

Source Code Repository



The Adelia source code repository is a service enabling the functions of the [source import and export tool](#) to be directly integrated into the 4GL editor and the Adelia object manager.

It also provides an interface to Git and SVN source control tools, enabling direct access to a program's history from the editor.

NB: like the source import and export tool, the repository is not compatible with the Microsoft Access database manager®.

The features provided are:

- Direct import/export of data from the object manager or 4GL Editor.
- Automatic export of the source code when saving a document.
- Comparison of disk and Adelia repository versions.
- Automatic synchronization (optional) of data from the environment and repository. In this case, any changes to an object in the environment (rename, delete) will be automatically made in the repository.

In addition, if integration into a source control manager is configured, it is also possible to:

- show the history of an object or the whole Adelia repository
 - compare an object's source with a version from the history
 - import a version of a program from the history
 - commit (create a version in the source control manager) or go back to changes in progress
 - in an Adelia version manager production environment, automatically commit changes received following a correction validation.
- In this case, the comment generated for the commit can also indicate the Jira tickets if a bug manager is configured in the correction environment.



Automatic synchronization limitations

Automatic synchronization automatically exports the changed objects and transparently manages object deletion (git rm or svn rm) or renaming (git mv or svn mv) with the source control manager. However, you may encounter the following limitations:

- An object is considered to be renamed if its name or visibility (transition from public to private, or vice versa) is changed. However, the application area change for a private object is not taken into account. In Adelia this requires two separate operations (detaching from the old application area then attaching to the new one).
- If an object is renamed outside the Adelia repository (direct use of git mv), the source repository cannot monitor the change. In this case, two records appear for the object in the commit or revert dialog boxes: the old object name appears with the word "Deleted" and the new one with the word "Renamed". The two records must be committed or cancelled at the same time to avoid the risk of generating inconsistencies.
- If an object is deleted from the source control manager (because it was deleted or renamed in the environment) and you create a new object of the same name and type, the source repository will refuse to export this new object until the deletion has been committed. In addition, errors may be returned if you try to cancel the deletion or renaming when the repository contains an object with the same name as the one that needs to be restored.
- Automatic synchronization is transparent and runs in the background of all Adelia applications (apart from the Adelia version manager and exchanges which explicitly manage synchronization). As a result, if there is a synchronization error, no message box is displayed. You can view the logs if you encounter problems ("com.hardis.adelia.sync" logger in log4c, server-side ".adelia\server.log" file in the source repository).

Automatic synchronization is optional and is enabled in the [environment parameters](#).

NB: Adelia only provides the basic functions for the source control manager interface. Advanced features (branch management, version merge, etc.) must be managed directly using the dedicated tools of the source control manager used.