Natural Main Menu

The Natural **Main Menu** provides access to Natural development functions, environment settings, utilities and example libraries.

This section contains information on the functions and input options provided by the Natural **Main Menu** and its subordinate menus.

Invoking or Closing the Natural Main Menu

There are two methods of invoking or closing the Natural **Main Menu**:

- You can define a default setting by switching menu mode on or off. Menu mode causes the Natural **Main Menu** to be invoked automatically for the next session started.
- You can invoke or close the Natural **Main Menu** within a Natural session whenever required.

To switch menu mode on or off before session start

- At Natural startup, specify either of the following profile parameters:

  ```
  MENU=ON
  ```

  (activates menu mode)

  or

  ```
  MENU=OFF
  ```

  (deactivates menu mode)

  See also *MENU* in the *Parameter Reference* documentation.

To invoke or close the Natural Main Menu within a session

1. Enter either of the following system commands:

```
MAINMENU
```

(invokes the menu)

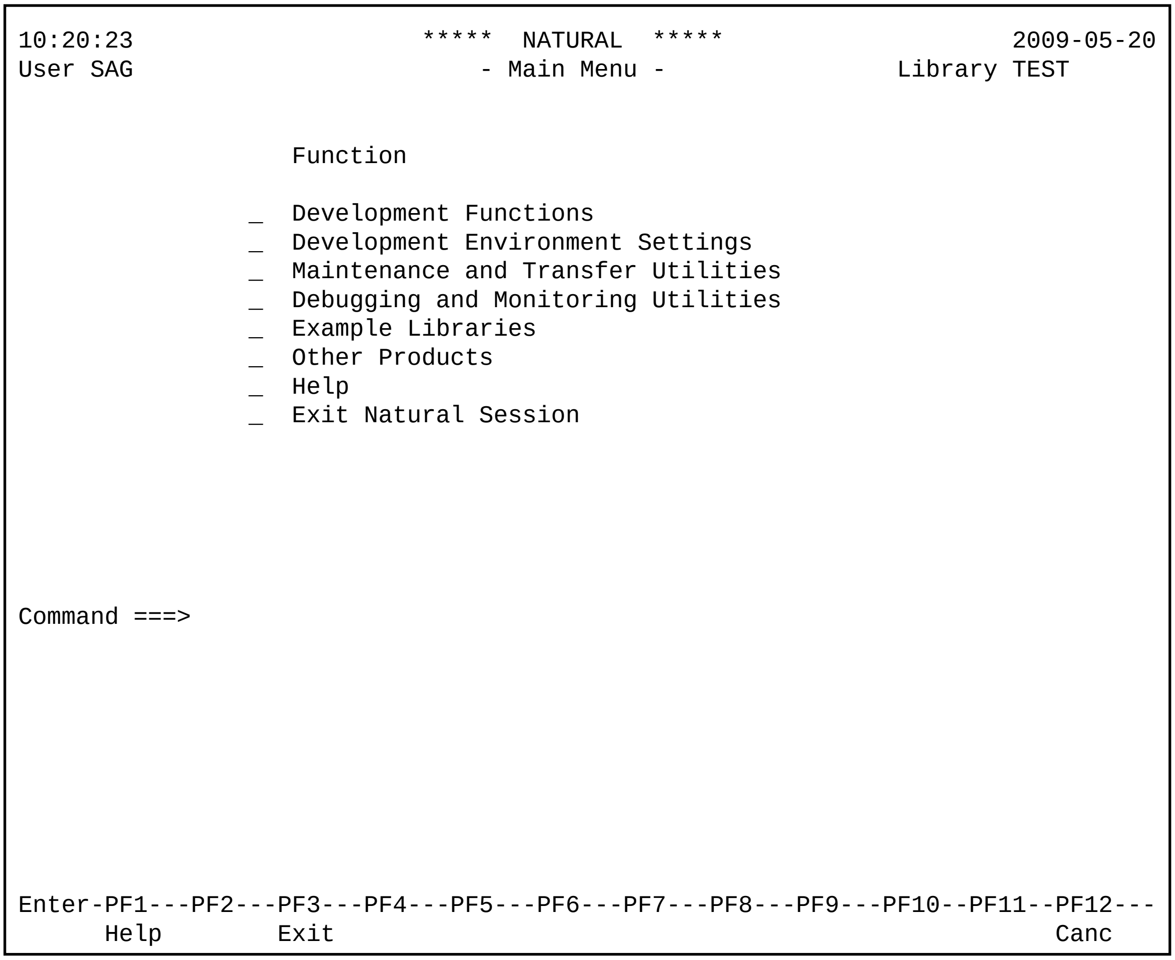
or

```
MAINMENU OFF
```

(closes the menu)

2. Choose ENTER.

The Natural **Main Menu** looks similar to the example shown below:



```
10:20:23                    *****  NATURAL  *****                    2009-05-20
User SAG                        - Main Menu -                Library TEST


                   Function

                _  Development Functions
                _  Development Environment Settings
                _  Maintenance and Transfer Utilities
                _  Debugging and Monitoring Utilities
                _  Example Libraries
                _  Other Products
                _  Help
                _  Exit Natural Session






Command ===>









Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help        Exit                                                  Canc
```

At session start, Natural assigns you to a default library that is indicated on the screen. In the example screen above, the ID (name) of the library is shown in the **Library** field in the top right-hand corner of the screen. See also *Default Library Assignment*.

For instructions on performing a menu function, refer to the relevant section in *Using Commands and Menu Functions*.

Each function listed in the Natural **Main Menu** invokes a menu of the corresponding name where you can select further functions:

Function in Natural Main Menu	Explanation of Functions in Corresponding Menu
Development Functions	Creates and maintains programs, maps, data areas and other components that make up a Natural application.
Development Environment Settings	Displays and modifies various settings that affect your Natural session.
Maintenance and Transfer Utilities	Invokes a Natural utility to create and maintain certain objects or transfer them to another environment.
Debugging and Monitoring Utilities	Invokes a Natural utility to monitor your Natural applications and locate errors in their processing flow.
Example Libraries	Selects libraries containing example programs and Application Programming Interfaces (APIs).
Other Products	Invokes other Software AG products.

Note:
The position and color of the message line and PF-key lines in the Natural **Main Menu** and its subordinate menus can be changed with the user exit routine USR2003P described in the *Operations* documentation.

Development Functions

The functions listed in the **Development Functions** menu are those you will need most frequently when you develop an application with Natural. The functions apply to all Natural objects that are available in the library where you are currently logged on.

The table below contains information on the fields provided in the **Development Functions** menu:

Field	Explanation
User	The ID of the Natural user who logged in the current session.
Library	The library currently active. See also the section *Using Natural Libraries*.
Mode	The programming mode: reporting or structured mode. See *Programming Modes*.
Work area empty	Indicates that no source has been loaded into the source work area. If a source has already been loaded into the source work area, the type and the name of the object will be displayed instead, for example, `Program PROGX`.
Code	The code that corresponds to the function required, for example, **C** for **Create Object**. See also *Performing a Menu Function*.
Type	The type of object such as `P` for program. For details, see *Object Types* in the *Programming Guide*. You can leave the **Type** field blank if you specify the name of a Natural object that already exists. If you want to change the object type, see also *Setting the Object Type*.
Name	The name of the object. For an explanation of valid object names, see *Object Naming Conventions*.
Command ===>	The command line. It is an input field in which you can enter a Natural command. For example: To edit an existing program named PROGX, you would enter the following system command: `EDIT PROGX` See also *Using Commands and Menu Functions*.
PF (function keys)	PF keys (function keys) can be used as an alternative to using commands or menu functions. The PF-key lines at the bottom of the screen indicate which function is assigned to which key. See also *Standard PF Keys*.

The table below contains information on the functions provided in the **Development Functions** menu. For most of the menu functions, there are equivalent Natural system commands. These alternative system commands are listed in the table and further explained in the relevant sections in the *System Commands* documentation.

Function	Code	Explanation
Create Object	`C`	Invokes a Natural editor where you can create a new object such as a program, map or data area. Specify the type and the name of the object to be created. You can enter a question mark (?) in the **Type** field, to select an object type from a list of all types available for this function. See also *Creating and Editing Objects*.
Edit Object	`E`	Invokes a Natural editor and displays the source of the specified object in modify mode. Specify the name of an existing object to be edited. You can also invoke a selection list of objects: see *Specifying Object Ranges*. You can enter a question mark (?) in the **Type** field, to select an object type from a list of all types available for this function. See also *Creating and Editing Objects*. Equivalent system command: `EDIT`
Rename Object	`R`	Invokes the **Rename Objects** window where you change the name of the specified object and/or the object type. See also *Renaming Objects*. Equivalent system command: `RENAME`
Delete Object	`D`	Invokes the **Delete** window for the specified object. In the **Delete** window, confirm the deletion by entering the name of the object again in the relevant input field. You can also invoke a selection list of objects as described in *Specifying Object Ranges*. In this list, you can mark one or more objects for deletion. See also *Deleting Objects*. Equivalent system command: `DELETE`
Execute Program	`X`	Executes an object of the type program. Specify the name of the object to be executed. Other object types cannot be executed by themselves, but must be invoked from another object. See also *Executing Programs*. Equivalent system command: `EXECUTE`

Function	Code	Explanation
List Object(s)	`L`	Displays the source code of the specified object. Specify the name of the object to be displayed. You can also invoke a selection list of objects: see *Specifying Object Ranges*. You can enter a question mark (?) in the **Type** field, to select an object type from a list of all types available for this function. See also *Listing Objects in a Library*. Equivalent system command: `LIST`
List Subroutines Used	`S`	Ascertains which objects use which external subroutines and classes. Equivalent system command: `ROUTINES`

This section covers the following topics:

- Programming Modes
- Natural Editors
- Specifying Object Ranges

Programming Modes

Natural offers two programming modes: reporting mode and structured mode. We recommend that you exclusively use structured mode, because it provides for more clearly structured applications. Therefore, all explanations and examples in the Natural tutorial *First Steps* and the *Editors* documentation refer to structured mode. Any peculiarities of reporting mode will not be taken into consideration.

For further information on programming modes, see the section *Natural Programming Modes* in the *Programming Guide*.

The **Mode** field in the top right-hand corner of the **Development Functions** menu indicates the programming mode currently in effect: structured or reporting.

To switch programming modes

1. In the upper right-hand corner of the **Development Functions** screen, in the **Mode** field, overwrite the first position with an `S` to switch on structured mode, or an `R` to switch on reporting mode.

   Or:
   Enter either of the following system commands:

   ```
   GLOBALS SM=ON
   ```

   (switches on structured mode)

or

```
GLOBALS SM=OFF
```

(switches on reporting mode)

2. Choose ENTER.

   The contents of the **Mode** field has changed from `Reporting` to `Structured` or vice versa.

Related Topics:

- *Natural Programming Modes* - *Programming Guide*
- *GLOBALS* - *System Commands* documentation

Natural Editors

Depending on the type of object specified in the **Development Functions** menu, Natural invokes the appropriate editor: the program editor, the map editor or the data area editor. For further information on these editors, see the relevant sections in the *Editors* documentation.

Specifying Object Ranges

The functions **Edit Object**, **List Object(s)** and **Delete Object** provide the option to either specify the name of an individual object or a range of names. When you specify a range of names, a list of objects is displayed from which you can select one or more objects you wish to edit or list, or mark for deletion.

To list all objects

1. In the **Name** field, enter an asterisk:

   ```
   *
   ```

2. Choose ENTER.

   A list of all objects available in the current library is displayed.

To list objects using a start value

1. In the **Name** field, enter a start value followed by an asterisk (*).

   This option to enter a value followed by an asterisk is referred to as asterisk notation.

   For example:

   ```
   AB*
   ```

2. Choose ENTER.

   A list of all objects with names that start with AB (for example, AB, AB1, ABC, ABEZ) is displayed for the current library. The list does not include object names that start with AA1 or ACB, for example.

Note:
The **List Object(s)** function provides further options to specify object name ranges as described for the equivalent system command `LIST`.

Development Environment Settings

The table below contains brief descriptions of the functions provided in the **Development Environment Settings** menu, and lists the Natural system commands that correspond to these functions. For further information on a system command, refer to the relevant section in the *System Commands* documentation.

Function	Explanation	Correspond. Command
Function-Key Settings	Assigns functions to PF keys to be used in your Natural session.	`KEY`
Compilation Settings	Sets options that affect the way in which Natural objects are compiled.	`COMPOPT`
Session Parameter Settings	Changes the settings of Natural session parameters. See also *Configuring your Natural Environment* and *Session Parameters* in the *Parameter Reference* documentation.	`GLOBALS`
Profile Parameter Settings	Changes the settings of Natural profile parameters. Profile parameters are described in the *Parameter Reference* documentation and in *Profile Parameter Usage* in the *Operations* documentation. The system command `SYSPARM` invokes a utility of the same name that is described in the *Utilities* documentation.	`SYSPARM`
Technical Session Information	Displays technical information on your Natural session such as the current user ID, library and operating system.	`TECH`
System File Information	Displays the current definitions of Natural system files. See also *Natural System Files* in the *Natural System Architecture* documentation.	`SYSPROF`
Product Installation Information	Displays a list of the products installed at your site and information on these products.	`SYSPROD`
Security Profile Information	Only available if Natural Security is installed. Displays the security profile currently in effect.	`PROFILE`

Maintenance and Transfer Utilities

The table below contains brief descriptions of the functions provided in the **Maintenance and Transfer Utilities** menu, and lists the Natural system commands that correspond to these functions. Each of these commands invokes a Natural utility that is described in the *Utilities* documentation.

Function	Explanation of Utility	Correspond. Command
Maintain Error Messages	Creates and maintains messages you wish to issue in your Natural applications.	`SYSERR`
Maintain DDMs	Creates and maintains data definition modules (DDMs).	`SYSDDM`
Maintain Command Processors	Creates and maintains the command processors you wish to use in your Natural applications.	`SYSNCP`
Maintain Remote Procedure Calls	Establishes and maintains remote procedure calls and provides the settings required to execute a Natural subprogram located on a remote server.	`SYSRPC`
Transfer Objects to Other Libraries	Transfers Natural objects between different libraries.	`SYSMAIN`
Transfer Objects to Other System Files	Unloads or loads Natural objects. You can use either the system command `SYSUNLD` to invoke the initial utility menu for unloading or loading objects, or the system command `NATUNLD` or `NATLOAD` to directly invoke the subordinate load or unload utility. NATUNLD utility: unloads Natural objects from a Natural system file to a work file. NATLOAD utility: loads Natural objects from a work file into a Natural system file. Note that the functionality of NATUNLD and NATLOAD is covered by the Natural Object Handler. We recommend that you use the Object Handler instead.	`SYSUNLD`
Transfer Objects to Other Platforms	Transfers Natural objects and Adabas FDTs from one hardware platform to another. Note that the functionality of SYSTRANS is covered by the Object Handler. We recommend that you use the Object Handler instead.	`SYSTRANS`
Transfer Objects to Other Systems	Invokes the Object Handler to process Natural objects and foreign objects for distribution in Natural environments.	`SYSOBJH`

Debugging and Monitoring Utilities

The table below contains brief descriptions of the functions provided in the **Debugging and Monitoring Utilities** menu, and lists the Natural system commands that correspond to these functions. Each of these commands invokes a Natural utility that is described in the *Utilities* documentation.

Function	Explanation of Utility	Correspond. Command
Debugging	Searches for errors in the processing flow of programs.	`TEST`
Logging of Database Calls	Logs database commands.	`TEST DBLOG`
Issuing Adabas Calls	Passes Adabas commands directly to the database.	`SYSADA`
Buffer Pool Maintenance	Monitors the Natural buffer pool and adjusts it to meet your requirements.	`SYSBPM`
Editor Buffer Pool Maintenance	Monitors the buffer pool of the Software AG Editor and adjusts it to meet your requirements.	`SYSEDT`
TP-Specific Monitoring	Monitors and controls TP-monitor-specific characteristics of Natural.	`SYSTP`
Data Collection and Tracing	Collects monitoring and accounting data about the processing flow of a Natural application.	`SYSRDC`
Error Information on Abnormal Termination	Provides information Software AG technical support requires for error diagnosis.	`DUMP`

Example Libraries

When you select **Example Libraries** from the Natural **Main Menu**, a list of libraries is displayed. These libraries contain example programs for demonstration purposes and Application Programming Interfaces (APIs) provided by Software AG:

Library	Contents
SYSEXPG	Example programs shown and referred to in the *Programming Guide*.
SYSEXRM	Example programs shown and referred to in the *Statements* documentation and the *System Variables* documentation.
SYSEXV	Example programs that illustrate new Natural features.
SYSEXT	APIs and example programs for using the APIs. See also the system command *SYSEXT* described in the *System Commands* documentation.
SYSEXTP	Example programs and APIs for specific functions that apply only under certain TP monitors.

Other Products

When you select **Other Products** from the Natural **Main Menu**, a list of Software AG add-on products appears. These products are installed at your site and can be accessed from this menu.