Article
Patrick Newton · Mar 10, 2016 5m read

Atelier Synchronization

Atelier Synchronization Logic

Background

Documents may be edited on the client independently of a connection to a Caché server. However, when a connection is established with a server for the purposes of saving, compiling and debugging code, then the client and server should agree with respect to the version of documents that are being operated upon. When they achieve this agreement, we say that they are “ in-sync ” and that the synchronization process is completed. There are several occasions for the client and server to get out-of-sync including when creating a new file on the client or modifying a file on the client or server. There are also several edge cases that need to be specially handled, which will be detailed in the following sections. Atelier creates an internal version model for all workspace projects and provides wizards and other user interface elements to give users full control over their source code. This document focuses primarily on the logic behind the user interface elements rather than on the elements themselves.

Create New File

If a user creates a new file on the client using the Atelier file creation wizards, then that file will not exist on the server until a server save is triggered. This represents a conflict of versions because the version on the client is physically different from the one on the server – the server version does not yet exist. As a result, a warning marker is attached to the file alerting users that the file has not yet been saved on the server. Once a server save is triggered, either manually or automatically, the server responds with its version data which is recorded on the client. At this point the client and server will have the same file version and content data – they are in-sync – and the warning marker is removed.

If a user creates a new file on the client by copying a file from the server to a project on the client, then the server provides both file version and content data. These are saved on the client in a new file at the user specified workspace location. No markers are added or removed in this case because it does not represent a conflict in the workspace. By definition the files are in-sync.

Importing files (e.g. from source control) for the first time is also an example of file creation. In this case, files from the file system are created in the workspace. As a result, they will not have any associated version information in their project and as such a warning marker is attached to each imported file. Atelier provides a batch resolve wizard which facilitates synchronizing multiple files at once. This process updates any missing version information and allows users to replace server files with the newly imported files as well. Users also have the option to synchronize files individually if they prefer.

Modify File

File modification on the client is composed of two actions (i) making changes in the Atelier editor and (ii) saving those changes. The first step changes the model of the file within the Atelier workspace, while the second step saves those changes to the client ’ s file system. It is important to recognize that the Atelier workspace and client file system are not exactly the same components. Rather the workspace references the file system. Files can exist in the file system without being a part of the workspace, just as changes in the workspace will not be reflected in the file system until they are saved. If a user modifies a file in the Atelier editor and saves their change, a server save is triggered and the content and version data is synchronized.

In a shared server scenario and a second user independently modifies a file causing the server version to change, a warning marker is added to the file alerting the first user that the file has been changed from beneath them on the server. This state can be identified by the fact that the version information on the client no longer matches that on the server. Users can verify differences in version information by looking at the file properties.

The server can also automatically modify files during the compilation process (e.g. persistent classes are modified to include a storage block). These changes are automatically incorporated into the client document and version information synchronized. In a shared server scenario, all other clients will become out-of-sync and a warning marker will be added to alert users.

Handling Edge Cases

Dirty Files

If a user modifies a file in the Atelier editor without saving it, it becomes “ dirty ” . Attempts to compile or debug this file will result in the user being prompted to save their changes. Since inadvertent changes are fairly common, Atelier does not assume that unsaved changes should be automatically saved. Instead, users are prompted to save all desired changes and Atelier synchronization is then able to reliably act on the version from the local file system.

If changes are not saved, then the version on the local file system will be used. If the server does not need to be updated, then no additional file content or version information is sent to the server.

Once the client and server are in-sync again, compile and debug actions can proceed as usual.

Failed Connections

The version information on the client is populated by responses from the server, so if the server is not accessible no synchronization information can be updated on the client. Therefore synchronization state cannot be validated until the client can reconnect to the server. Once reconnected, users can use the Atelier synchronization wizards to get their workspace back in-sync with the server.

Failed server save attempts will cause the associated version information to be removed and a warning marker to be attached to the file. As a result, locally modified files can be properly synchronized with the server once the connection has been reestablished.

Conflicting Files

In the shared server scenario, users may often need to edit files that are out-of-sync because their teammates have made changes to the server. Saving changes to a file that is out-of-sync causes a conflict so users are prompted to select which version of the file to keep in order to resolve the issue. Atelier does not allow users to merge files to resolve conflicts – users must use their source control system if file merging is desired. Once the conflict has been resolved, one version of the file will be replaced with the other version and the client and server will be in-sync again.

Changes to Files Outside the Workspace

It is possible that files in an Atelier project might rely on files not included in that project or any other workspace project. If files outside of the workspace are modified, users may be interested in being alerted to these changes.

However, Atelier does not manage synchronization state for files outside of the workspace, so users are expected to either (i) verify it manually using the Server Explorer view or the project library from the Atelier Explorer view or (ii) add all relevant files to their workspace.

#Development Environment

Source URL:https://community.intersystems.com/post/atelier-synchronization