

Object Locking

In a Natural Single Point of Development environment, Natural Studio provides a locking mechanism that prevents concurrent updating of Natural objects. These can be local Natural objects or objects accessed on a remote development server.

This document describes the locking concept, the objects affected, the different behavior with remote and local objects and the restrictions that apply in conjunction with a mainframe server.

The following topics are covered:

- Object Locking Concept
 - Locking of Local Objects and Remote Objects
 - Mainframe Server Restrictions
-

Object Locking Concept

When being edited in one of Natural Studio's editors or tools, the Natural objects listed below are locked for local or remote access.

- Program
- Subprogram
- Subroutine
- Copycode
- Help Routine
- Text
- Dialog
- Map
- Class
- Local Data Area
- Global Data Area
- Parameter Data Area
- DDM
- Application (only remote)

The locks are set exclusively by the editors or tools. If the objects are locked by an editor, the locked objects are unlocked as soon as you close the editor. If the objects are locked by a tool (like e.g. the Class Builder when local classes are changed), the locked objects are unlocked when all internal links to the object are released. When you terminate your Natural session, all existing locks of that session are released.

Within your Natural session, you can edit different sources from different libraries in parallel in multiple editor sessions. In this case, a lock exists for all sources involved. You cannot edit the same source in different editor sessions, neither when you are in a single Natural session nor when you are in multiple Natural sessions. If you are using tools like the Class Builder, the object can be changed from all views; all views show the current state of the object.

Move, Delete and Rename

When you are issuing a Move, Delete or Rename command on a Natural source object, the locking state of the object concerned is checked:

- If the source object is locked, the command execution is rejected.
- If the source object to be moved, deleted or renamed is not locked, the command is executed.

The execution of such a command does not cause the object to be locked.

Save and Stow

You can execute a `SAVE` or `STOW` command only on a source object that is not locked. If the object has been locked by another user, the command execution is rejected.

The following scenario may cause inconsistent data:

1. A source object is being edited in Natural session A. The object is locked.
2. The same object is displayed in Natural session B using the `READ` command (by the same or another user). In this case, the original state of the source object is read. The object cannot be edited because it is locked.
3. In Natural session A, the modified object is stored using the `SAVE` command and the editor is closed. The editor unlocks the object.
4. In Natural session B, the object read is loaded into the editor with the `EDIT` command. It is modified or it is not touched, but a `SAVE` or `STOW` command is executed on it. This causes the modifications made in session A to be overwritten.

Locking of Local Objects and Remote Objects

Local Objects

Locking of sources in the local environment does not require the existence of a Single Point of Development environment. It is effected using Windows operating system functions. There is no difference with respect to the locking mechanism available with Natural for Windows Version 4.1.

Remote Objects

Remote objects are locked only if they are edited using one of Natural Studio's local editors. This requires the installation of a Natural Remote Development Server plug-in on the server side. The locks are managed by the remote development server and are stored as user-specific data in the remote development server file.

Different Locking Behavior with Remote and Local Objects

In the local environment, it is possible to execute a `SAVE` or `STOW` command on a locked object only in that Natural session in which the object was locked.

In the remote environment, it is possible for a user who has locked an object in a Natural session to execute a `SAVE` or a `STOW` command on this object from within another session. However, this user will not be able to edit this object in a Natural Studio editor.

If, after session termination, Natural Studio is unable to unlock objects in the event of an error, then the locks on objects in the local environment will normally be released by the operating system. This does not apply to locked objects in the remote environment. These objects can be unlocked using the `UNLOCK` command.

Mainframe Server Restrictions

The locking of source objects of a Development Server is supported only in a Single Point of Development environment. The source objects are locked only when they are edited in Natural Studio.

To ensure correct object locking you are strongly recommended to make changes on the same FUSER system file used in a SPoD environment exclusively in that environment. Using the FUSER system file concurrently by another stand-alone Natural session may give rise to inconsistencies. This applies also to a Natural session which works with an FNAT system file with installed Remote Development Server but is not operated from within Natural Studio.

When working with Natural Studio, care must be taken to start all commands or utilities from within Natural Studio. It is not admissible to issue system commands in the terminal emulation window, for example at a MORE prompt or in a command line. In such a case, the object locking will not be supported. Particularly, editing of source objects directly with the server's editors will provoke inconsistencies.