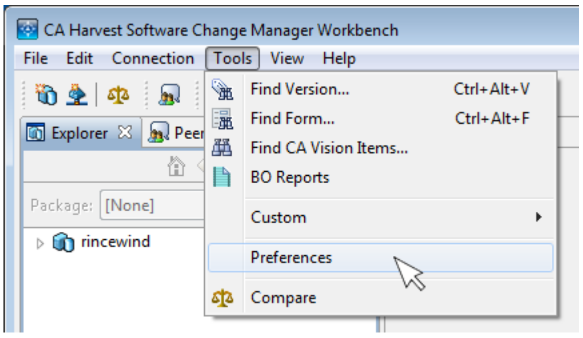
We are willing to license the programs to you only upon the condition that you accept all the terms contained in this agreement. Read the terms carefully as the installation of any

Click **Finish** and then click **Install** when prompted to verify the feature. Say **Yes** when asked if you wish to restart Workbench.

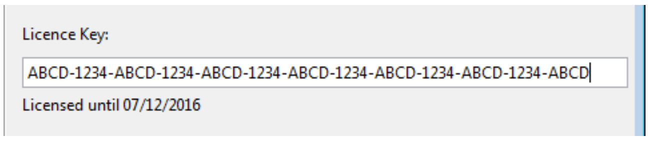
Licensing Babbage

Babbage requires a licence key to be entered before it can be used.

From the main Workbench menu, select **Tools** > **Preferences** and click on **Babbage**.



Enter the licence key, exactly as it appears, into the **Licence Key** box and click **Apply**.



Licence Key:

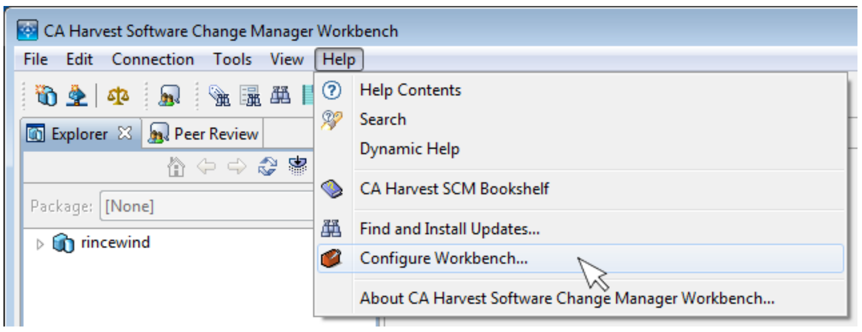
ABCD-1234-ABCD-1234-ABCD-1234-ABCD-1234-ABCD-1234-ABCD-1234-ABCD

Licensed until 07/12/2016

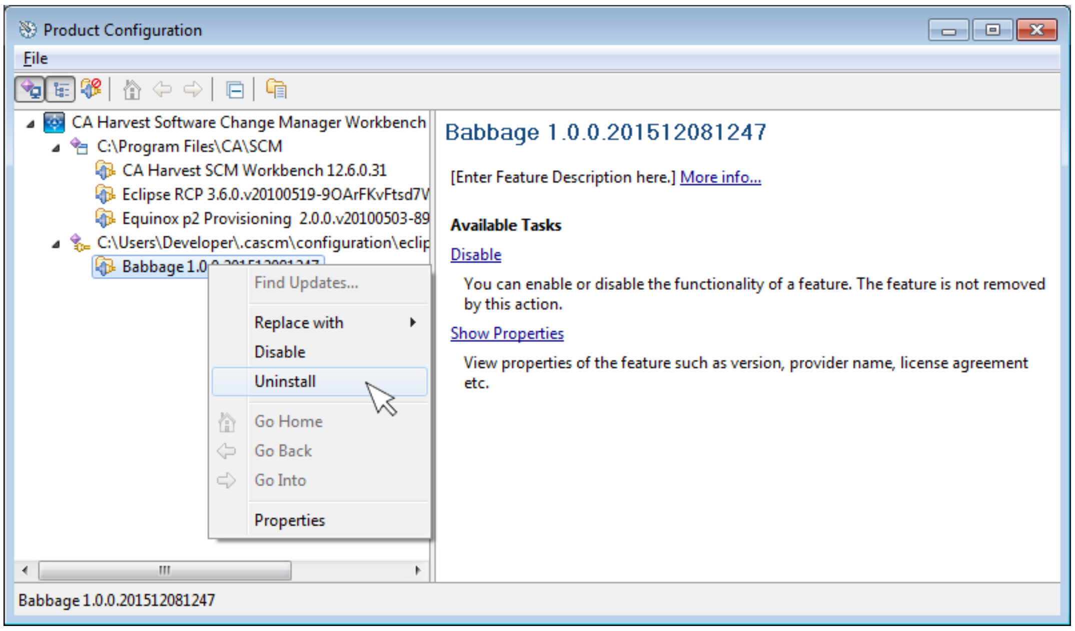
Babbage will immediately verify the key and tell you the expiry date of the key or indicate that the key you have entered is invalid. Please note that Babbage licence keys are host-locked, so you need to ensure that you enter the right key for the machine you are using.

Uninstalling Babbage

To uninstall Babbage, select **Help** > **Configure Workbench...** from the main Workbench menu.



Expand the tree-view so that you can see the Babbage plugin, then right-click on the plugin and select **Uninstall** from the menu that appears.


Product Configuration
File
CA Harvest Software Change Manager Workbench
C:\Program Files\CA\SCM
CA Harvest SCM Workbench 12.6.0.31
Eclipse RCP 3.6.0.v20100519-9OArFKvFtsd7V
Equinox p2 Provisioning 2.0.0.v20100503-89
C:\Users\Developer\.cascm\configuration\eclip
Babbage 1.0
Find Updates...
Replace with
Disable
Uninstall
Go Home
Go Back
Go Into
Properties
Babbage 1.0.0.201512081247
[Enter Feature Description here.] More info...
Available Tasks
Disable
Show Properties
Babbage 1.0.0.201512081247

You can enable or disable the functionality of a feature. The feature is not removed by this action.

View properties of the feature such as version, provider name, license agreement etc.

Click **OK** to confirm that you wish to uninstall the plugin and say **Yes** when asked if you wish to restart Workbench.

You can re-install the plugin at any time by following the installation instructions again.

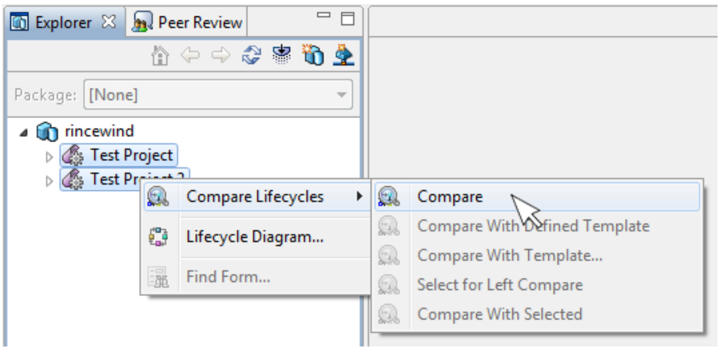
Chapter 3

Making Comparisons

There are several ways in which Babbage can be invoked.

Compare Lifecycles

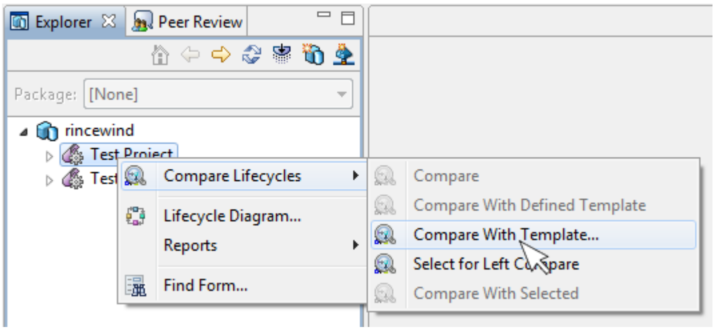
Using the Workbench Explorer View select two projects, right-click and select **Compare Lifecycles** > **Compare** from the context menu. The project lifecycles will be scanned and compared.



If you have more than one broker, it is possible run a comparison across brokers by selecting a project on one broker and then Ctrl-clicking a project on another broker and then right-clicking and selecting **Compare Lifecycles** > **Compare**.

Compare With Template

Using the Workbench Explorer View, right-click on a project and select **Compare Lifecycles** > **Compare With Template...** from the context menu. This will open the Select Template Dialog.

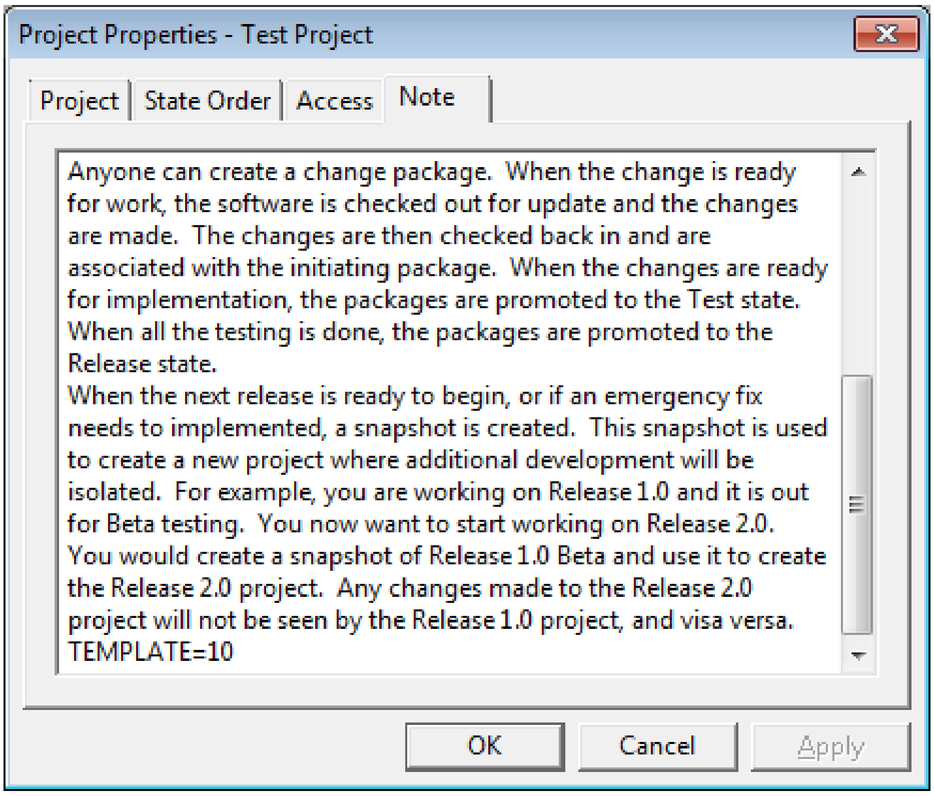


Using the dialog, select a project template to compare against and click **OK**. The project lifecycle will be scanned and compared with the selected template.

Compare With Defined Template

Using the Workbench Explorer View, right-click on a project and select **Compare Lifecycles** > **Compare With Defined Template** from the context menu. The project lifecycle will be scanned and compared with the defined template.

To define which template the project lifecycle was created from, you need to add some extra markup to the Notes field of the project. Using *Administrator*, right-click on the project and select **Properties**.



Anyone can create a change package. When the change is ready for work, the software is checked out for update and the changes are made. The changes are then checked back in and are associated with the initiating package. When the changes are ready for implementation, the packages are promoted to the Test state. When all the testing is done, the packages are promoted to the Release state.
When the next release is ready to begin, or if an emergency fix needs to implemented, a snapshot is created. This snapshot is used to create a new project where additional development will be isolated. For example, you are working on Release 1.0 and it is out for Beta testing. You now want to start working on Release 2.0. You would create a snapshot of Release 1.0 Beta and use it to create the Release 2.0 project. Any changes made to the Release 2.0 project will not be seen by the Release 1.0 project, and visa versa.
TEMPLATE=10

Click on the **Notes** tab and edit the text to add the following on a new line:

```
TEMPLATE=123
```

Where 123 is the object id of the template project. You can find this by manually comparing the project against the template and then selecting the project node and looking for ENVOBJID in the right-hand column.

Now when you right-click on the project lifecycle, the **Compare With Defined Template** menu option will be enabled and selecting it will compare with the template you have defined.

Note: Object ids are used for this feature as templates can potentially be renamed. As templates must exist in the same database as the projects they are used to create, this feature only allows you to reference a template within the same broker.

Select For Left Compare

Using the Workbench Explorer View, right-click on a project and select **Compare Lifecycles** > **Select For Left Compare** from the context menu. The project lifecycle will be remembered and can be used with the **Compare With Selected** menu option. This provides a more convenient way of selecting projects which are some distance apart in the Explorer tree or are on different brokers. It also provides an easy way of comparing multiple projects with the same base project.

Compare With Selected

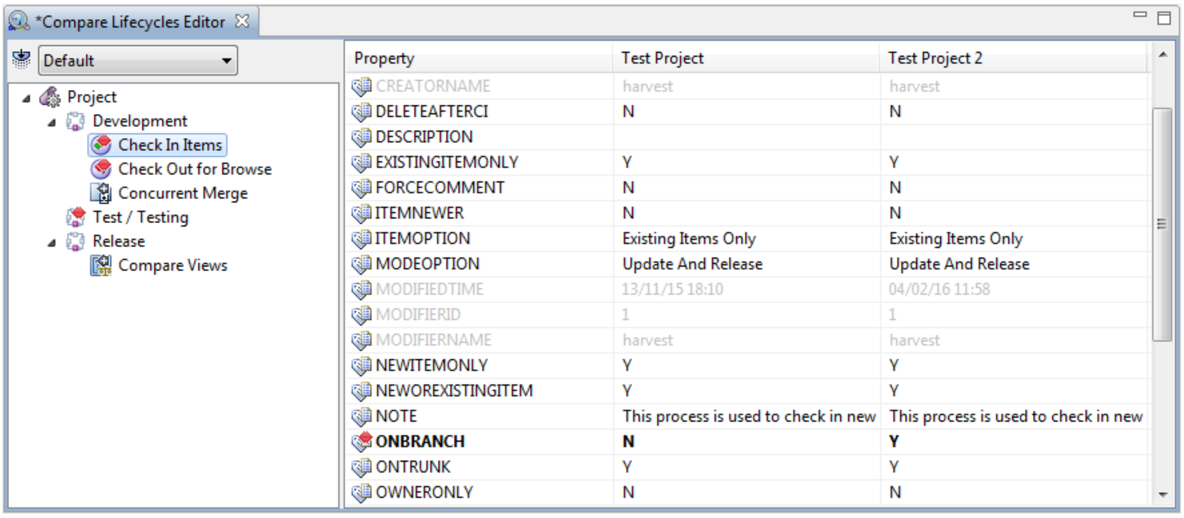
Using the Workbench Explorer View, right-click on a project and select **Compare Lifecycles** > **Compare With Selected** from the context menu. The project lifecycle will be compared with the lifecycle previously selected with the **Select For Left Compare** option.

Chapter

4

Lifecyle Comparison Editor

The Lifecycle Comparison Editor is split into two panes. The left-hand tree-view is used to navigate the lifecycle hierarchy of State, Process and Linked-Process elements. The right-hand list-view is used to view the properties associated with whatever element is selected in the tree-view.



- Project
  - Development
    - Check In Items
    - Check Out for Browse
    - Concurrent Merge
  - Test / Testing
  - Release
    - Compare Views

Property	Test Project	Test Project 2
CREATORNAME	harvest	harvest
DELETEAFTERCI	N	N
DESCRIPTION		
EXISTINGITEMONLY	Y	Y
FORCECOMMENT	N	N
ITEMNEWER	N	N
ITEMOPTION	Existing Items Only	Existing Items Only
MODEOPTION	Update And Release	Update And Release
MODIFIEDTIME	13/11/15 18:10	04/02/16 11:58
MODIFIERID	1	1
MODIFIERNAME	harvest	harvest
NEWITEMONLY	Y	Y
NEWOREXISTINGITEM	Y	Y
NOTE	This process is used to check in new	This process is used to check in new
ONBRANCH	**N**	**Y**
ONTRUNK	Y	Y
OWNERONLY	N	N

Element Tree Icons

The icons in the tree-view represent the Project, State and Processes that make up the lifecycle. These icons should already be familiar from Harvest Workbench. The icons are further decorated with additional markers to indicate differences, or items that appear on the left- or right-side only.

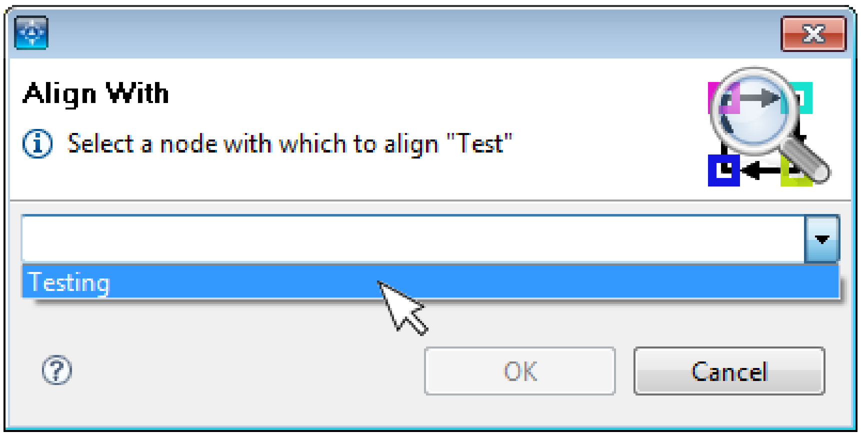
Filtering the Element Tree

By default, the tree-view shows only the changes between the two lifecycles. If an element has been changed it will appear with an additional decorator indicating the type of change (different, left-only or right-only). The parent elements of the changed element are also shown, even if they have not been changed, in which case they will have no decorator.

Sometimes it is desirable to see all of the elements in the tree, for example to see all of the processes in a given state. To do this, simply click the filter icon in the toolbar above the tree-view. Clicking the filter icon again returns to the changes-only view.

Aligning Elements in the Tree

The differencing algorithm compares elements at each level of a lifecycle with those at the same level and the same name. Where elements have been renamed, they show up as an addition on each side. For example, if one lifecycle contains a state named "Test" and the other contains a similar state named "Testing", you will initially see two states in the tree, one marked as left-only and the other as right-only. The two states can be aligned so that they can be compared. To do this, right-click on one of the elements you wish to align and select **Align With...**. In the dialog that opens, select the element that you want to align with (only elements at the same level are shown, and in the case of processes, only processes of the same type are shown). The lifecycles will be recompared and the tree-view will now show a single element containing both names seperated by a slash. For our states example, this would be "Test / Testing".


Align With

Select a node with which to align "Test"

Testing

The Properties List

The Properties List shows the properties for the currently selected element in the Element Tree. It is split into three columns, showing the property name and the left- and right-hand property values. Properties which have changed between the two lifecycles are highlighted in bold. Properties which have been excluded from the comparison are shown in grey (note that a number of properties are excluded by default, as they are always different between any two lifecycles, for example the object ids). Where a property appears on one side and not the other, it is decorated with an arrow similar to those in the Element Tree.

Value Properties

The majority of properties are simple values - either names or numbers. A small number of these value properties are enumerations, and where possible the values are displayed as an interpretted value (for example, the default path option for a check-in or check-out process is shown as "Preserve and Create" rather than as a number, which is how it is stored in the database).

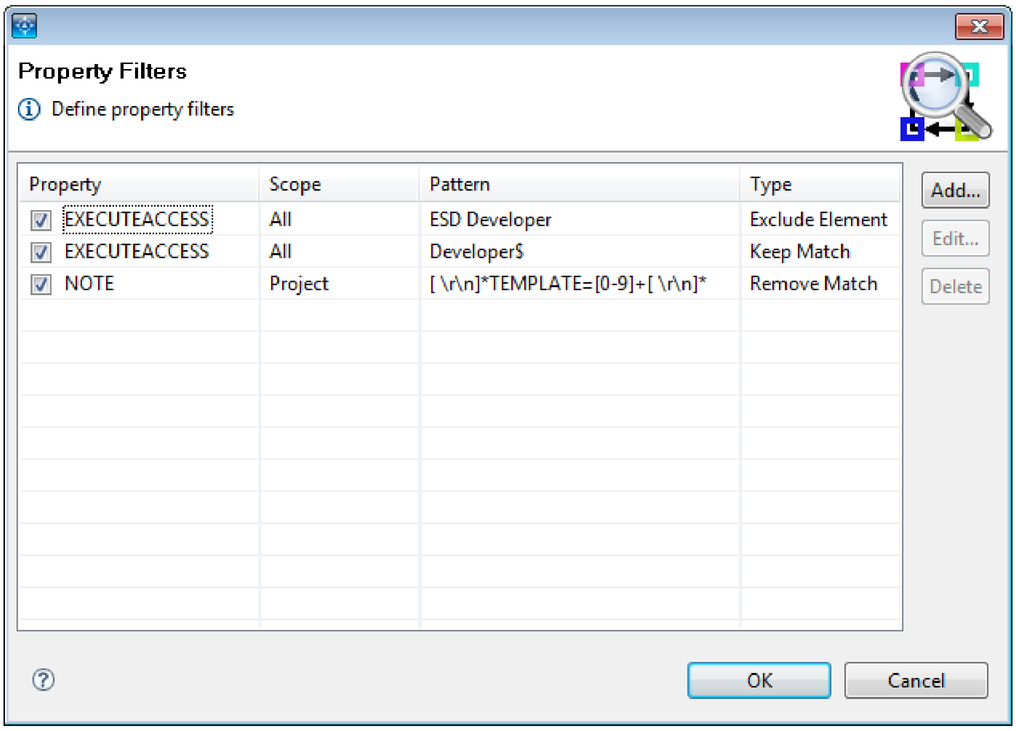
List Properties

Some properties are actually lists of values. These are mainly access-control lists, but there are also lists of recipients for notify processes and lists of approvers for approve processes. Lists are not compares as flat text - instead the individual entries in the list are compared. The list-view highlights individual entries that are different in red and entries that have been excluded in grey.

Chapter 5

Property Filters Dialog

The Property Filters Dialog allows you to add, edit and delete property filters.



Property Filters

Define property filters

Property	Scope	Pattern	Type
✓ EXECUTEACCESS	All	ESD Developer	Exclude Element
✓ EXECUTEACCESS	All	Developer$	Keep Match
✓ NOTE	Project	[\r\n]*TEMPLATE=[0-9]+[\r\n]*	Remove Match

Filters are defined per property and consist of a type, a pattern and a scope. There are three basic filter types:

Filter Type	Value Properties	List Properties
Keep Match	The pattern is applied to the value and only the matching portion is kept	The pattern is applied to each list entry in turn and only the matching portion is kept
Remove Match	The pattern is applied to the value and the matching portion is removed	The pattern is applied to each list entry in turn and the matching portion is removed
Exclude Element	If the pattern matches, the entire property is excluded from the comparison	If the pattern matches any of the list entries, that list element is excluded from the comparison

Filter Patterns

The pattern is a standard Java regular expression, as used with java.util.regex.Pattern class. Here are some examples:

Pattern	Description
Developer$	The value ends with "Developer". This can be used to either make "ABC Developer" equate to "DEF Developer" for project access, or can be used to remove all Developer groups from the comparison, depending upon which filter type is used
[\r\n]*TEMPLATE=[0-9]+[\r\n]*	Matches the complete line (including whitespace) used for defining a template value e.g. TEMPLATE=123. This would typically be used as a Remove Match filter to makes the Notes fields for a project and template equal if the only difference is the Babbage template definition line

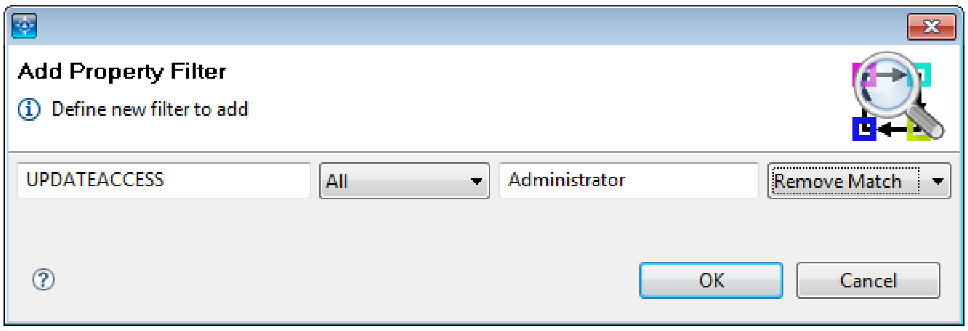
Filter Scopes

Scopes limit the types of elements to which the filter is applied. The following scopes are defined:

Scope	Description
All	The filter applies to all element types: projects, states, processes and linked processes
Project	The filter only applies to project elements
State	The filter only applies to state elements
Process	The filter only applies to process elements irrespective of the process type
Linked Process	The filter only applies to linked process elements irrespective of the process or parent-process type

Adding a Filter

To add a new filter, click the **Add...** button and enter the details of the new filter into the Add Filter dialog.


Add Property Filter
Define new filter to add
UPDATEACCESS
All
Administrator
Remove Match

Editing a Filter

To edit an existing filter, select it in the list and then click the **Edit...** button and make the changes you wish to make in the Edit Filter dialog.

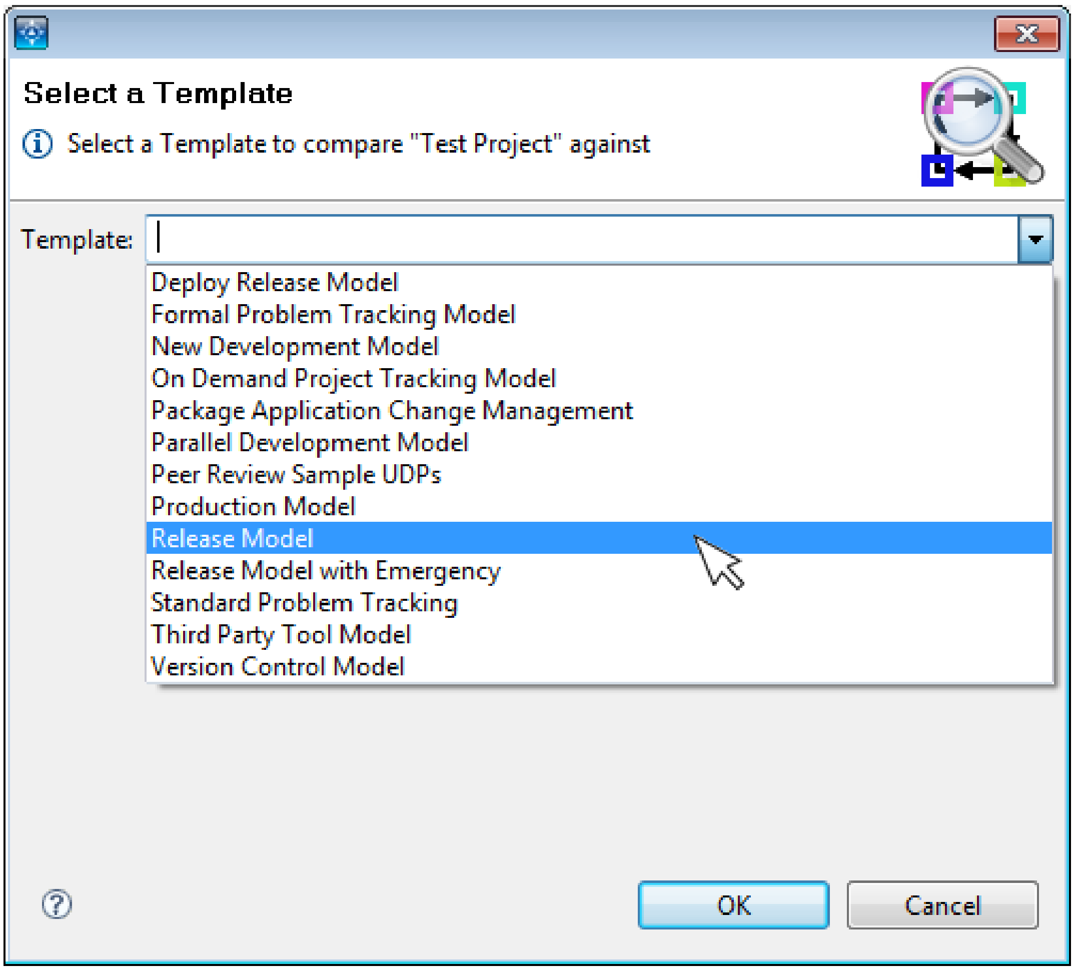
Removing a Filter

To remove a filter, select it in the list and then click the **Remove** button.

Chapter 6

Select Template Dialog

The Select Template Dialog allows you pick a template project to compare against.



Select a Template

Select a Template to compare "Test Project" against

Template:

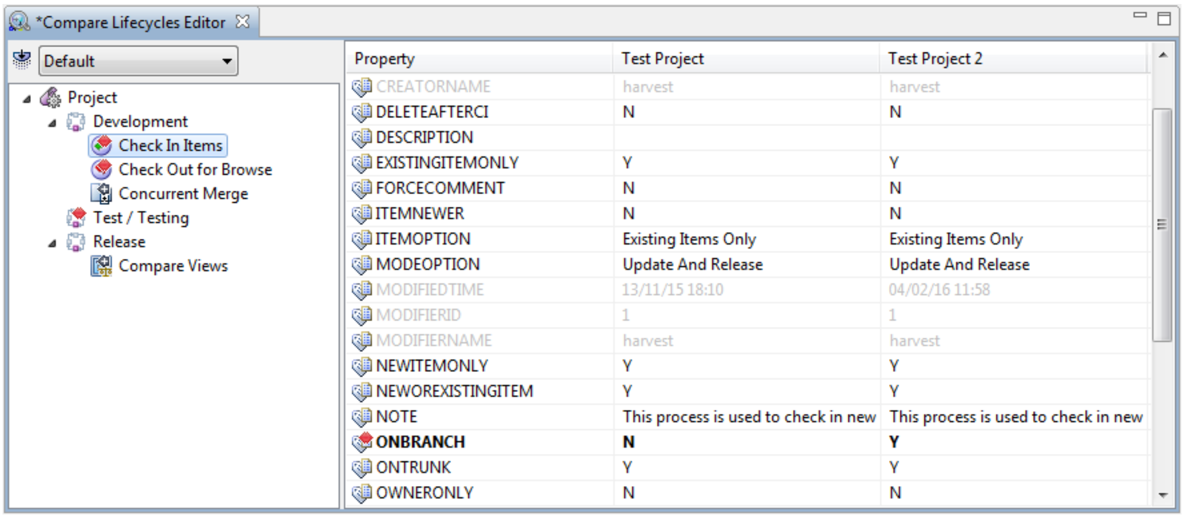
Deploy Release Model
Formal Problem Tracking Model
New Development Model
On Demand Project Tracking Model
Package Application Change Management
Parallel Development Model
Peer Review Sample UDPs
Production Model
Release Model
Release Model with Emergency
Standard Problem Tracking
Third Party Tool Model
Version Control Model

Chapter

7

Interpretting the Output

Babbage output should be familiar to those user who regularly use Harvest's synchronizer. Here is a simple example showing the differences between two closely related projects.



- Project
  - Development
    - Check In Items
    - Check Out for Browse
    - Concurrent Merge
  - Test / Testing
  - Release
    - Compare Views

Property	Test Project	Test Project 2
CREATORNAME	harvest	harvest
DELETEAFTERCI	N	N
DESCRIPTION		
EXISTINGITEMONLY	Y	Y
FORCECOMMENT	N	N
ITEMNEWER	N	N
ITEMOPTION	Existing Items Only	Existing Items Only
MODEOPTION	Update And Release	Update And Release
MODIFIEDTIME	13/11/15 18:10	04/02/16 11:58
MODIFIERID	1	1
MODIFIERNAME	harvest	harvest
NEWITEMONLY	Y	Y
NEWOREXISTINGITEM	Y	Y
NOTE	This process is used to check in new	This process is used to check in new
ONBRANCH	**N**	**Y**
ONTRUNK	Y	Y
OWNERONLY	N	N

The key differences that are indicated in the tree-view are as follows:

- The **Check In Items** and **Check Out for Browse** processes in the common **Development** state have different properties (indicated by the red double-headed arror). We will examine these differences later.
- The **Development** state of **Test Project 2** contains an additional **Concurrent Merge** process (indicated by the left-facing arrow with a plus symbol, showing the change would need to be merged to the left to make both lifecycles the same).
- The **Release** state of **Test Project 2** contains an additional **Compare View** process (again, marked with a left-facing arrow with a plus).
- Finally, the **Test** state has a different name in each lifecycle. In **Test Project** it is called "Test" and in **Test Project 2** it is called "Testing". The tree-view indicates that we have used the align-with option to compare the two states. As there are no child nodes, the processes in these two states are identical. The red double-headed arrow indicates there are differences between the state properties, in this case it is only the state name that is different.

Now, if you look the right-hand pane, you can see a comparison of the properties for the selected **Check In Items** process, which was marked as containing differences in the tree-view.

The only difference is the line highlighted in bold, showing that the **New Item on Branch** setting for the process is different between the two lifecycles. If you look at the actual values, you can see that **Test Project 2** allows new items to be checked in on the branch, whereas **Test Project** does not.

You can also see that a number of the properties are shown in grey. These are shown only for reference and their values are not used in the comparison. By default Babbage recognises and excludes certain properties which are normally different (for example object ids), but you can exclude properties yourself using the property filter mechanism.

Index