



ALLEGRIA SOFTWARE, INC.

A Technical White Paper

by Ruiwei Wang

Sr. Software Engineer

October, 1996

**The Integration of
ForReview with Documentum**

Overview

In today's corporate world and global marketplace, the need to share, view, mark-up and manage office documents, images and technical drawings has become a growing challenge. To address this need, **Allegria Software, Inc.**, an innovator in image and documentation applications, has recently integrated *ForReview*TM for Windows, an enterprise-wide view and mark-up product, with *Documentum*TM, a technical document management system designed for managing technical documents and drawings.

ForReview for Documentum is a separate integration module that seamlessly integrates *ForReview* with *Documentum*. This integration allows *Documentum* users to quickly and easily load documents directly into *ForReview* by utilizing *ForReview*'s Graphical User Interfaces (GUI's), and *Documentum*'s powerful search and select facilities. It should be emphasized that the integration module does not modify any of *ForReview*'s underlying code. The module communicates between *ForReview* and *Documentum* by using *ForReview*'s powerful Application Programming Interface (API) and via Dynamic Data Exchange (DDE).

The *ForReview for Documentum* module adds a "ForReview" menu entry to the *Documentum* menu bar, allowing users to quickly access *ForReview* tasks. The "drop down menu" from this entry allows the user to easily *view*, *mark-up* and *print* a selected document by selecting the corresponding menu item, or by double-clicking on the document itself. If multiple documents are selected, the menu operation will apply to each individual document in a sequential manner.

The user can also associate a *ForReview*-native mark-up with a particular document in the database, list all the mark-up files associated with the selected document in the *Documentum* database, and set up the proper image formats to be recognized by both *Documentum* and *ForReview for Documentum*. Users can also specify the default vault or folder location to store new mark-up objects, and indicate whether or not to keep mark-up annotations attached when assigned a new version number to the base document. This can be done at either the user level or the system administrator level.

ForReview always protects original documents by not allowing them to be edited in any way. Redlining is always performed on user defined layers, and saved in a separate file called a "mark-up" file (recognized by the *.mkp* extension). To accommodate these mark-up files in *Documentum*, a new type known as *Markup* (derived from *dm_note*) is defined in the *Documentum* database, inheriting all attributes of a *Documentum*-native note object. Using this new *Markup* type, *ForReview* can automatically check in or out documents/mark-ups to or from the database.

ForReview automatically creates and maintains the relationship between mark-up files and their corresponding base documents in the *Documentum* database. When querying about a base document stored in *Documentum*, the user can quickly find and access all related mark-up files. Once the mark-up file/s have been loaded, the user can then make use of *ForReview*'s extensive mark-up and layer management.

The *ForReview for Documentum* module essentially provides two modes of operation: **View** and **Mark-up**.

- In **View** mode, the user can only *look* at drawings and mark-ups files. View mode restricts the user by not allowing changes to mark-up files, or creation of any new mark-ups files.
- In **Mark-up** mode, the user *can* modify existing mark-up files, or create new mark-up files. Any modified or newly created mark-up file will be saved to the database when the base document is closed. Furthermore, the mark-up files loaded are actually *checked-out* from the database, so it is impossible for any two users to make modifications to the same file concurrently.

Functional Architecture

ForReview for Documentum allows *ForReview* to be seamlessly integrated with *Documentum*. Note that *ForReview for Documentum* is a true integration, not just a "launch". This means that when a file from *Documentum* is opened in *ForReview*, *ForReview* will first automatically check it out of the *Documentum* database before displaying it in *ForReview*. Similarly, when you are done with the document or the annotations, *ForReview* will automatically check the document(s) back into the *Documentum* database.

When working in *Documentum*, *ForReview* is started only when a *ForReview* function is evoked and vice versa. In the event that the connection between *Documentum* and *ForReview* is lost (due to a network failure for example), any work done in *ForReview* can be saved locally to the back-up directory “\markup” in Workspace’s working directory. If files are saved locally for this reason, they can be checked back in to the *Documentum* database after the connection is re-established. This offers an additional layer of protection for the user’s work. At any time, with or without the integration, you can use *ForReview* to view and mark-up files that are local or outside the *Documentum* database, but you will obviously not be able to access any files stored in the *Documentum* database if you are using *ForReview* independently.

To modify or create mark-ups, highlight the base document to be redlined from *Documentum*, then select **Markup** either from the **ForReview** menu or from the right-mouse button, or double-click on the document (if *Markup* is the default double-clicking behavior). At this time, the integration module in the form of a DLL is loaded into memory, and a network connection is made to the *Documentum* database. This is accomplished by invoking the appropriate API functions provided by *Documentum*.

The object ID of the selected document is then retrieved from the *Documentum* database, and a copy of the file is made and stored in a temporary directory on the local disk. Next, a list of mark-up files associated with the base document is retrieved through a query to the database, and the files are grouped in the *Select Markup Files* dialog along with their attributes such as object name, version number, author, permissions, etc., to aid the user in selecting which mark-up files are to be loaded into *ForReview for Documentum*. The mark-up files are organized according to their version trees, and the user can decide whether to load all versions on the tree or only the latest one. The user can also decide whether to load a modifiable mark-up file in view mode only, to prevent accidental change of the mark-up file. For those mark-up files selected by the user, a temporary copy of the file is also made and stored in the same temporary directory as the base document.

Users can now proceed to “mark-up” the base document using facilities provided in *ForReview for Documentum*. *ForReview for Documentum* keeps track of the base document and its mark-ups as “external”, to distinguish from “native” mark-ups not from the *Documentum* database. When the user has finished redlining the base document, either by closing the base document or by explicitly unloading a particular mark-up file, the integration module is notified of this action through callback functions set in *ForReview for Documentum*. The integration module either updates the database for an existing mark-up object or creates a relationship between the base document and the new mark-up file and stores the new mark-up object into the database. In either case, the user is prompted with a *Documentum*-native dialog to change or set any relevant attribute of the mark-up file. After all its mark-ups are unloaded and the document is closed, copies in the temporary directory are automatically deleted.

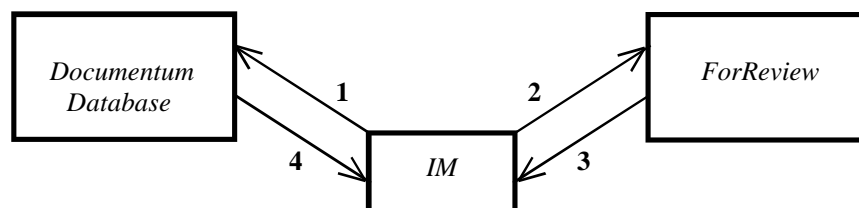


Fig.1 Flowchart of functional architecture

1: Object information retrieved from *Documentum*, and passed to the Integration Module (IM). 2: After processing the information, the integration module loads the base document and associated mark-ups into *ForReview*. 3: When the user is finished with marking-up the base document, *ForReview* informs the integration module of changes using callback functions that the integration module has set in *ForReview*. 4: The integration module stores the new or modified mark-ups back into the *Documentum* database by calling the appropriate *Documentum* API's.

Modules in Action

Following is a demonstration of the integration modules in action. There are no menu changes to *ForReview* when running *ForReview for Documentum*. However, *ForReview for Documentum* adds a **ForReview** menu entry to *Documentum* to quickly access *ForReview* tasks. These entries include **View**, **Markup**, **Print**, **Import Markup**, **List Markups**, **Preferences**, and **Help** (Fig.2).

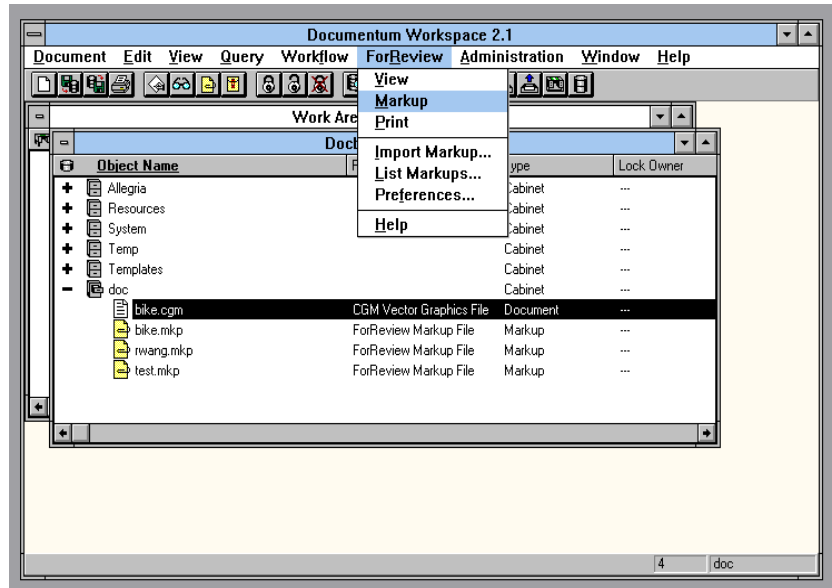


Fig.2 Menu changes to *Documentum*

To mark-up a base document, in this case - *bike.cgm*, highlight the base document and select **Markup** from the **ForReview** menu.

ForReview for Documentum is automatically launched, if it is not already running, and the base document, (*bike.cgm*) is loaded. The first time *ForReview for Documentum* is invoked from *Documentum*, the user is prompted with a *ForReview for Documentum Login Dialog*. The login information is required for *ForReview for Documentum* to access the *Documentum* user database.

The *Select Markup Files* dialog appears (see Fig.3), allowing the user to choose what associated mark-up files are to be loaded with the document. The mark-up files are organized according to their version numbers, with only the newest version on each version tree displayed when the dialog first comes up. By toggling the **All Versions** checkbox, all other versions on the version tree can be displayed or collapsed. Furthermore, if the user has more than viewing privilege, the **View Only** checkbox can be checked to indicate that the particular mark-up should be loaded in view mode.

The user can also select the **Attributes...** button to bring up the *Documentum-native Attributes Dialog*, further aiding him or her in selecting which mark-up to load into *ForReview for Documentum*. For the newest version, the user can also change some of the mark-up attributes “on the fly”. By toggling the symbol before the filename, the user can indicate if the mark-up file is to be selected. The default “plus” symbol means to load the mark-up. The four column fields, mark-up file name, version number, author name, and permissions, can further aid the user in selecting the right mark-up (Fig.3).

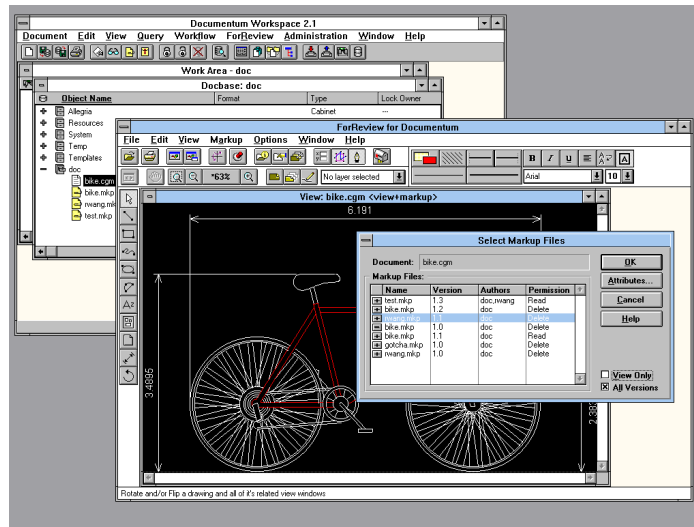


Fig.3 Loading base document and selecting markup files

By clicking the **OK** button, the selection state is accepted, and the selected mark-ups will be loaded. By selecting the **CANCEL** button, no mark-ups in the list will be loaded. When finished redlining the base document, either by modifying an existing mark-up file or creating a new mark-up file, the user is prompted with a *Documentum*-native *Attributes* dialog to set any relevant attributes of the mark-up file (Fig. 4).

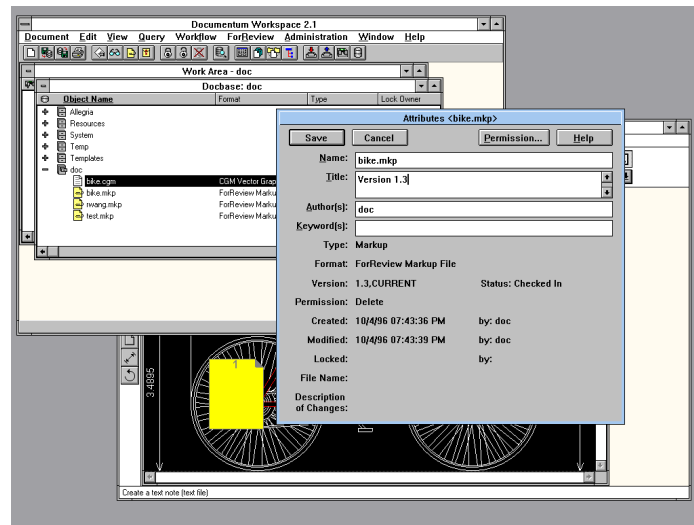


Fig.4 Setting the attributes of a modified markup file

This completes the steps involved in a **Markup** operation. The **View** and **Print** operations are essentially the same, in that the user is not allowed to make any modifications to the mark-ups. These three modes are the basic operations that can be performed on a base document and its mark-ups.

Occasionally, the user might just want to see a list of mark-ups associated with a particular document. This functionality is provided by the **ForReview|List Markups...** operation. In still some other cases, users might want to associate a “native” mark-up file created in *ForReview* with a base document already in the *Documentum* database. To do this, highlight the base document, and choose the **ForReview | Import Markup...** command. An *Open File* dialog will appear, allowing the user to browse through the directories to select the desired mark-up. As in the case of storing a new or modified mark-up file back into the *Documentum* database, the user is also prompted with a *Documentum*-native *Attributes* dialog to set any relevant attributes of the mark-up file.

Finally, the **Preferences...** option allows the user to easily update the relation between an image format and its designated viewer, providing additional user customization flexibility. When the *Preferences Dialog* is invoked, it shows all of the supported formats in both the *Documentum* and *ForReview* systems, with the current settings “+” on or “-” off. The user can also specify the preferred path of the cabinet/folder where the new mark-up objects created by *ForReview for Documentum* should be stored. If the cabinet/folder does not exist, it will be created. The default cabinet/folder is the user’s login name prefixed by a ‘/’. Finally, the user can also indicate if the mark-up objects should be kept attached when the when the base document is assigned a new version number. The when the base document is assigned a new version number using *Documentum* facility. These settings can be updated any time in a session (Fig.5).

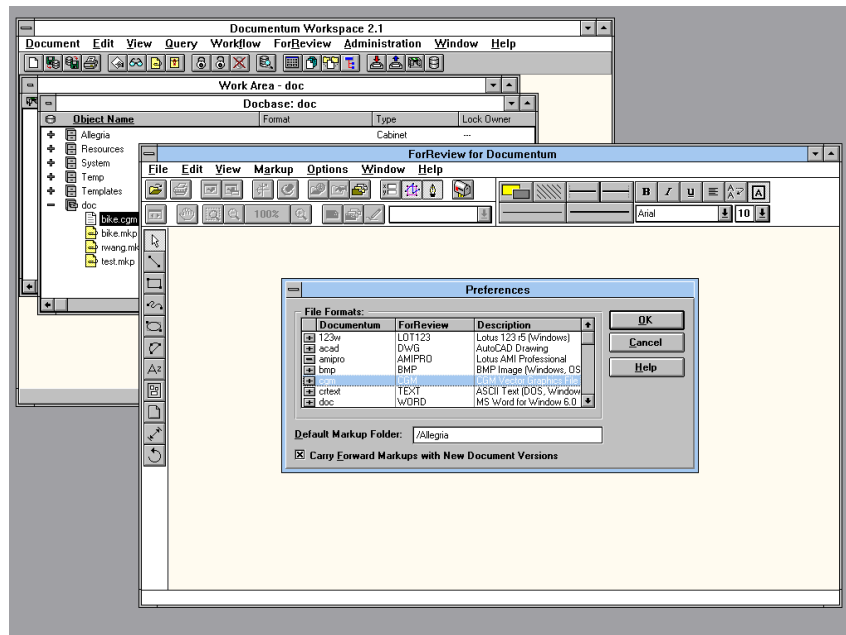


Fig.5 Setting user preferences

Conclusion

In conclusion, we have demonstrated that *ForReview*, a state-of-the-art view and mark-up tool has been tightly integrated with *Documentum*, a Technical Document Management system. We believe this integration will meet the challenge of viewing and managing office documents, images and technical drawings in a demanding way. The integration can also serve as a template architecture followed by other integration work.