AllFusion Harvest Change Manager
Help Guide

Help and Reference Document

Developed for:
Ministry of Community Services
Ministry of Tourism, Sport and the Arts

Version 6.00
March 29, 2007

Revision Notes

Version	Date	Description	Document/ Change Owner
1.0	Jan 20, 2004	First copy for client review	Rodrigo Pena
2.0	Mar 29, 2004	Formatting changes, added missing content	Rodrigo Pena
2.1	April 13, 2004	Implemented changes based on 3rd client review	Rodrigo Pena
3.0	April 16, 2004		Maureen Bird
3.1	July 15, 2004	Updating headers	Maureen Bird
4.0	July 22, 2005	Remove List Differences between Views, List Versions; Add Set Default Context, Compare Files, Package Properties, Find features, change password, User roles	Maureen Bird
4.1	Sept 12, 2005	Added Harvest Folder Structure, Harvest Repository Structure, "Important" and "Tip" boxes and clarified where needed.	Maureen Bird
4.11	Nov 25, 2005	Removed Ministry of Agriculture & Lands reference on title page.	Maureen Bird
4.12	Dec 21, 2005	Change broker and DTS (VO) references – RFC_000394	Kerri Warnes
5.00	Apr 28, 2006	Added "create DOC package" section for BA's in SR - RFC_000589	Kerri Warnes
6.00	Mar 29, 2007	Update content to reflect changes as a result of move to Harvest release 7.1 (ref RFC_000640)	Maureen Bird

TABLE OF CONTENTS

1. Introduction

The use of AllFusion Harvest Change Manager is divided into administrative tasks and day-to-day tasks. The administrative tasks are performed primarily when first implementing Harvest, and are typically performed by someone designated as a Harvest Administrator. Such tasks include setting up the repository, defining the lifecycle to be used, setting up user groups, and setting access permissions.

After a Harvest project is set up, day-to-day operation is straightforward. Users are given access to information, provided a set of processes that can be performed, and notified of actions that have been or need to be taken.

This guide will focus only on end-user processes and will not discuss any administrative tasks.

2. Purpose

This document provides detailed steps on how to execute the most commonly used Harvest processes. The document does not cover every Harvest feature or process. Additional information can be found in CA's "AllFusion Harvest Change Manager User Guide". Please see your Harvest Administrator for details.

3. Audience

All users that participate in the different stages of the Ministry's SCM Methodology or System Development Life Cycle (SDLC) should use this as a quick reference.

4. Harvest Terminology

This section introduces the reader to Harvest terminology used throughout this document and within the Harvest application.

Project   The entire control framework within Harvest that supports a particular development or maintenance activity. For this Ministry, this is equivalent to an application.

Repository   Harvest stores and maintains items and their changes in a set of database tables called a *repository*.

State   A distinct work area in which certain activities can take place as packages move from identification to completion. For this Ministry, this is equivalent to a phase in the Ministry SDLC.

Package   A basic unit of work that moves through the life cycle; a single request for change. The Ministry has established three types of packages:

RFC (Request For Change) – used to record and track requests for change. The RFC package is created in the Initiate State.
DEF (Deficiency) and is used to record and track deficiencies coming out of User Acceptance Testing (UAT). The "Create DEF package" process is run in the UAT State, but the package itself (and the subsequent versions associated with fixing that Deficiency) is created in the Development state.
DOC (Documentation) – a single DOC package is included with each package group (or Release). Any documentation related to the entire release (e.g. Requirements or Design document) is checked in against the DOC package. The DOC package is created in the System Requirements State by the Business Analyst.

|	
Package Group	A group of packages (RFCs, DEFs (if they exist) and DOC package) that make up a Release. Package Groups are created and named by the Harvest Administrator following Ministry Oracle release naming standards.
Form	A Form contains information/details on activities related to a request. One or more forms are associated with a package.
Process	An action that can be executed on an object (e.g. package, item or package group) in Harvest. Processes in a given State determine what activities can be performed in that state.
Item	The actual data under Harvest's control. Sometimes referred to as a "file"; equivalent to a file on a file system.
Version	Each item in the Harvest repository consists of versions of the item. Version "0" is the base version of the item. The versions are representative of the changes made to an item (file).
Data View	A view into a project's repository that allows changes to be isolated and managed. Views (or Data Views) are integrated into a life cycle by associating them one or more states. Several states can share the same view, but a state cannot use more than one working view. Views determine which item versions are accessible to users.

🔔 Important Harvest note.   👍 Harvest tip or helpful hint.

5. Harvest Folder Structure

This section addresses the structure of the Harvest folders and where to find some of the items described in the Terminology section of this document.

The image below shows the structure of the Harvest folders. Below the image is a table describing each of the levels.
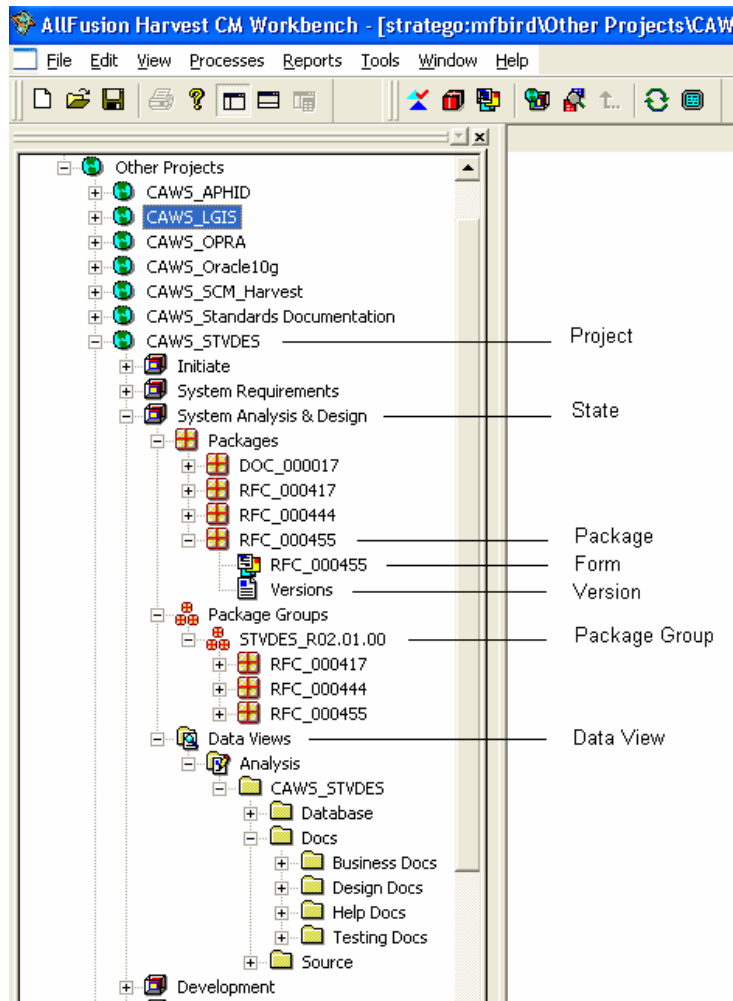
| Level / Node	Denoted by:	Description
Project		The top level of the Project. Expanded, it will list all the States in the Project.
State		There are 10 States in a standard Project. Expanding the State node provides access to Packages, Package Groups and Data Views.
Packages		Expanding the Packages node lists all of the packages currently in the selected State.
RFC_######		Expanding a package's node, displays the selected package's associated form and versions
Form		Displays the form associated with the selected package.
Versions		Displays the item versions associated with the selected package.
Package Groups		Expanding this node displays Package Groups in the selected State. Expanding a Package Group displays a list of packages that make up that group.
Data View		Expanding this node structure displays the repository structure and items/versions associated with the selected State's data view.

6. Harvest Repository Structure

The standard Harvest Repository Structure is shown below. This base structure should not be modified but additional folders can be created below these standard levels.
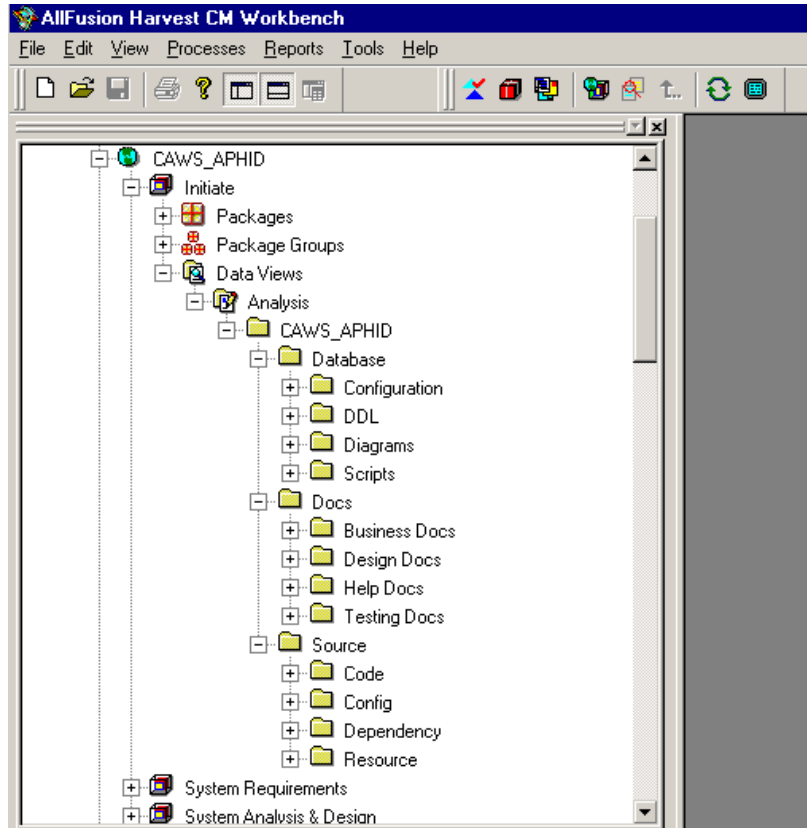
7. General Login and Harvest Context

7.1 The Harvest Login

From a PC:

1. Go to the **Start** menu. Go to **All Programs** | **Computer Associates** | **AllFusion** | **Harvest CM** | **Workbench**.
2. Enter the *Username* and *password* that you were given by the Harvest Administrator. (The Harvest username is patterned after your IDIR userid.)
3. In the **Broker** field, enter the name of the server where the Harvest Production broker is running (e.g. **stratego**). Please contact your Harvest Administrator for the name of the Production Broker.

> 🔔   Username, password and broker name are case-sensitive.

4. Click **OK**.
5. If desired, set your Harvest Context and click **OK**. If you don't want to set or change your Context, click **Cancel**. (See Setting the Default Context for more details.) You will see the Harvest window:
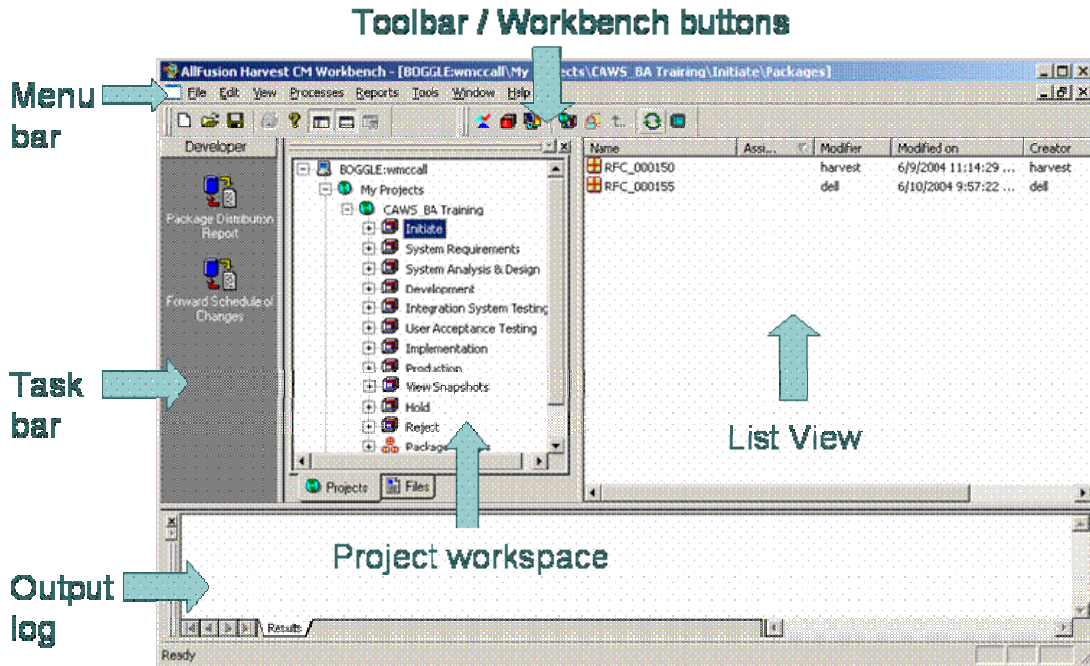
> 👍 To remove the Task Bar to allow more viewing space, go to **View** | **Task Bar** to toggle it off.

7.2 Setting the Default Context

7.2.1. Setting the Default Context at Harvest Login

At Harvest Login, you will be presented with the following *Default Context* window. Here you can establish the context of your Harvest activities, e.g. default Project and State or Check In/Out preferences and paths.

1. Select the **Project** you want to work in.
2. Select the **State** you want to work in.
3. If you are going to perform a Check Out or a Check In process it is recommended you set the View Path and the Client Folder you will Check Out to or Check In from. (Note: If you do not have privileges on the state that you selected, the View Path Icon will not be enabled.)

If you set a context, care should be taken when performing any action to ensure that the settings you have applied are still relevant.

For example: If you set a default client path, during check in/out ensure the path is still the one you wish to use. It could place the file in a location different from that intended.

Context settings remain in effect until you modify them. Once you set your context, it will be saved from one Harvest session to the next. Settings are stored locally so if you run Harvest from another PC you will need to reset your Context.

## 7.2.2.   Setting the Default Context from the Tools menu

From the Harvest Workbench, go to the **Tools** menu and select **Set Context**. The *Default Context* window will be presented. Follow steps 1-3 in Setting the Default Context at Harvest Login to set or change your context.

10

8. Harvest Processes

This section describes the most commonly used Harvest processes in the context of their use with the Ministry. Processes are listed in alphabetical order. (For a description of all the Harvest processes, please refer to CA's "AllFusion Harvest Change Manager User Guide".)

8.1 Approve Package

1. Make sure you are logged into the **Harvest Workbench** and you have selected the **Project** and **State** you wish to work in.
2. Expand the **State** node and then the **Packages** node to display all the packages currently in that state.
3. Expand the **Package** node to open the F**orm** associated with the package. Enter/update any fields on the form pertaining to your role and the state the package is in.
4. Click **OK**.
5. Right-click on the Package you wish to approve and choose **Processes** | **Approve Package** from the shortcut menu.
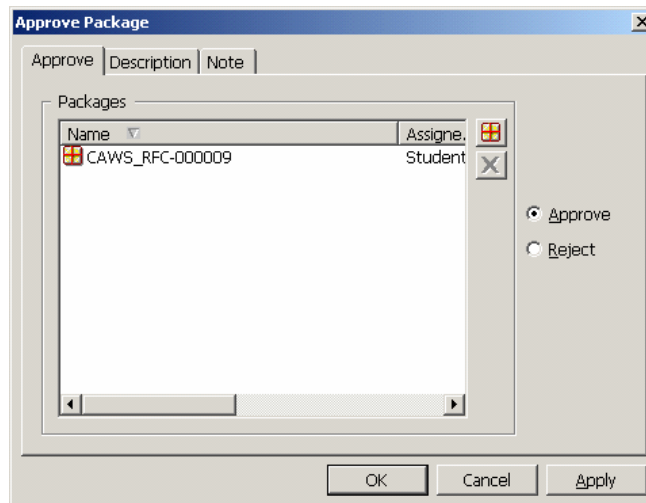
6. Select the **Approve** radio button.
7. Click **OK**.

> 🔔 Once you approve a Package it becomes "frozen" and no further updates can occur on the package. A Package can be "unfrozen" by Rejecting the Approval. Please see Rejecting a Package for details.

8.1.1. Rejecting a Package (Reversing an Approval)

1. Follow steps 1 to 5 in Approve Package above.

2. Select the **Reject** radio button.
3. Click **OK**.

> ☝ Once a Package is rejected, the user who performed the Rejection must re-Approve the Package before it can proceed through the Life Cycle.

8.1.2. Approving Multiple Packages

You can approve more than one package at a time.

1. Make sure you are logged into the **Harvest Workbench** and you have selected the **Project** and **State** you wish to work in.
2. Expand the **State** node and click on the **Packages** node. The list of packages in that State will be displayed in the List View.
3. Ensure all relevant information has been completed on the Package forms.
4. From the List View, Ctrl- or Shift-Click to select the packages you wish to approve.
5. Right-click on the selected packages and choose **Processes** | **Approve Package** from the shortcut menu.
6. Select the **Approve** radio button.
7. Click **OK**.

8.2 Change Password

1. From the Harvest Workbench's Menu bar, choose **Tools** | **Change Password**.

2. Enter your *old password*.
3. Enter your *new password*.
4. Re-enter your *new password* to confirm.
5. Click **OK**.

> ☝ If you have forgotten your Harvest password please contact the Harvest Administrator to have your password reset. Your previous Harvest password can not be retrieved.
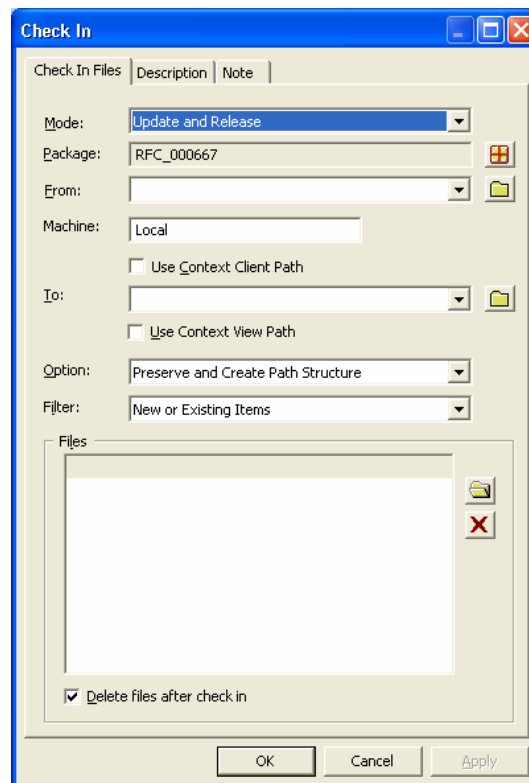
8.3   Check In (Package Level)

Use this Check In process if:

🔔   • You are on a PC and you want to check in new or
    previously checked out items

1. Make sure you are logged into the **Harvest Workbench** and you have selected the **Project** and **State** you wish to work in.
2. Expand the **State** node and then the **Packages** node to display all the packages currently in that State.
3. Right-click on the package you want to check items in against. Choose **Processes** | **Check In** from the shortcut menu.

    a. <u>If you are checking in a new file</u>:

        i. You will see the following dialog box:

👍   The 🗁 , 🗁 and 🗀 icons all give you access to Browse folders/directories.

        ii. Since you initiated the Check In from the Package level, the Package name is filled in for you.

13

iii. **From:** field - Specify where the client files are coming from.
- If you have set your Default Context and wish to use the file path specified there, check the *Use Context Client Path* option; or
- To specify the path, click the 📁 icon next to the **From** field. This opens a *Select a Directory Path* dialog that allows you to set the desired path; or
- Click the drop-down to get a list of most recently used paths.

iv. **To:** field - Specify where in the Repository the files will be checked in.
- If you have set your Default Context and wish to use the file path specified there, check the *Use Context View Path* option; or
- To specify the path, click the 📁 icon next to the **To** field. This opens the Select a Repository Item Path dialog that allows you to set the desired path; or
- Click the drop-down to get a list of most recently used paths.

v. **Options:** field - Provides different ways to check in and filter the files being checked in. Select one of the following:
- *Preserve Directory Structure* checks the selected files in to repository view paths with names that correspond to their client directory location, if these view paths currently exist.
- *Preserve and Create Path Structure* (default) checks selected files in to paths with names that correspond to their client directory location, and creates any view paths that do not currently exist.
- *All Files to Same View Path* checks in all selected files to the same path in the destination view, ignoring the client directory structure.

vi. Verify the **Filter** field is set to *New or Existing Items.*

vii. To select the items to be checked in, click on the 📁 button next to the Files section (see "Check In" screen shot above).
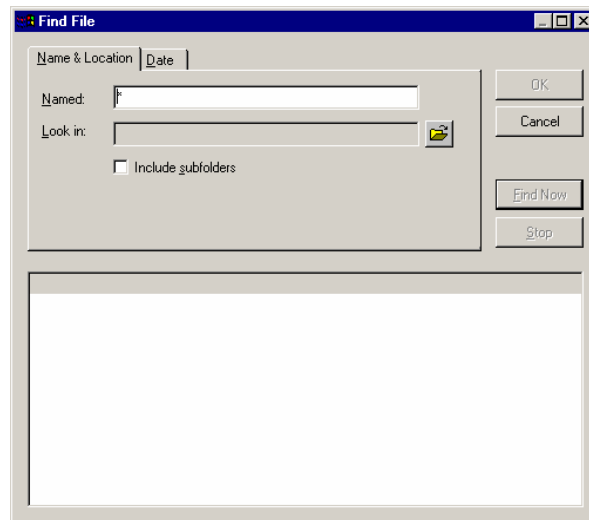

viii. Click the 📂 icon beside the **Look in** field and browse to the location of the file(s) you want to check in.

14

ix. Check the *Include subfolders* option if you want to browse directories below the one you're looking in.

x. Click **Find Now** to get a file list. Select the file(s) you want to check in.

xi. Click **OK**. You will be brought back to the previous screen where you can verify all the items you are checking in and their respective locations in the repository.

xii. Leave the *Delete Files after Check In* option "checked". This will ensure that multiple copies of the file(s) don't exist and that the version in Harvest is the "master".

> If the "Delete files after check in" option is <u>not</u> selected, a copy of the file is left in the client folder and the file is automatically set to read-only so that changes cannot be made to it until it is checked out again.

xiii. Click **OK**.

xiv. Confirm that the file(s) have been checked in by expanding the Packages node, clicking on **Versions** and checking the files displayed.
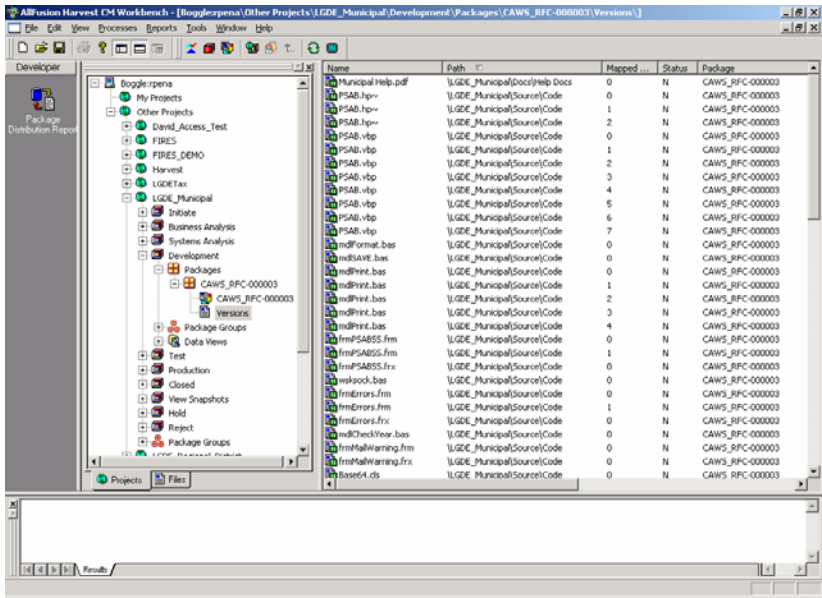
> Each package has a Versions node which will display all the items/versions associated with that package.

b. <u>If you are checking in an item that is currently checked out (reserved)</u>:

i. Items will be listed in the Versions list box because they were associated with the package when you checked them out.

Check In

Check In Reserved | Check In Files | Description | Note

Mode: Update and Release

Option: Preserve and Create Path Structure

Versions

Name ▽	Client Location
\Training Repository\SR...	H:\A_work_area\CAWS_DEF

☑ Delete files after check in

OK   Cancel   Apply
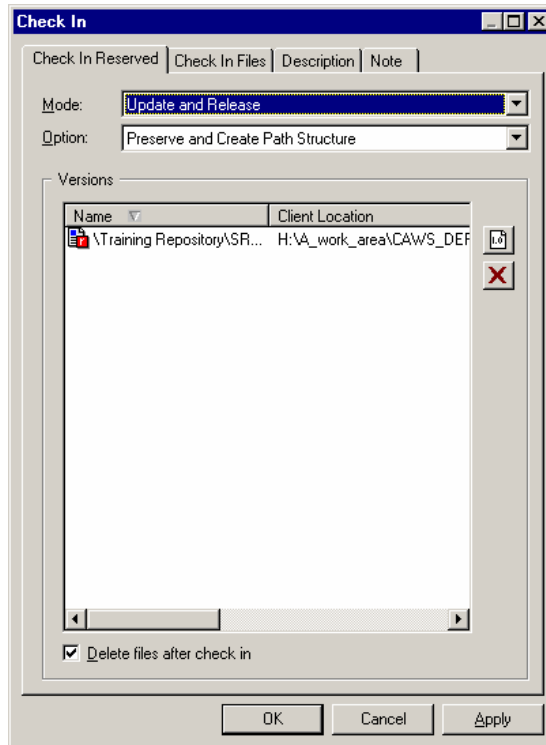
> ☞ If you wish to check in a new item and you already have another item checked out against the same package, you can check in the new item and leave the reserved one as is. Use the Check In Files tab of the dialog and follow the instructions for checking in a new file (Step (a) above).

    ii. Choose **Update and Release** from the **Mode** drop-down list. The *Update and Release* option creates a new version and removes the lock from the version so it can be checked out again later.
    iii. Click **OK**.

4. Confirm that the file(s) have been checked in by checking the Output Log or by expanding the Packages node, clicking on **Versions** and checking the files displayed.

> ☞ Each package has a Versions node which will display all the files associated with that package.

8.4   Check In (State Level)

> 🔔 Use this Check In process if:
> - You are on a PC and you want to check in new or previously checked out items; OR

> - You are on a PC and you want to check in an item that was checked out from a Harvest session at a different PC.
>
>   **Note**: If you need to check in an item that you had originally checked out to another location, please see your Harvest Administrator for details.

1. Make sure you are logged into the **Harvest Workbench** and you have selected the **Project** and **State** you wish to work in.
2. Right-click on the **State** and choose **Processes** | **Check In** from the shortcut menu.
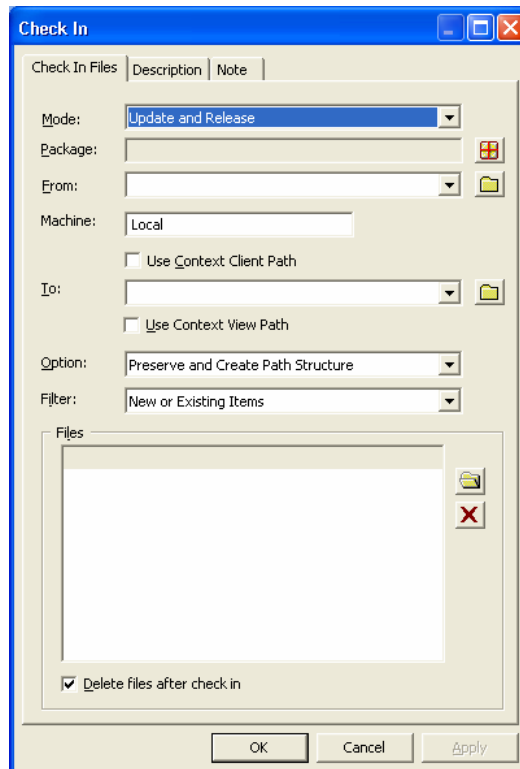
> ☞ The ⊡, ☞ and ☐ icons all give you access to Browse folders/directories.

3. Confirm that the **Mode** is set to **Update and Release**. This will check the item or version into the repository and, if the item was reserved by a package during a previous check out, the reserved tag (red "**r**") is removed from the item.

> 🔔 If the *Delete files after check in* option is <u>not</u> selected, a copy of the file is left in the client folder and the file is automatically set to read-only so that changes cannot be made to it until it is checked out again.

Other available modes are:

- *Update and Keep*—The item in the view is updated (or created) and the current package keeps it reserved, and the file permissions unchanged, so that more updates can be made.
- *Release Only*—No check in is performed and the item is not updated, but the item is no longer marked as reserved for the current package. The permission on the client file is automatically set to read-only so that the file cannot be changed until it is checked out again. NOTE: From a State-level Check In the Release Only mode requires the local file system file to exist. If the file does not exist on the local file system, you must perform the Check In from the Package or Version level.

4. Any updates made to an item are associated with the package selected in the **Package** field. You <u>must</u> select a Package you want to check in the files against.

5. Click the ⊞ button next to the Package field (see screen shot below Step 2). Select the package you're checking in against and click **OK**.
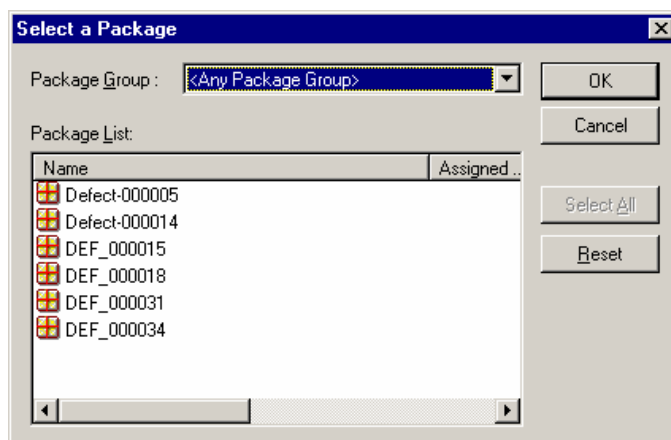
6. **From:** field - Specify where the client files are coming from.
   - If you have set your Default Context and wish to use the file path specified there, check the *Use Context Client Path* option; or
   - To specify the path, click the 📁 icon next to the **From** field. This opens a *Select a Directory Path* dialog that allows you to set the desired path; or
   - Click the drop-down to get a list of most recently used paths.

7. **To:** field - Specify where in the Repository the files will be checked in.
   - If you have set your Default Context and wish to use the file path specified there, check the *Use Context View Path* option; or
   - To specify the path, click the 📁 icon next to the **To** field. This opens the Select a Repository Item Path dialog that allows you to set the desired path; or
   - Click the drop-down to get a list of most recently used paths.

8. **Options:** field - Provides different ways to check in and filter the files being checked in. Select one of the following:

- *Preserve Directory Structure* checks the selected files in to repository view paths with names that correspond to their client directory location, if these view paths currently exist.
- *Preserve and Create Path Structure* (default) checks selected files in to paths with names that correspond to their client directory location, and creates any view paths that do not currently exist.
- *All Files to Same View Path* checks in all selected files to the same path in the destination view, ignoring the client directory structure.

9. Verify the **Filter** field is set to **New or Existing Items**.
10. To select the items, click on the 🗁 button next to the Files section (see screen shot below Step 2).
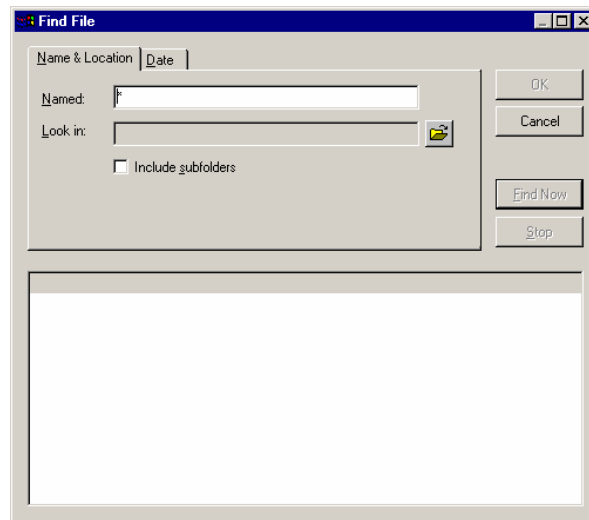
11. Click the 📂 icon beside the **Look in** field and browse to the location of the file(s) you want to check in.
12. Check the *Include subfolders* option if you want to browse directories below the one you're looking in.
13. Click **Find Now** to get a list of files. Select the file(s) you want to check in. (You can use Ctrl- or Shift-click to select multiple files.)
14. Click **OK**. You will be brought back to the previous screen where you can verify all the items you are checking in and their respective locations in the repository.
15. Leave the *Delete Files after Check In* option "checked". This will ensure that multiple copies of the file(s) don't exist and that the version in Harvest is the "master".
16. Click **OK**.

☞ If you forgot to select one of the criteria necessary for the check-in, the following reminder will appear:

Harvest Error

Please make sure you have selected a package, a From path, a To path and one or more files before executing this process.
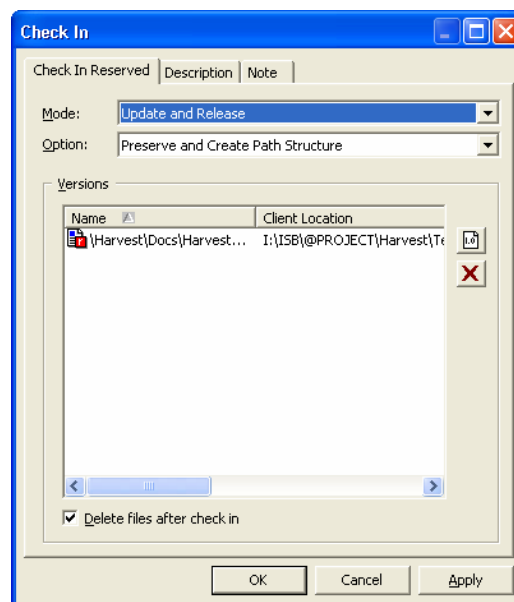
[OK]

17. Check the Harvest output log to confirm that the file(s) have been checked in. You can also do this confirmation by expanding the Packages node, clicking on **Versions** and checking that the files displayed.

8.5   Check In (Version Level)

> 🔔 • Use this Check In process if you want to check in a previously checked out items from the Version level

1. Make sure you are logged into the **Harvest Workbench** and you have selected the **Project** and **State** you wish to work in.
2. Expand the **Package** associated with the version then click on **Versions** to see a list of versions associated with the Package.
3. You should see a version with a red "**r**" or "reserved" tag on it. Right-click on the version then click Processes | Check In.

4. Confirm that the Mode is set to Update and Release. This will check the item or version into the repository and, if the item was reserved by a package during a previous check out, the reserved tag (red "**r**") is removed from the item.

> 🔔 If the *Delete files after check in* option is **not** selected, a copy of the file is left in the client folder and the file is automatically set

20

> to read-only so that changes cannot be made to it until it is
> checked out again.

Other available modes are:
- *Update and Keep*—The item in the view is updated (or created) and the current package keeps it reserved, and the file permissions unchanged, so that more updates can be made.
- *Release Only*—No check in is performed and the item is not updated, but the item is no longer marked as reserved for the current package. The permission on the client file is automatically set to read-only so that the file cannot be changed until it is checked out again. Check In for Release Only does not require the local file system file to exist. If the file does not exist on the local file system, you need to right-click the reserved version and then choose the check in process.

5. Options: field - Provides different ways to check in and filter the files being checked in. Select one of the following:
- *Preserve Directory Structure* checks the selected files in to repository view paths with names that correspond to their client directory location, if these view paths currently exist.
- *Preserve and Create Path Structure* (default) checks selected files in to paths with names that correspond to their client directory location, and creates any view paths that do not currently exist.
- *All Files to Same View Path* checks in all selected files to the same path in the destination view, ignoring the client directory structure.

6. Leave the Delete Files after Check In option "checked". This will ensure that multiple copies of the file(s) don't exist and that the version in Harvest is the "master".
7. Click OK.
8. Check the Harvest output log to confirm that the file(s) have been checked in. You can also do this confirmation by expanding the Packages node, clicking on Versions and checking that the files displayed.

8.6   Checking Package Properties

1. Make sure you are logged into the **Harvest Workbench** and you have selected the **Project** and **State** you wish to work in.
2. Expand the **State** node and then the **Packages** node to display all the packages currently in that State.
3. Right-click on the **Package** for which you want to view the properties and choose **Properties**.
4. You will see seven tabs to the Properties window:

a. Package: Provides detail on the package name, who it is assigned to, project, state, creation and update dates and times.
b. Approval: Provides you with the status of package approvals (Need/Approved/Rejected) for each of the roles responsible for approval in this state.
c. Groups: Tells you which Package Group/Release this package belongs to, if any.
d. Forms: Tells you which form is associated with the selected package plus details on when it was created and last modified and by whom.
e. History: Provides details on the history of processes run against this package, e.g. created, approved, promoted. It includes action taken, when it was taken, by whom and in which Project and State. It does not provide detail on Check In/Out processes.
f. Description: Descriptions associated with this package may appear here. The description typically reflects the contents of the Short Description field.
g. Notes: Notes associated with this package may appear here.

8.7   Check Out for Browse

1. Make sure you are logged into the **Harvest Workbench** and you have selected the **Project** and **State** you wish to work in.
2. Expand the **Packages** node to display all the packages currently in that state.
3. Right-click on the **Package** you wish to **Check Out** from then choose **Processes** | **Check Out for Browse**.
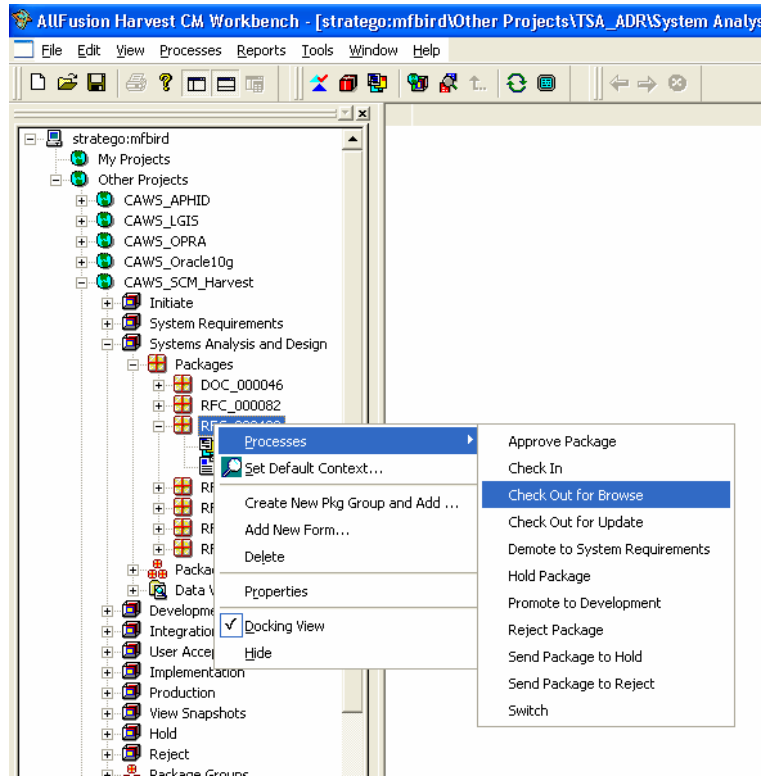
4. Confirm the **Mode** is set to **Browse**.
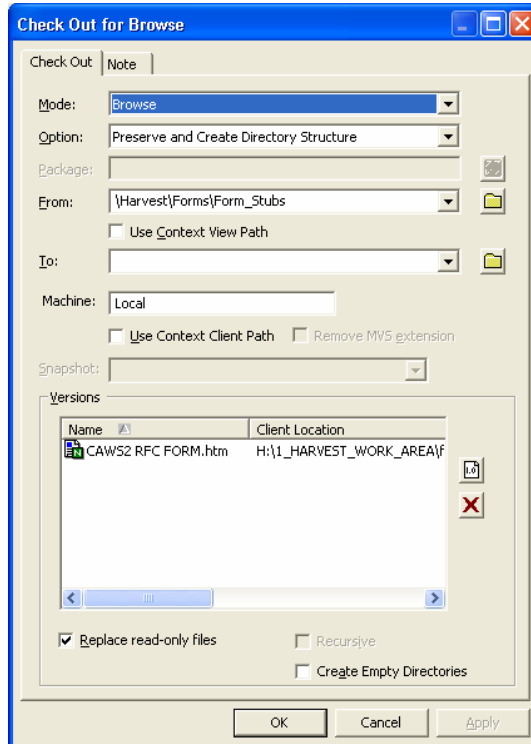
23

5. Verify the Option field is set to be Preserve and Create Directory Structure. This will create a mirror image of the directory structure where the item resides within the repository on your local file system.
6. Confirm that the **Mode** is set to **Browse**.
7. Verify the Option field is set to be Preserve and Create Directory Structure. This will create a mirror image of the directory structure where the item resides within the repository on your local file system.
8. The **From** field lets you set the repository path you want to check items from within Harvest.
    - If you have set your Default Context and wish to use the file path specified there, check the *Use Context Client Path* option; or
    - To specify the path, click the ⬛ icon next to the **From** field. This opens a *Select a Directory Path* dialog that allows you to set the desired path; or
    - Click the drop-down to get a list of most recently used paths.

9. The **To** field lets you set the file system path you want to check items out to.
    - If you have set your Default Context and wish to use the file path specified there, check the Use Context View Path option; or
    - To specify the path, click the ⬛ icon next to the To field. This opens the Select a Repository Item Path dialog that allows you to set the desired path; or
    - Click the drop-down to get a list of most recently used paths.

10. You now need to select the items you want to check out. Click on the ⬛ icon to the right of the Versions section of the dialog.
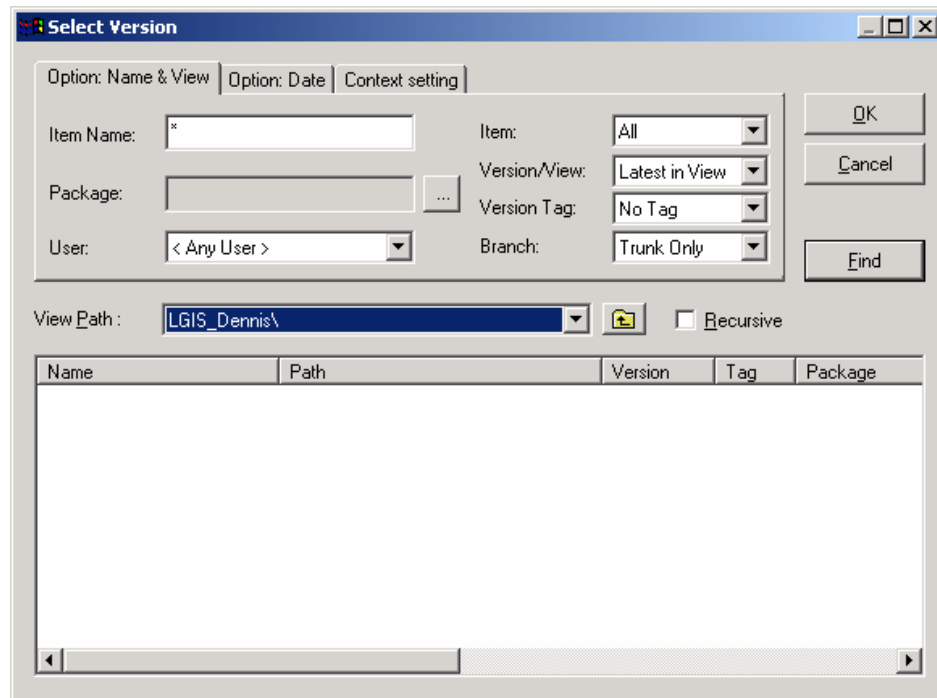
11. Set your selection criteria, ensuring that you check your Context Setting (3rd tab) and select the **Recursive** check box.

12. Click the **Find** button. This will display all the items matching the search criteria. By default all the items on the list will be selected. To deselect the ones you don't want to check out hold the control key and click on the item.
13. Click the **OK** button. This will bring you back to the *Check Out for Browse* dialog. Verify these are the files you want to check out.
14. Confirm that the **Replace read-only files** option is checked.
15. Click **OK**.

> If you forget to select a target or source location or a version, the following reminder will appear:
>
>
>
> Harvest Error
>
> Please make sure you have selected a from path, a To path, and one or more versions before executing this process.
>
> OK

> You can also perform a Check Out for Browse from the State and Data View levels. The only difference when checking out from the Data View level is that the "From" field is already filled in since it knows where the item is coming from.

> An "Error List" dialog will appear if an error occurs during the check out process. It allows you to return to the Check Out dialog without closing the Error list, fix the errors and run the process again, all without closing the Check Out dialog or having to check the Output log.

8.8   Check Out for Update

1. Make sure you are logged into the **Harvest Workbench** and you have selected the **Project** and **State** you wish to work in.
2. Expand the **Packages** node to display all the packages currently in that state.
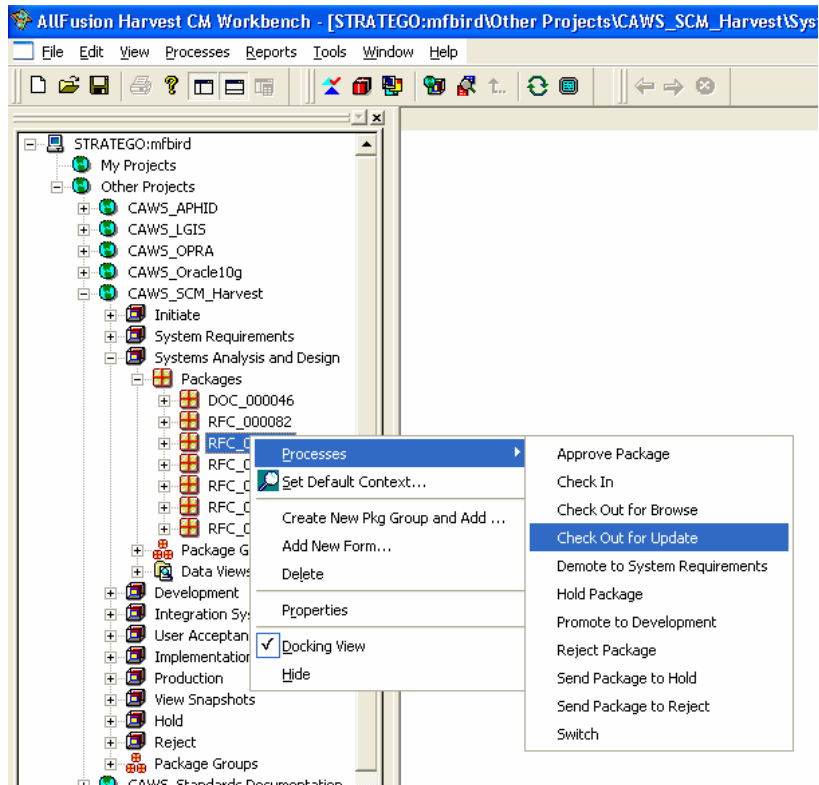
3. All changes to items must be associated with a package. As a package moves through the lifecycle it brings the associated changes with it. Right-click on the desired package and choose **Processes** | **Check Out for Update**.
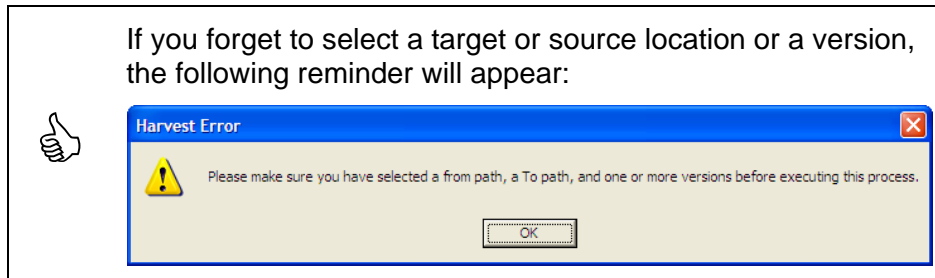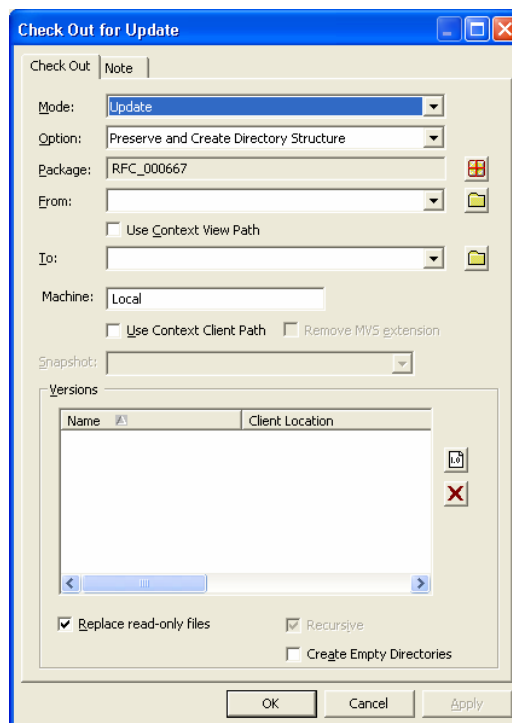
4. Confirm that the **Mode** is set to **Update**.
5. Verify the **Option** field is set to be **Preserve and Create Directory Structure**. This will create a mirror image of the directory structure where the item resides within the repository on your local file system.
6. Ensure that the package you are working on is shown in the **Package** field. If it is not, click the button next to the field to locate and select it.
7. The **From** field lets you set the repository path you want to check items from within Harvest.
   - If you have set your Default Context and wish to use the file path specified there, check the Use Context Client Path option; or
   - To specify the path, click the icon next to the From field. This opens a Select a Directory Path dialog that allows you to set the desired path; or
   - Click the drop-down to get a list of most recently used paths.
8. The **To** field lets you set the file system path you want to check items out to.
   - If you have set your Default Context and wish to use the file path specified there, check the Use Context View Path option; or
   - To specify the path, click the icon next to the To field. This opens the Select a Repository Item Path dialog that allows you to set the desired path; or
   - Click the drop-down to get a list of most recently used paths.

   NOTE: If you check off the "Use Context View Path" option in the above examples and you do not have a path set in your Context Settings you will see the following warning:
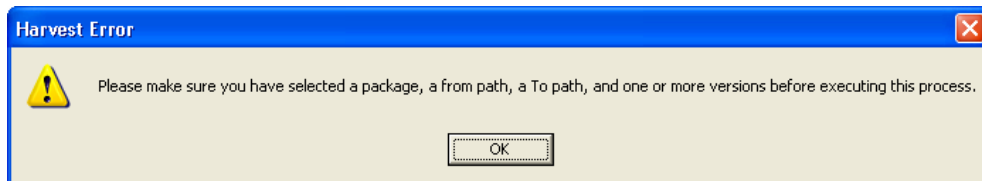
9. You now need to select the items you want to check out. Click on the icon to the right of the Versions section of the dialog.
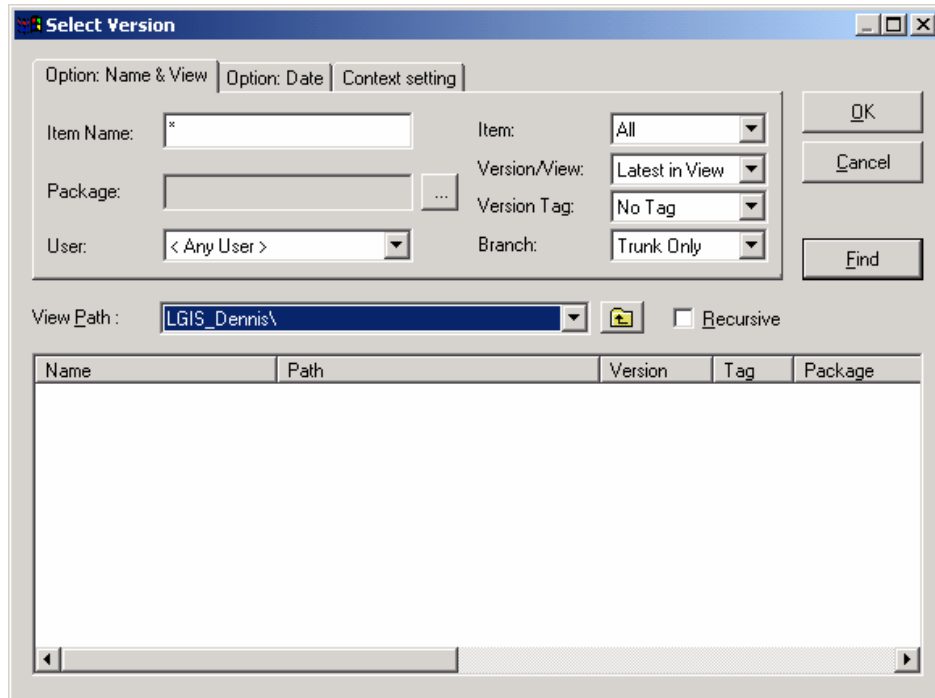
10. Set your selection criteria, ensuring that you check your Context Setting (3rd tab) and select the **Recursive** check box.
11. Click the **Find** button. This will display all the items matching the search criteria. By default all the items on the list will be selected. To **de**-select the ones you don't want to check out hold the control key and click on the item.
12. Click the **OK** button. This will bring you back to the *Check Out for Update* dialog. Verify these are the files you want to check out.
13. Confirm that the **Replace read-only files** option is checked.
14. Click **OK**.

If you:

- forgot to select a target location, source location, or a version; OR

- selected a "Use Context View Path" option but didn't set the View Path in your Context Settings

you will see the following warning:

15. On the data tree, expand the Package you are working on then click on **Versions**. Here you can see the items you have checked out. Notice the red "**r**" tag – this indicates that the items are now reserved in Harvest and you need to check them in or release them before anyone else can check them out.

You can also do a Check Out for Update from the State level or from the Data View. In each of these cases, you <u>must</u> select the Package you wish to Check Out against or the Check Out process will fail.

An "Error List" dialog will appear if an error occurs during the check out process. It allows you to return to the Check Out dialog without closing the Error list, fix the errors and run the process again, all without closing the Check Out dialog or having to check the Output log.

8.9   Compare Files/Versions

The Compare Files/Versions feature allows you to compare the contents of two files or versions. The feature uses Harvest's Visual Difference report to display a side-by-side, line-by-line comparison of two files.

You can use this feature to compare non-binary files from any file list. For example, you can call up Windows Explorer or MS Word, pull a non-binary file from its file lists and compare it to one in the Harvest Repository. Binary files (e.g. Word, Excel) <u>cannot</u> be viewed with this tool. If you wish to compare two Word documents, you must use Word's Compare feature.

For this example, a file in the Harvest Repository will be compared with a file on the user's file share

1. From the Harvest Workbench, go to the Harvest Toolbar. Click the **Compare Files/Versions** button .   The following *Compare Files* dialog will appear.

```
┌─────────────────────────────────────────┐
│ Compare Files                        ▓▓│
├─────────────────────────────────────────┤
│  Please use Drag and Drop to select files/versions. │
│                                          │
│  Left File:   [                      ]   │
│                                          │
│  Right File:  [                      ]   │
│                                          │
│              [  Compare  ]  [  Cancel  ] │
└─────────────────────────────────────────┘
```
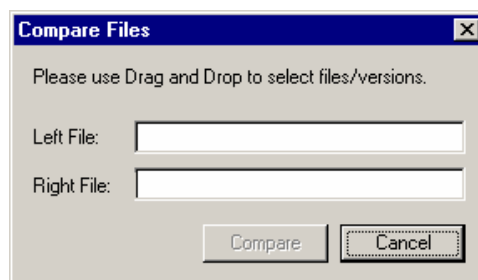
2. Open the file list(s) that contain the files you wish to compare. In our example, go to the data view list and the listing from the file share.
3. Drag and drop the first file from the data view list to the "Left File:" field.

4. Drag and drop the file from the file share into the "Right File:" field. In the example below, *CAWS.ini* version 3 (Left field) from the Harvest Repository is being compared to *CAWS.ini* from the file share (Right field).
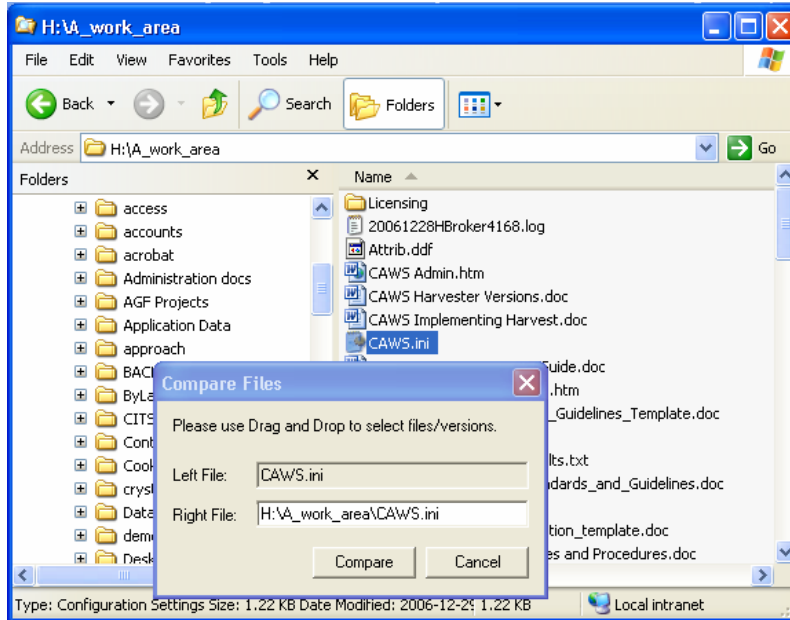
5. Click **Compare**.
6. A *vdiff* (or *Visual Differences*) screen will appear, similar to that below. Scroll through the files on each side to view the differences between them. Blue text indicates text that has been deleted, red text indicates text that has been changed and green text indicates text that has been added.
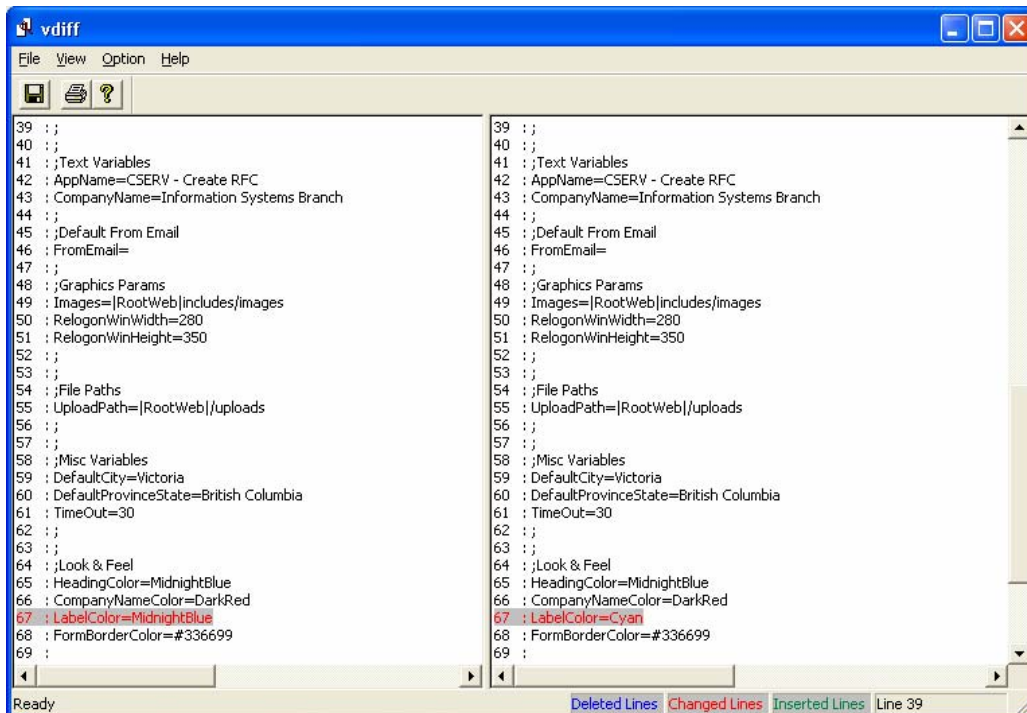
	When the files are viewed using this feature, the files are read-only and cannot be modified.

8.10 Create DEF (Deficiency Package)

1. Make sure you are logged into the **Harvest Workbench** and you have selected the **Project** you wish to work in.
2. Select the **User Acceptance Testing State**. (Note: You can only create Deficiency packages from the UAT State.)
3. Right-click on the state and choose **Processes** | **Create Deficiency Package** from the shortcut menu.

4. The **DEF** package name will be generated automatically. Click **OK**.
5. The new DEF package is created and you will be left at the Details tab of the DEF form.

	While the process to create a Deficiency package is initiated in the UAT State, the actual DEF package is created in the Development State. Developers are sent an email notification when this occurs so they can begin fixing the deficiency.

6. Fill in the fields on the Details tab of the form. If there is supporting documentation you wish to attach to the form, please see **Attachments** in Create RFC (Harvest Workbench).
7. Click **OK** to save your changes

8.11 Create DOC (Harvest Workbench)

Note: A DOC Package is one created to track all documentation relevant to an entire Release (Package Group). (e.g. Business Requirements, Detailed Design, Test Documents, etc.) Each Release in a Harvest Project should contain a DOC Package. The DOC package moves through the Project lifecycle with the other RFCs in the Release.

1. Make sure you are logged into the **Harvest Workbench** and you have selected the **Project** you wish to work in.
2. Select the **System Requirements State**. (Note: You can only create DOC Packages in the System Requirements State.)
3. Right-click on the state and choose **Processes** | **Create Document Package** from the shortcut menu.

4. The DOC name will be generated automatically. You can choose to compete the "Assign To:" information.
5. Click OK.
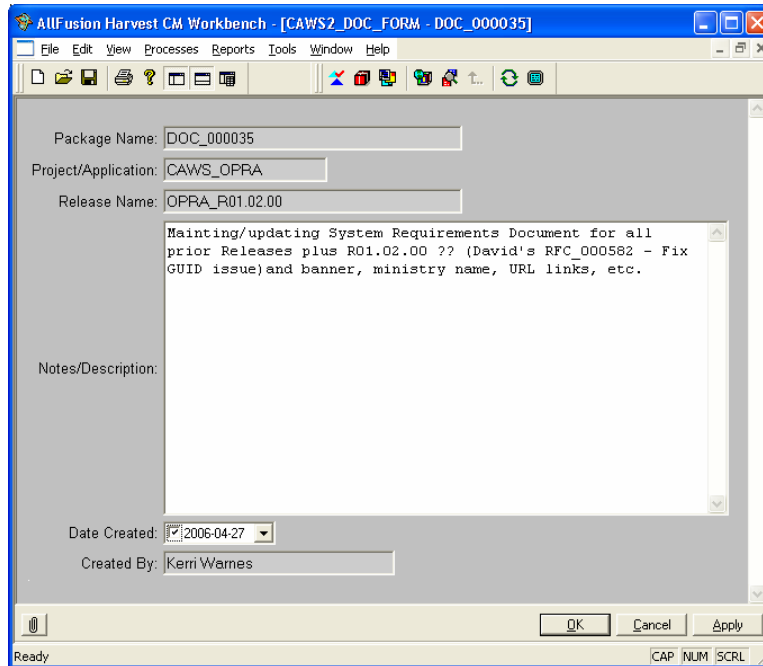
6. Fill in the "Notes/Description:" field on the form. If there is supporting documentation you wish to attach to the form, please see **Attachments** below.
7. Click OK to save your changes.

8.12 Create RFC (Harvest Workbench)

1. Make sure you are logged into the **Harvest Workbench** and you have selected the **Project** you wish to work in.
2. Select the **Initiate State**. (Note: You can only create RFC Packages in the Initiate State.)
3. Right-click on the state and choose **Processes** | **Create RFC** from the shortcut menu.

33
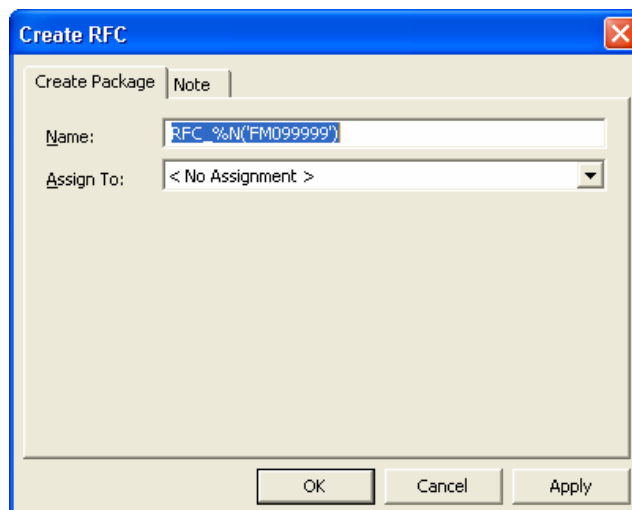
4. The RFC name will be generated automatically. Click **OK**.
5. The new RFC package is created and you will see the Details tab of the RFC form.
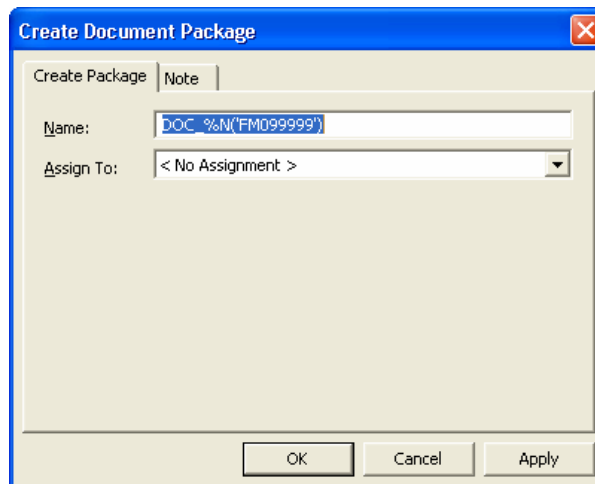6. Fill in the fields on the Details tab of the form. If there is supporting documentation you wish to attach to the form, please see **Attachments** below.
7. Click **OK** to save your changes

8.12.1. Attachments

Attachments should only include information that does not need to be versioned. If versioning is required, please use the **Check In** process.

1. Expand the **Package** then click on the icon to view the package's associated form.

2. Click on the icon in the bottom left hand corner of the pane.
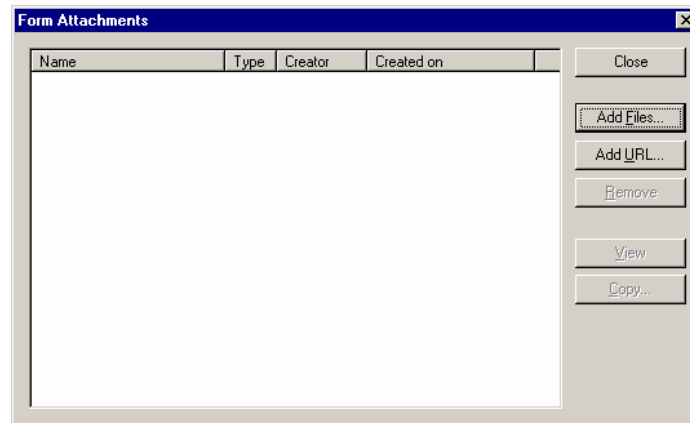
3. To add a File:
   a. Click on **Add Files…**

34
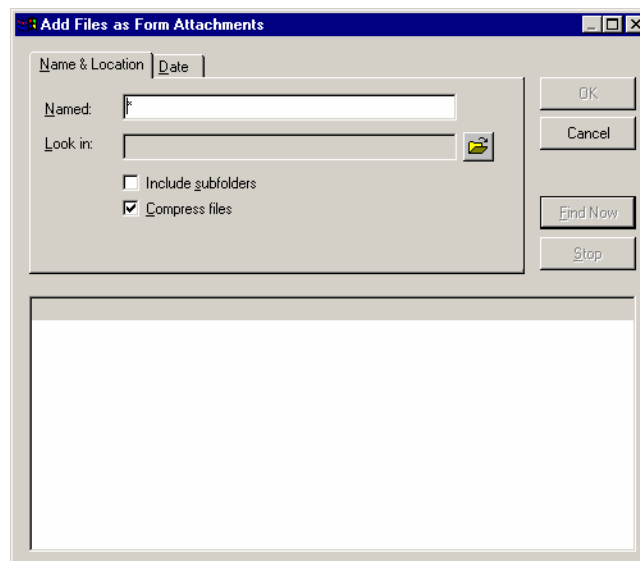
b. The following reminder appears. Click OK leaves you to find and select the files you wish to attach.
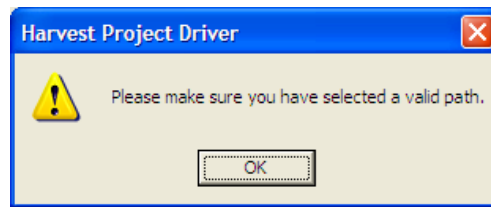
c. Use the "Named" and "Look In" field to search for the file(s) to attach.
d. Click **Find Now**.
e. Select the file(s) you want to attach from the list displayed and click OK.

4. To add a URL:
   a. Click on Add URL
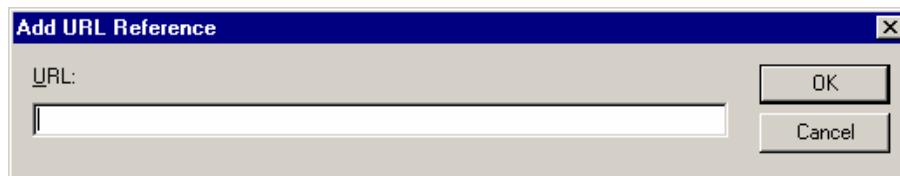
b. Type in the URL.
c. Click OK.

5. Verify the file(s) and/or URL(s) you have selected.
6. Click **OK**.

8.13 Demote Process

The Demote process is used to demote a package to the previous state in the life cycle.

1. Make sure you are logged into the **Harvest Workbench** and you have selected the **Project** and **State** you wish to work in.
2. Expand the **State** node and then the **Packages** node to display all the packages currently in that state.
3. Right-click on the package you wish to demote and choose **Processes** | **Demote** from the shortcuts menu.
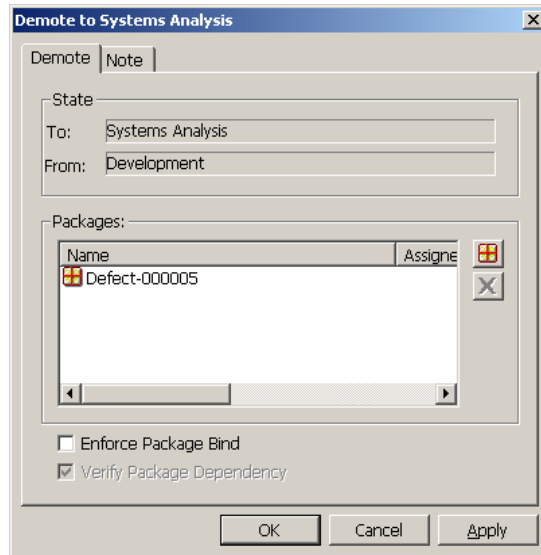
4. Click **OK** to confirm that you want to demote the package. Notice that the package no longer exists in the current state.

8.13.1. Demoting Multiple Packages

You can demote more than one package at a time.

1. Make sure you are logged into the **Harvest Workbench** and you have selected the **Project** and **State** you wish to work in.
2. Expand the **State** node and click on the **Packages** node.
3. Ctrl- or Shift-Click to select the **packages** you wish to approve.
4. Right-click on the selected packages and choose **Processes** | **Demote to <Previous State>** from the shortcut menu. The *Demote to <Previous State>* dialog will appear with selected packages listed.
5. Click **OK**.

8.14 Find Package

The **Find Package** feature allows you to search for a package based on a number of criteria. After locating the package(s) you can view any associated forms, execute available processes directly from the found list, view the package properties, etc.

1. To access the **Find Package** feature, go to **Tools** | **Find Package** or click on the **Find Package** icon from the Harvest Toolbar.
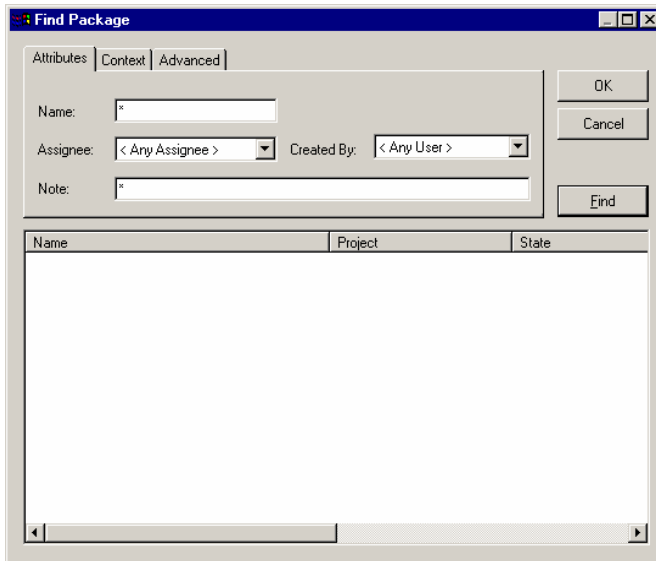
2. Use the Attributes, Context and Advanced tabs to refine your search.
    a. Attributes tab (see dialog in Step 1):
        i. Name: Use this field to search by Package name. You can use wildcards (*) for multiple character matching, question marks (?) for single character matching.
        ii. Assignee: Allows you to search for packages assigned to a user. Use the drop-down list to select the user.
        iii. Created by: Allows you to search for packages created by a specific user. Use the drop-down list to select the user.
        iv. Note: Allows you to search for a Package Properties' Note tab for text strings. You cannot do a wildcard (*) search in this field.
        (Example: A Note reads "This is the note for RFC_000123.". You can search for **123** or **RFC** or **This is the note**. The search is case-sensitive so in this example if you typed **rfc** in your search criteria it would not find this package.)
    b. Context tab:
        On this tab you will set the context of your search, i.e. set the Project, State and/or Package Group by which you wish to restrict your search.
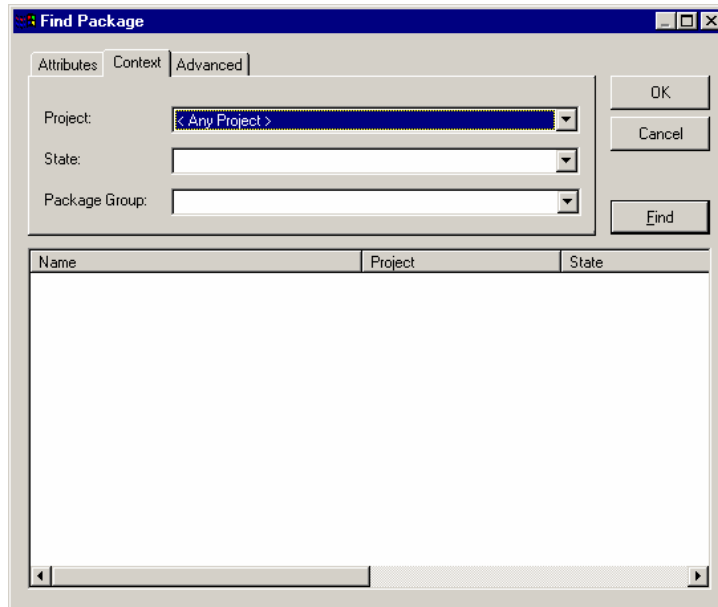
i.   <u>Project</u>: Use the drop-down list to select the Project
      you want to search in. Use <Any Project> to search all
      active projects.
ii. <u>State</u>: Use the drop-down list to select the State you
      want to search in. Use <Any State> to search all
      states.
iii. <u>Package Group</u>: Use the drop-down list to select the
      Package Group you want to search in. If you have
      selected a Project, you can display only the packages
      that belong to package groups within that Project.
      Similarly, if you select a State, the search is narrowed
      to look in Package Groups within the specified State.
c. Advanced tab:
      On this tab you can set additional selection criteria.
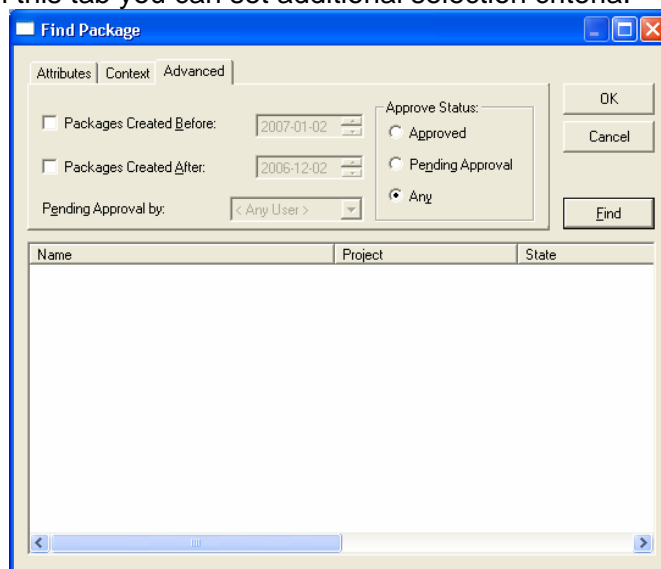
     i. <u>Packages Created Before</u>: Use this field to search for packages created before a specified date.

    ii. <u>Packages Created After</u>: Use this field to search for packages created after a specified date.

   iii. <u>Waiting Approval by</u>: Use the drop-down list to find packages that are pending approval by a specific user.

☞ Please note that *Pending Approval by:* filter uses the <u>role</u> of the user named in this field rather than the actual user in its search.

Example: Project A, State X is selected as my Context. JDOE is <u>not</u> a member of Project A but is a member of the Developer's user group. If I select JDOE in this field, it will display all packages in Project A, State X pending approval by a Developer.

---

   iv. <u>Approved Status</u>: You can locate packages based on their approval status.
1. <u>Approved</u> – based on users for whom packages have been approved.
2. <u>Not Approved:</u> based on users for whom packages are awaiting completion of approvals.
3. <u>Any:</u> based on users for whom packages have been approved and are awaiting approval.

☞ Searching for an Approve Status of "Pending Approval" results in all packages that awaiting one or more approvals. To find whose approvals are pending, use the "Pending Approval by:" option.

---

3. After you've set all your filter criteria, click **Find**. The packages matching your filter criteria will be displayed in the list box.
4. Right-click on a package to see a list of available processes you can run against the package. You can:
   a. View the Form associated with the selected package.
   b. View the Properties of the selected package.
   c. Run available processes.
      (NOTE: "Available" here means those processes you are authorized to run in the package's Project and State. Example: A Program Administrator does not have Approve permissions in Development. If the package they select from the list is in the Development State they will not see the Approve process.)

8.15 Find Version (or Select Version)

The **Find Version** feature allows you to search for a version of Harvest item based on a number of criteria. After locating the version(s) you can execute available processes directly from the found list.

> This section describes how this feature works when invoked from the Tools menu or the Harvest toolbar. The feature can also be invoked from other dialogs (e.g. during a Check Out for Browse or Check Out for Update). If invoked from another dialog, this feature is called Select Version but the functionality is the same.

1. To access the Find Version feature, go to **Tools** | **Find Version** or click on the **Find Version** icon ![icon] from the Harvest Toolbar.
2. Use the *Option: Name & View, Option: Date* and/or *Context Setting* tabs to refine your search. It is recommended to set your Context before selecting other filter criteria.

    a. Context Setting tab:
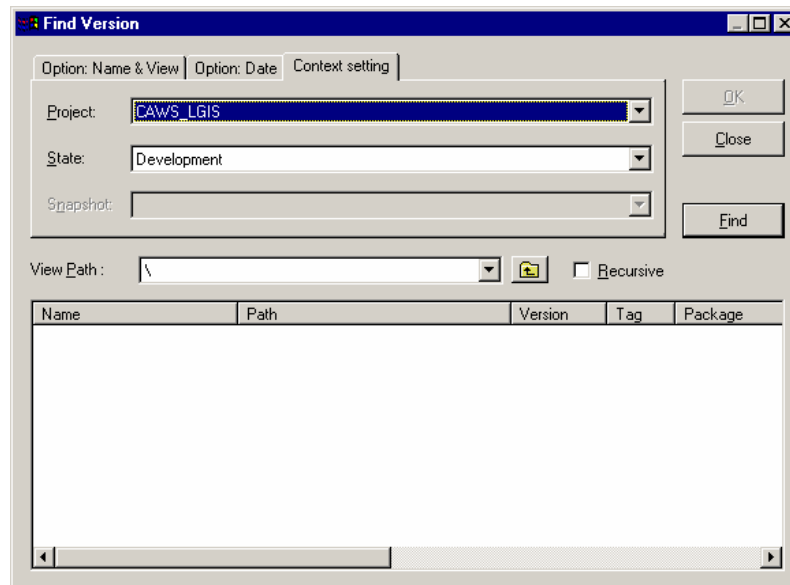
        i. Project – Allows you to specify the Project you wish to search in.
        ii. State – Allows you to specify the State you wish to search in.

    b. Option: Name & View tab:
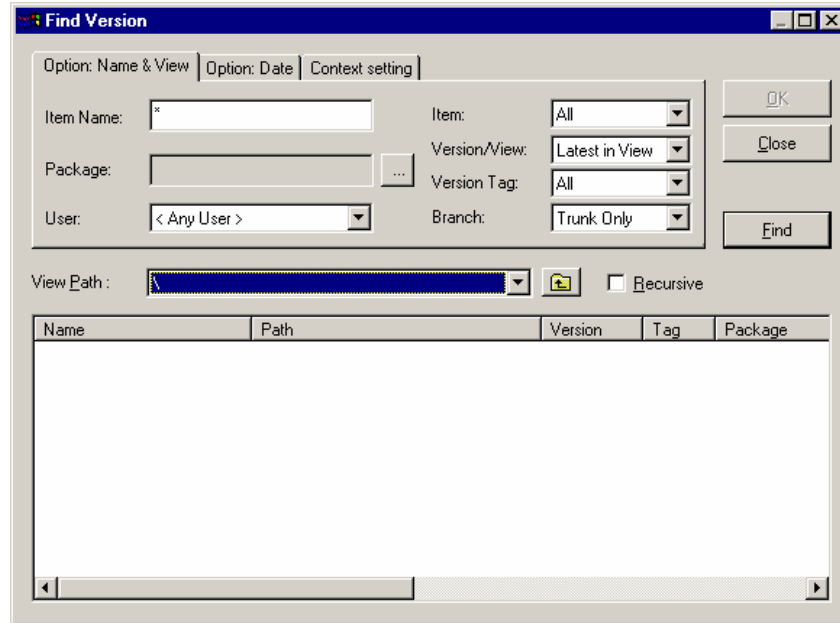
i. <u>Item Name</u>: Allows you to filter by item name. Wildcards (*) can be used for multiple character matching, question marks (?) for single character matching

ii. <u>Package</u>: Allows you to filter by the package that created the version (dependent on which Project/State context you selected).

iii. <u>User</u>: Allows you to filter by the user who created the version.

iv. <u>Item</u>: Allows you to filter by the modification status of the item.

    1. Modified: Displays only versions of items that have been modified.

    2. Not Modified: Displays version for items only if the item has not been modified in the project.

    3. All: Displays all version of all items whether they've been modified or not.

v. <u>Version/View</u>:

    1. All in View: Displays all versions matching the other filtering criteria that exist in current state's view. Any later versions of an item that exist in another view that have not yet been promoted to the current state are not displayed.

    2. Latest in View: Displays the latest version of each item in the current view that matches the other filtering criteria. Only one version per item is displayed.

    3. All: Displays all versions, including those that do not yet exist in the current state's view, that match the other filtering criteria.

41

4. Latest: Displays the latest version of each item that matches the other filtering criteria, including those that do not exist in the current state's view.

vi. <u>Version Tag</u>:
1. All: Displays versions that match the other filter criteria regardless of their tag.
2. No Tag: Displays versions that don't have a tag associated and that match the other filter criteria.
3. Reserved: Displays only versions that are tagged as Reserved (R) that match the other filter criteria.
4. Merged: (Not applicable to our Harvest environment.)
5. Removed: Displays only versions that are tagged as Removed (D) that match the other filter criteria.
6. Any Tag: Displays only versions that are tagged as Reserved (R), Merged (M) or Removed (D) and that match the other filter criteria.

vii. <u>Branch</u>: (Not applicable to our current Harvest environment.) Leave at default value of "Trunk only".

viii. <u>View Path</u>: Allows you to change the current location to any position within the current path. Clicking the button at the right of the drop-down list moves up one directory in the current path.

c. Option: Date
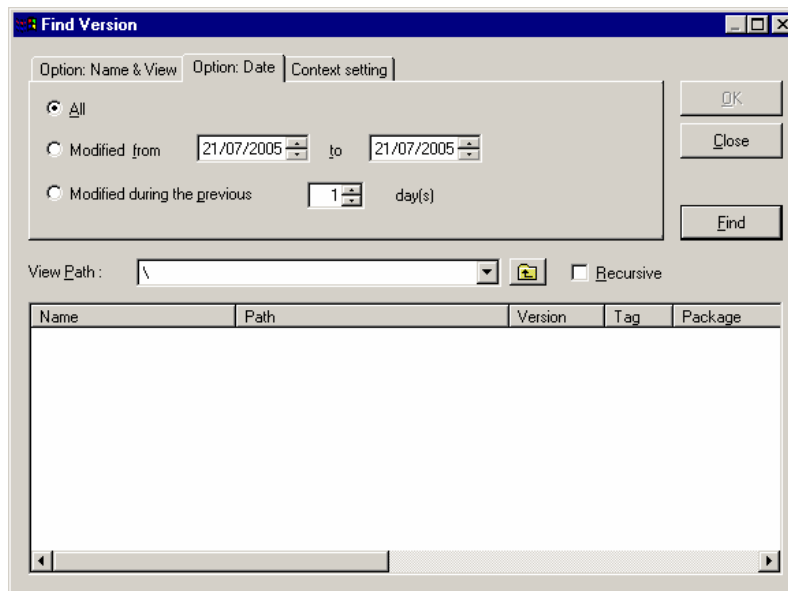
i. <u>All</u>: Indicates date checking will not be performed.

ii. <u>Modified From <Date> to <Date></u>: Use this option to search between two dates. Edit the dates manually or use the scroll buttons to select the month, day and year.

iii. <u>Modified during the Previous X days</u>: Use this option to search the previous X number of days.

d. <u>Recursive</u>: Use this option to display versions that match the other filtering criteria in every path below the current one. You can then select versions from multiple paths to be included in this operation.
<u>Note</u>: Selecting the Recursive option disables the View Path drop-down list. To change your position after filtering recursively, deselect Recursive before you try to move.

3. Click **Find**.
4. A list of items matching your filter criteria will be presented. The resulting list will display item name, path, version, tag, package, creator, create date, modifier and modified date.
5. Right-clicking on an item on the list will present the processes that you are permitted to run on that item in that view.

What processes you can run here will depend on your access permissions in the state/view where the item resides.

For example, if the item exists in the Development view but you don't have Check Out for Update privileges in Development, you will not see the Check Out for Update process listed.

If you filter your search by Version/View "All" or "Latest", you will find that you are unable to perform processes directly from the list. This is because the versions span multiple views and your permissions may vary between those views.

8.16 Hold Package

The **Hold Package** process is a promote process to the Hold State.

1. Make sure you are logged into the **Harvest Workbench** and you have selected the **Project** and **State** you wish to work in.
2. Expand the **State** node and then the **Packages** node to display all the packages currently in that state.
3. Right-click on the package you wish to put on hold and choose **Processes** | **Hold Package** from the shortcut menu.
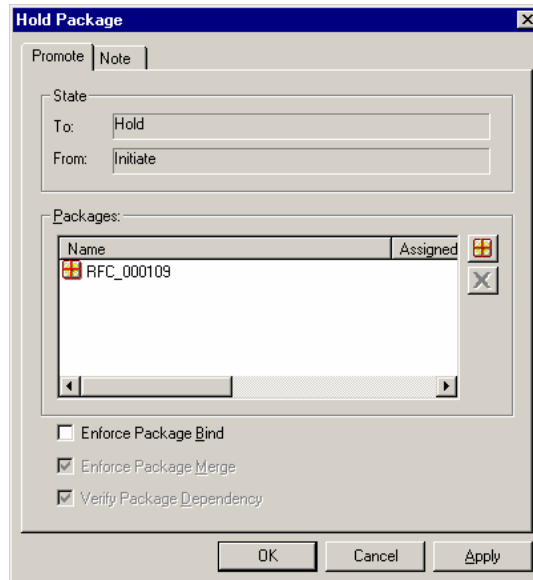
4. Click **OK** to confirm that you want to put this package into a Hold state. Note that the package no longer exists in the current state.

8.16.1. Putting Multiple Packages on Hold

You can place more than one package on Hold at a time.

1. Make sure you are logged into the **Harvest Workbench** and you have selected the **Project** and **State** you wish to work in.
2. Expand the **State** node and click on the **Packages** node. The list of packages in that State will be displayed in the List View.
3. Ctrl- or Shift-Click to select the **packages** you wish to put on Hold.
4. Right-click on the selected packages and choose **Processes** | **Hold Package** from the shortcut menu.
5. Click **OK**.

8.17 Printing

Harvest Forms may be printed using a number of methods (see below). Other Harvest reports may be generated using the (third-party) Harvester application.

With a Harvest Form displayed, you can print the form using any of the following 3 Print methods:
1. Click on the **Print…** button (located on the bottom left of the form); or
2. Go to File | Print...; or
3. Right-click on the Form, then click **Print**.

8.18 Promote Process

1. Make sure you are logged into the **Harvest Workbench** and you have selected the **Project** and **State** you wish to work in.
2. Expand the **State** node and then the **Packages** node to display all the packages currently in that state.
3. Right-click on the package you wish to promote and choose **Processes** | **Promote** from the shortcut menu.
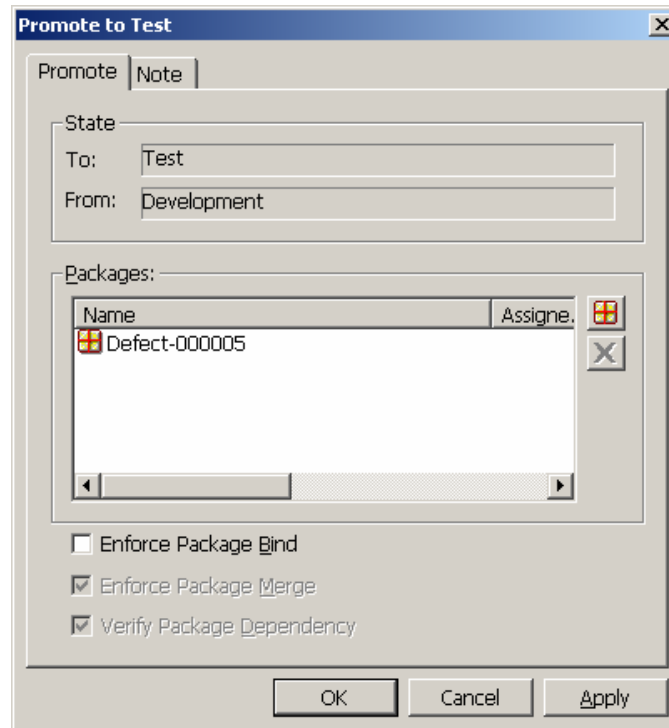
4. Click **OK** to confirm that you want to promote the package. Notice that the package no longer exists in the current state.

8.18.1. Promoting Multiple Packages

You can Promote more than one package at a time.

1. Make sure you are logged into the **Harvest Workbench** and you have selected the **Project** and **State** you wish to work in.
2. Expand the **State** node and click on the **Packages** node. The list of packages in that State will be displayed in the List View.
3. Ctrl- or Shift-Click to select the **packages** you wish to Promote.
4. Right-click on the selected packages and choose **Processes** | **Promote to <Next State>** from the shortcut menu. The *Promote to <Next State>* dialog will appear with selected packages listed.
5. Click **OK**.

8.19 Reject Package

The Reject Package process is a promote process to the Reject State.

1. Make sure you are logged into the **Harvest Workbench** and you have selected the **Project** and **State** you wish to work in.
2. Expand the **State** node and then the **Packages** node to display all the packages currently in that state.
3. Right-click on the package you wish to reject and choose **Processes** | **Reject Package** from the shortcut menu.

4. Click **OK** to confirm that you want to Reject the package. Notice that the package no longer exists in the current state.

> Once a package has been "rejected", it can only be brought back by the Harvest Administrator. Alternatively, a new package can be recreated in the Initiate State.
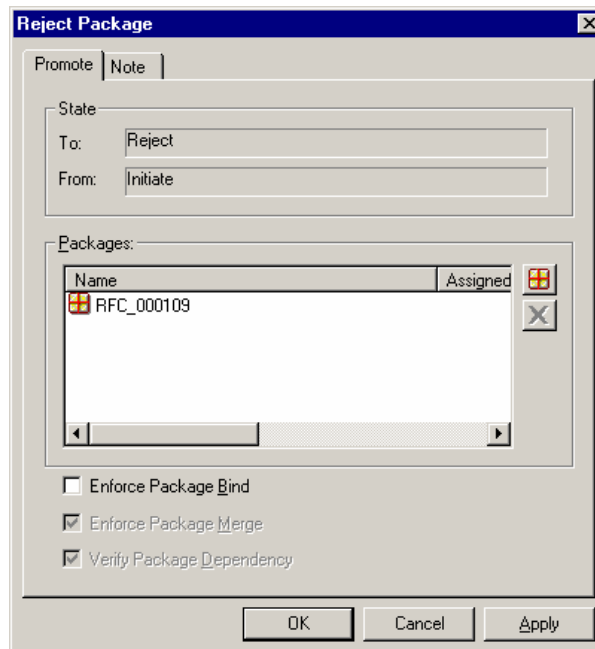
8.19.1. Rejecting Multiple Packages

You can Reject more than one package at a time.

1. Make sure you are logged into the **Harvest Workbench** and you have selected the **Project** and **State** you wish to work in.
2. Expand the **State** node and click on the **Packages** node. The list of packages in that State will be displayed in the List View.
3. Ctrl- or Shift-Click to select the packages you wish to Promote.

4. Right-click on the selected packages and choose **Processes** | **Reject Package** from the shortcut menu. The *Reject Package* dialog will appear with selected packages listed.
5. Click **OK**.

Note: The "Hold Package" and "Reject Package" processes are available only with the following states:

- Initiate

- System Requirements

If you would like to put a package on Hold or Reject a Package in **SAD** state the processes to use are:

- "Send Package on Hold"

- "Send Package to Reject"

You are not able to put Packages on Hold or Reject Package in any other states other than those listed above.

8.20 Take Snapshot

1. Make sure you are logged into the **Harvest Workbench** and you have selected the **Project** and **State** you wish to work in.
2. Right-click on the **State** and choose **Take Snapshot** from the shortcut menu.
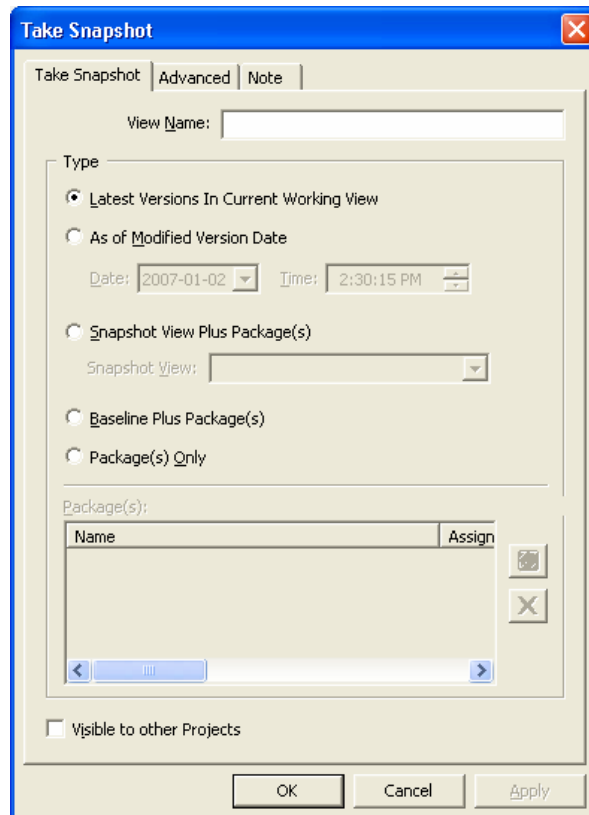

3. Enter a name for the snapshot in the **View Name** field. The name should be entered according to the standards identified. See the Harvest Administrator for reference to naming conventions.
4. Select **Type** as required:
   a. Latest Versions in Current Working View: Captures the latest versions in the current working view. If the latest version in the working view has been reserved, the latest one with a Normal tag is selected.
   b. As of Modified Version Date: Captures versions in the current working view during a specified timeframe;
   c. Snapshot View Plus Package(s): captures versions in a specified snapshot view plus listed packages in the Packages list box.
   d. Baseline Plus Package(s): Specifies the latest versions in the baseline plus the latest versions in the specified packages.
   e. Package(s) Only: Specifies the latest versions in the specified packages.

5. To create a partial snapshot of the repository go to the *Advanced* tab, select **Partial Repository** and select which parts of the repository you want to include.
6. Click **OK**.
7. Verify the snapshot by expanding the **View Snapshot** state, expanding the Data Views node and you will see the snapshot you created.
8. Select the "Visible to other Project" option if you would like this snapshot viewable to other Projects.

9. Harvest Options

On the Workbench, the Harvest Options menu allows you to establish window and file default settings. Some of these options are described below. For a full description of all options, please refer to CA's *AllFusion Harvest Change Manager User Guide*.

9.1 General
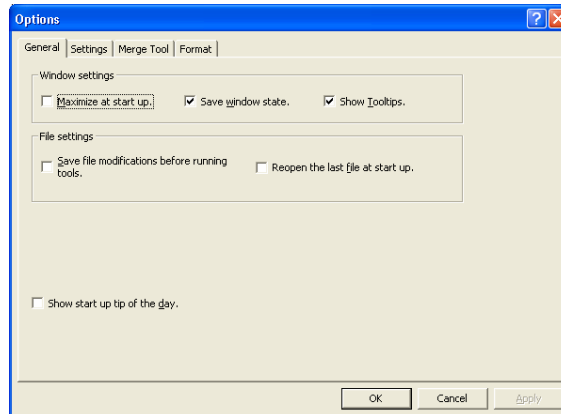
1. Windows Settings
   a. Maximize at start up – If checked, the Harvest window is maximized at start up.
   b. Save window state – If checked, the Harvest window state is retained.
   c. Show Tooltips – If checked, Tooltips are displayed if you hover your mouse over an icon.

9.2 Settings
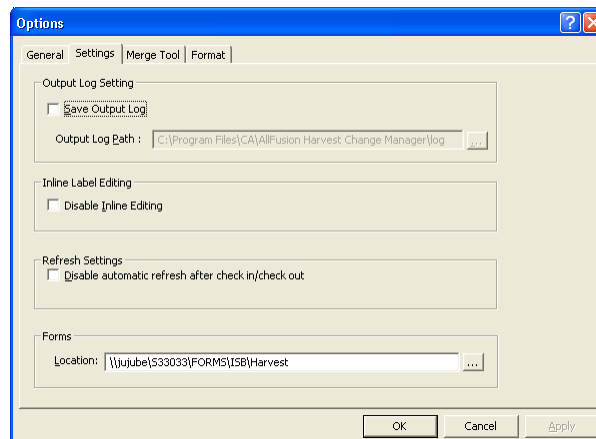
1. Output Log Setting
   Save Output Log – allows you to save the Output log to a file. When this option is enabled, a file is created when you log into Harvest and log information is recorded in it.
2. Inline Label Editing
   Disable Inline Editing – If checked, disables inline editing for objects in the workspace tree and lists.
3. Refresh Settings
   Disable automatic refresh after check in/check out – allows you to disable automatic refresh of version lists after check in/check out.
4. Forms
   Location - Allows you to specify the location path where the Harvest forms are stored. Note: For the Ministry, this defaults to a shared location. Changing the setting may mean you will be unable to view forms associated with Harvest packages.

9.3   Merge Tool

The Merge Tool tab allows you specify an external difference/merge tool to use for file comparison and merging. The default merge tool is Harvest for the internal Harvest Compare View/Visual Difference, Compare File, and Interactive Merge processes. To learn more about the Merge Tools, please refer to the CA's AllFusion Harvest Change Manager User Guide.