ClearCase/FrameMaker Integration Overview

Peter Klenk
Atria Software, Inc.

Agenda

1. Introduction
2. Integration Overview
3. Integration Goals
4. What It Does
5. Configuring the Integration
6. Diff
7. Merge
8. Alternate Approach to Diff and Merge
9. Bookfiles
10. Future Ideas
11. Q&A/Discussion
    Frame Integration, PureDDTs Integration, Integration Requests, Etc.

Integration Overview

ClearCase/FrameMaker Integration v1.0 released 1Q95.

- Supports AIX, HP-UX, IRIX, Solaris, SunOS.
- Requires ClearCase 2.0 or later.
- Requires FrameMaker or FrameBuilder (FM+SGML) 4 or later.
- Is a Frame API client developed using the Frame Developer's Kit (FDK).
- Uses existing element type (compressed file) and type manager (whole_copy_z). Note that `atria/config/magic/default.magic` specifies Frame-specific element types (i.e. frame_document) which are used if they exist in the vob.

Integration Goals

Allow Frame users to version control their documents in ClearCase without leaving the familiar Frame GUI. Minimal ClearCase knowledge is required.

This allows documentation to follow the same process as source code with minimal learning on the part of Frame users.

However, users must understand:

- Using a view, either by setting a view prior to starting Frame or using view-extended naming.
- Basic checkout/checkin version control model.

What It Does

The integration adds a ClearCase submenu to Frame's File menu. Fundamental version control commands are available in the menu:

- Checkout
- Checkin
- Uncheckout
- Compare predecessor

The `mkelem` operation is hidden behind checkin. Trying to checkin a view-private file will prompt the user to make a new element (which is then checked in). Directory checkouts and checkins are handled automatically.

The integration also helps the user do the right thing:

- Query to checkout on first modification of a checked-in version.
- Query to save a modified document before checkin.
- Avoid duplicate successive versions (for any element type).

Configuring the Integration

Enabling the Integration

The integration is enabled by an entry in the `fminit/usenglish/APIArea/apiclients` file. The global file lives in `$FMHOME`, but users can override it in their home directory.

The `CCASE_FRAME_OPTS` environment variable allows changing default behavior:

- `-off` turns off the integration
- `-ci_identical` allows creation of identical successor versions
- `-co_unreserved` does unreserved checkouts instead of reserved checkouts. Make sure you have a policy for merges if simultaneous revisions are allowed.
- `-unco_keep | -unco_rm` controls whether view private copies are saved when the user does an uncheckout operation (prompt with a dialog box is the default).

Diff

Compare functionality is provided by Frame. ClearCase provides the set of historical revisions.

Compare Predecessor

- Available directly from the *File->ClearCase* menu.

Comparing Arbitrary versions

- Open each version you wish to compare.
- Select *File->Utilities->Compare Documents*.

Navigating Version-extended Namespace

- Choose *File->Open*.
- Select the file name of an element.
- Manually type the extended naming symbol (`@@`) in the edit field after the file name and press return.
- You can now navigate branches as directories, and versions and labels as files.

Merge

Merge is not available from the *File->ClearCase* menu because Frame does not support automatic merge.

Trivial Merge

A ClearCase merge involves a base and two or more contributors. The base is the common ancestor of the contributors. In a trivial merge only one contributor differs from the base. Trivial merge is supported by all Atria-provided type managers.

Trivial merge allows the branching model used for code to be applied to documentation, as long as only one branch has active development at a time. Therefore config specs and views can be shared.

Manual merge

- Detect necessary merges with:
  `cleartool findmerge ... -print`
- Diff the versions and use copy/paste to create the merged result.
- Draw merge arrows with:
  `cleartool merge -to pname -ndata pname`

Alternate Approach To Diff and Merge

Take advantage of the fact that Maker Interchange Format (MIF) files are ASCII text.

WARNING: This technique requires understanding the syntax and semantics of MIF, which is not supported by Atria.

- Always work with MIF files in Frame.
- Use text file delta type manager for MIF elements.
- Use standard `cleartool diff` and `merge` commands.
- Watch out for `DNextUnique` and `Unique` statements when merging.

Or

- Continue storing and editing documents in Frame binary format.
- Create a type manager that redefines the diff and merge methods to convert files to MIF before using the text file delta manager, and converts merge results back to Frame binary format.

Bookfiles

A Frame book is a list of document files and the relationships between them.

Both the bookfile itself and each the member file can be versioned. Operations can be performed on just the bookfile, or the bookfile and all the members files.

The ClearCase menu has the following additional items for books:

- Checkout Entire Book
- Checkin Entire Book
- Uncheckout Entire Book

Frame does not support comparing book files.

Related Offering

Atria contributed package (email server `contrib@atria.com`) T0010 offers an element type and trigger type that recursively apply ClearCase `checkout`, `checkin`, and `uncheckout` operations (from `cleartool`, `xclearcase`, or elsewhere) on bookfiles to the bookfile and each component file.

Future Ideas

Access Frame from ClearCase UI:

- Invoke Frame diff from `cleartool` or `xclearcase`.
- Open Frame documents from `xclearcase`.
- Text-based diff and merge of bookfiles.

Help on using `clearmake` to:

- Build postscript versions of documents.
- Generate TOC, Index, etc. for books.