Natural ISPF

User's Guide

Version 8.2.8

October 2020

This document applies to Natural ISPF Version 8.2.8 and all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 1989-2020 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA, Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at http://softwareag.com/licenses.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at http://softwareag.com/licenses/ and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third-Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at http://softwareag.com/licenses and/or in the root installation directory of the licensed product(s).

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with Software AG.

Document ID: ISP-USERGUIDE-828-20200913

Table of Contents

Preface vii
1 About this Documentation 1
Document Conventions 2
Online Information and Support 2
Data Protection 3
2 Logging on 5
Logging on to Natural ISPF 6
Cursor Sensitivity 7
Natural ISPF Main Menu 7
Natural ISPF Screens 10
Working in a Multi-CPU Environment 11
Logging off from Natural ISPF Sessions 12
Returning to Natural from Natural ISPF 13
3 Command Logic 15
Command Logic Overview 16
Session Commands - Definition 16
Function Commands - Definition 17
Local Commands - Definition 18
PF Key Assignments 19
Multiple Command Input 21
Issuing Commands by Magic Character 22
Selection Windows and Wildcards 23
4 Useful Features 25
Handling and Navigational Features 26
Functional Features 48
Additional Features in Lists 83
5 Common Objects 91
Natural Objects 92
Natural Views 113
Natural Error Messages Member: ERRMENU 124
Predict Descriptions 127
Con-nect Documents 129
User Workpool 134
Recovery Files 143
Container Files 145
6 Profile Maintenance 149
Profile Maintenance Overview 150
PF-Key Definition 151
Library Definition 153
Magic-Character Definition 155
Editor Profile 157
User Defaults 163
BS2000 Defaults 166

Natural Defaults ... 171
Editor Color Definition ... 173
7 Editor ... 175
Editor Overview ... 176
Starting the Editor ... 176
Special Features ... 178
Editor Commands ... 179
Invoke the Online Help Facility ... 226
Main Commands ... 229
Line Commands ... 192
8 z/OS Objects ... 197
PDS Members ... 198
Load Modules and CSECTs ... 207
z/OS Dataset Maintenance ... 216
z/OS Job Information ... 233
z/OS System Operations ... 245
9 CA Panvalet Members ... 265
CA Panvalet Overview ... 266
Function Commands ... 268
Line Commands ... 271
Local Commands ... 272
Handling CA Panvalet Control Cards ... 272
10 CA Librarian Members ... 273
CA Librarian Members Overview ... 274
Function Commands ... 348
Line Commands ... 280
Local Commands ... 281
Handling CA Librarian Control Cards ... 282
Previous Versions ... 282
11 z/VSE Objects ... 285
z/VSE Members ... 286
z/VSE File Maintenance ... 295
z/VSE Job Information ... 305
z/VSE System Operations ... 310
12 BS2000 Objects ... 315
BS2000 Files ... 316
LMS Library Elements ... 329
BS2000 Job Information ... 341
BS2000 Job Variables ... 347
Console Display ... 356
13 Command Reference ... 357
Note about Command Syntax Symbols ... 358
Session Commands - Description ... 358
Function Commands - Description ... 381
Local Commands - Description ... 405

14 Keyword Parameters 415

Preface

Natural ISPF is Software AG's application development tool and provides a full range of application programming and system programming facilities in any environment that includes Natural.

This documentation is intended for the user of Natural ISPF and contains a full description of the Natural ISPF user interface and features. For a description of additional programming facilities (Macro Facility, Incore Database, Open NSPF, Natural ISPF API), consult the *Natural ISPF Programmer's Guide*.

This documentation covers the following topics:

Logging on	Describes how to log on to Natural ISPF and provides information about the online system: it introduces you to the Main Menu and explains the structure of system screens. A final subsection describes logoff procedures.
Command Logic	Explains Session Commands, Function Commands and Local Commands and their way of functioning.
Useful Features	Describes handling and navigational features, functional features and additional features in lists.
Common Objects	Explains all the functions you can perform on common objects, that is, objects available in all or any operating environment.
Profile Maintenance	With the profile maintenance facility, you can modify your user profile according to your personal requirements when working with Natural ISPF.
Editor	Provides an overview of Editor commands available in the Natural ISPF environment.
z/OS Objects	Provides information on all the functions you can perform on objects only available in a z/OS environment.
CA Panvalet Members	The Panvalet facility allows access to CA Panvalet members, which you can maintain using Natural ISPF functions.
CA Librarian Members	The Librarian facility allows access to CA Librarian members, which you can maintain using Natural ISPF functions.
z/VSE Objects	Explains all the functions you can perform on objects only available in a z/VSE environment.
BS2000 Objects	Describes all the functions you can perform on objects only available in a BS2000 environment.
Command Reference	Lists, explains and provides examples of all Natural ISPF session commands, function commands and local commands.
Keyword Parameters	Contains a list of all possible keywords and their synonyms that can be used as keyword parameters on Natural ISPF function commands.

Natural ISPF object types are grouped according to their relevant operating environment: not all of these may be relevant to you, depending on the operating platform and subsystems installed

at your site. For example, if the file management system CA Panvalet is not installed at your site, the section entitled *CA Panvalet Members* is not relevant to you.

Examples in this Documentation

This documentation contains a large volume of information, much of which is illustrated by examples. Many examples of commands are contained in the section that describes the various object types (for example `LIST`), but other examples may appear elsewhere.

Additionally, the Examples Library supplied on the Natural ISPF installation medium contains example programs, macros and command scripts that may be useful to your work. To see an overview of these examples, simply issue the command:

```
HELP EXAMPLES
```

from any Natural ISPF screen.

1 About this Documentation

- Document Conventions 2
- Online Information and Support 2
- Data Protection 3

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
`Monospace font`	Identifies service names and locations in the format *`folder.subfolder.service`*, APIs, Java classes, methods, properties.
`Italic`	Identifies: Variables for which you must supply values specific to your own situation or environment. New terms the first time they occur in the text. References to other documentation sources.
`Monospace font`	Identifies: Text you must type in. Messages displayed by the system. Program code.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
|	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the | symbol.
[ ]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [ ] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

Online Information and Support

Software AG Documentation Website

You can find documentation on the Software AG Documentation website at **http://documentation.softwareag.com**. The site requires credentials for Software AG's Product Support site Empower. If you do not have Empower credentials, you must use the TECHcommunity website.

Software AG Empower Product Support Website

If you do not yet have an account for Empower, send an email to empower@softwareag.com with your name, company, and company email address and request an account.

Once you have an account, you can open Support Incidents online via the eService section of Empower at **https://empower.softwareag.com/**.

You can find product information on the Software AG Empower Product Support website at **https://empower.softwareag.com**.

To submit feature/enhancement requests, get information about product availability, and download products, go to **Products**.

To get information about fixes and to read early warnings, technical papers, and knowledge base articles, go to the **Knowledge Center**.

If you have any questions, you can find a local or toll-free number for your country in our Global Support Contact Directory at **https://empower.softwareag.com/public_directory.aspx** and give us a call.

Software AG TECHcommunity

You can find documentation and other technical information on the Software AG TECHcommunity website at **http://techcommunity.softwareag.com**. You can:

- Access product documentation, if you have TECHcommunity credentials. If you do not, you will need to register and specify "Documentation" as an area of interest.
- Access articles, code samples, demos, and tutorials.
- Use the online discussion forums, moderated by Software AG professionals, to ask questions, discuss best practices, and learn how other customers are using Software AG technology.
- Link to external websites that discuss open standards and web technology.

Data Protection

Software AG products provide functionality with respect to processing of personal data according to the EU General Data Protection Regulation (GDPR). Where applicable, appropriate steps are documented in the respective administration documentation.

2 Logging on

- Logging on to Natural ISPF 6
- Cursor Sensitivity 7
- Natural ISPF Main Menu 7
- Natural ISPF Screens 10
- Working in a Multi-CPU Environment 11
- Logging off from Natural ISPF Sessions 12
- Returning to Natural from Natural ISPF 13

This chapter describes how to log on to Natural ISPF and provides information about the online system: it introduces you to the Main Menu and explains the structure of system screens. A final subsection describes logoff procedures.

Logging on to Natural ISPF

Since Natural ISPF is embedded in Natural, you can log on to Natural ISPF directly from the Natural menu or the `NEXT` prompt. Enter the command:

```
SPF
```

in the command input field of the Natural screen and press ENTER. The Natural ISPF logon screen appears:

```
--------------------------------------------------------------------------------
COMMAND ===>                                                          11:27:03
                                                                      11/02/07


            NN        NN     SSSSSSSSSS      PPPPPPPPPP    FFFFFFFFFF
           NNNN      NN     SS              PP      PP    FF
          NN NN     NN     SS              PP      PP    FF
         NN NNNN   NN     SSSSSSSSSS      PPPPPPPPPP    FFFFFFFFFF
       NN    NNN NN             SS      PP            FF
      NN      NNNN             SS      PP            FF
     NN        NN     SSSSSSSSSS      PP            FF




         User ID   HTR
         Password
         Node


     Press  ENTER to main menu or enter direct command
```

Enter your user ID and password and press ENTER to display the Main Menu. You are now ready for work with Natural ISPF.

Note: Your installation may be set up to bypass the logon screen. In this case, you are presented with the Main Menu immediately after invoking Natural ISPF.

If you are not automatically logged on to Entire System Server when logging on to Natural ISPF at your installation, Natural ISPF reminds you with a message whenever you wish to perform a function that requires a logon to Entire System Server. When this occurs, enter the command:

```
NATP-LOG
```

in the command input field in the second screen line and press ENTER. A window opens with your user ID and a prompt for your password and Entire System Server node ID. Enter your password and the required node number and press ENTER (if no security check is performed at your installation, no password is required). Natural ISPF confirms the successful logon and you can continue working as normal.

Cursor Sensitivity

All Natural ISPF screens are cursor-sensitive. You can select any menu item by either:

- typing its name in the command line, or
- marking the item line, or
- placing the cursor in front of the line you want to select.

Natural ISPF Main Menu

The following screen shows an example of the Natural ISPF Main Menu.

The Main Menu displayed at your installation may provide more and/or different options, depending on your environment. Some options are provided by Software AG as a standard, others may be added by your system administrator. This documentation describes all options provided by Software AG.

```
--------------------- NATURAL / NATURAL ISPF Main Menu ------------------------
OPTION  ===>
                                                                User ID  VMU
                                                                Time     14:11:05
 _ 1    NATURAL    - Development Functions                      Terminal DAEETCK7
 _ 2    NATURAL    - Development Environment Settings           Library  SYSISPHU
 _ 3    NATURAL    - Maintenance and Transfer Utilities         Node     148
 _ 4    NATURAL    - Debugging and Monitoring Utilities
 _ 5    NAT/NSPF   - Example Libraries

 _ 6    SAG        - SAG    Products
 _ 7    SYSTEM     - System Products

 _ 8    NSPF       - NSPF Administration

 _ 9    NSPF       - NSPF Changes

 _ HELP HELP       - NSPF    Help System
 _ NHLP HELP       - NATURAL Help System
 _ END  EXIT       - Exit NATURAL ISPF


Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

The options on the Main Menu illustrated above and the sub-menus invoked provide the following facilities:

Natural Development Functions

- **Natural:**
  Working with Natural objects means you have a full range of edit functions at your disposal for Natural members;
- **Views:**
  You can use this option to display Natural view definitions and database field contents in browse mode;
- **Error:**
  You can `LIST`, `EDIT` and `BROWSE` Natural error messages.
- **Predict:**
  Use this option to edit the Predict long description of any Predict object type;
- **Workpool:**
  With this facility, you can list and maintain the output of macro-type Natural programs and other objects that use the workpool as output destination;

- **Container:**
  You can `LIST` or `BROWSE` database files saved in the container file.

System Products - z/OS

- **Jobs:**
  This option allows you to display job SYSOUT according to selection criteria. You can hold, release and purge SYSOUT from the job entry system;
- **PDS:**
  Maintenance of PDS members;
- **Datasets:**
  Select this function to perform certain dataset maintenance functions such as compression, allocation, etc.

System Products - z/VSE

- **MEM:**
  Maintenance of z/VSE members;
- **z/VSE Files:**
  Maintenance of z/VSE files, libraries and sublibraries.

System Products - BS2000

- **BS2 Files:**
  Maintenance of BS2000 files;
- **LMS:**
  Maintenance of BS2000 library elements;
- **BS2 Jobs:**
  Use this option to display status information for BS2000 jobs;
- **JOB-Variables:**
  Maintenance of BS2000 job variables.

Other System Functions

- **Panvalet:**
  Maintenance of CA Panvalet members;
- **Librarian:**
  Maintenance of CA Librarian members;

- **System Operations:**
  This option allows the authorized user to perform system administrator functions and issue computer operator commands;
- **NSPF Administration:**
  This option allows the system programmer and administrator to customize Natural ISPF at installation time and to maintain system defaults and user definitions;
- **News:**
  Provides a summary of new commands and features available with the current version of Natural ISPF.

When you select an option from the Main Menu, the facility's Entry Panel is displayed.

Note: Not all Natural ISPF objects have Entry Panels, for example volumes, job SYSOUT, z/VSE sublibraries and previous versions of Natural, PDS or z/VSE members. These objects can be accessed through related facilities (Datasets, Jobs, z/VSE files, Natural objects, PDS objects, z/VSE files respectively) or by addressing them in commands.

Natural ISPF Screens

All Natural ISPF screens have a similar layout. Some data appear in every screen, other data are screen-dependent. Each system screen consists of four subsections:

- A header line with the name of the screen; error messages also appear in this line; the first columns of the header line contain session information, for example:

  Apart from the current session, there are two more sessions in this subsection of the physical screen (>>> = three more sessions, etc.).

`A*`	A command script generated by the `ALL` command is currently being executed.
`P*`	A command script is in `PAUSE` mode and can be continued with the `PAUSE` command.
`S*`	There are suspended sessions besides the current session.
`Z*`	Denotes a zoomed session in split-screen mode.

  These can be followed by a message that identifies the current function and object, for example, `LIST-PDS:MBE.COMN.SOURCE(*)/SCAN=ISPF`, which means that the `LIST` function was invoked for all members in the PDS library `MBE.COMN.SOURCE` which contain the string ISPF.
- A command input line; if the screen is an option menu, (for example the Main Menu), you can also enter the selected option code in this field;
- A main information subsection which consists of available options, fields in which you can identify the object for processing or in which you can specify certain session or system parameters;
- A two-line PF key subsection showing available PF keys and associated commands (optional).

The following figure illustrates a typical Natural ISPF screen: the Natural Objects Entry Panel (NATURAL option from the Main Menu):

```
---------------------- NATURAL OBJECTS - ENTRY PANEL ------------------------
 COMMAND ===>


    Library       ===>
    Member        ===>
    Type          ===>            ( Blank,P,S,N,C,M,G,L,A,H,T,O,4,8,Z,3,5,7,9 )
    Status        ===>            ( Blank,S,C,OS,OC )
    Scan for      ===>
    Edit macro    ===>
    Set number    ===>









 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
       Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

A common way of working with Natural ISPF objects is to enter a command in the command field, enter some object parameters in the other input fields and press ENTER.

Working in a Multi-CPU Environment

If you work in a multiple CPU environment, you can use Natural ISPF to access objects located on different physical machines. Each machine is identified by a unique node number (an Entire System Server node) which you can specify when addressing an object on that machine. The Main Menu contains a field labelled Node, in which you can specify a node to be addressed in all commands entered in this session (if different from the default node, usually 148).

For a list of available node numbers and a description of the associated machine, enter a question mark (?) in the Node field. A window opens with a list of defined machines and an availability status report (ACTIVE or INACTIVE). This feature is available in every screen that contains the Node field.

The list of available nodes is maintained by the system administrator (see also the *Natural ISPF Administration Guide*).

Logging off from Natural ISPF Sessions

There are several commands to end or interrupt work with Natural ISPF:

RETURN

To leave a Natural ISPF screen and return directly to the Main Menu of that session.

END

To leave a Natural ISPF session from the Natural ISPF Main Menu (usually assigned to PF3). If you are in single-session mode, you leave Natural ISPF. If you are in multi-session mode, the current session terminates and you are moved back to the next Natural ISPF session in the stack. You can also deactivate Natural ISPF sessions from the list invoked with the `ACTIVITY` session command.

LOGOFF [IMM] ;Natural-command

The `LOGOFF` session command allows you to terminate Natural ISPF even if you are working with several sessions.

- If your Editor profile has `AUTOSAVE=OFF` and you issue the `LOGOFF` command without parameters, logoff processing is interrupted if an Editor session with modified data is detected.
- If your Editor profile has `AUTOSAVE=ON` and you issue the `LOGOFF` command without parameters, all modifications are saved during logoff processing.

If Natural ISPF is defined as user interface for Natural, `LOGOFF` will close all sessions and return to the Natural ISPF Main Menu.

If you wish to terminate Natural ISPF immediately, that is, regardless of any modifications, you can use the `IMMEDIATE` parameter. All sessions are closed without saving and Natural ISPF is terminated.

You can concatenate the `LOGOFF` command with any valid Natural command. Examples are:

```
LOGOFF;FIN
```

Terminates Natural ISPF as described above and terminates the Natural session.

```
LOGOFF IMM;FIN
```

Terminates Natural ISPF immediately and terminates the Natural session.

```
LOGOFF IMM;SYSPROF
```

Terminates Natural ISPF immediately and executes the Natural utility `SYSPROF`.

NATURAL Natural-command

To temporarily interrupt work with Natural ISPF and start a session with Natural. An unqualified `NATURAL` command displays the Natural menu, but you can also issue the `NATURAL` command followed by a command from Natural itself. For example, the command:

```
NATURAL SYSDDM
```

gives you direct access to the Natural view maintenance menu. If you leave the Natural view maintenance facility in the normal way, you are automatically returned to the Natural ISPF screen from which you invoked the Natural facility.

You can return to your suspended Natural ISPF session simply by logging on to Natural ISPF from the Natural menu.

Note: If you do not return to Natural ISPF, all sessions are retained until a timeout occurs. It is therefore recommended that you return to Natural ISPF after using the `NATURAL` command.

ACTIVITY

To terminate several Natural ISPF sessions simultaneously, issue this command from the command line of any system screen. This displays a window with all active sessions. Enter a minus sign (-) in the command input line (`Cmd` column) for each session you wish to terminate and press ENTER.

Returning to Natural from Natural ISPF

When leaving Natural ISPF, you can concatenate Natural commands with Natural ISPF commands.

Examples:

```
END;SYSERR
```

issued from the Natural ISPF Main Menu in single-session mode ends the Natural ISPF session and invokes the Natural utility `SYSERR`.

```
RETURN;X;PROG1
```

issued from any screen except the Main Menu, returns to the Main Menu, terminates Natural ISPF and executes the Natural program `PROG1`.

```
LOGOFF;PROG2
```

logs off from Natural ISPF, even if you are in multi-session mode, returns you to Natural and executes Natural program `PROG2`. If you have any active `EDIT` sessions, the logoff is performed according to the setting of the `AUTOSAVE` parameter (see the description of the `LOGOFF` session command).

```
FIN
```

logs off from Natural ISPF and terminates Natural even if SAG Editor sessions not yet saved are open. The command corresponds to `LOGOFF IMM;FIN`.

You can also logon to another Natural library (change the library name) without leaving Natural ISPF by using the `LOGON` command with the new library name as command parameter. The new name appears in the Library field on the Natural ISPF Main Menu.

3 Command Logic

- Command Logic Overview 16
- Session Commands - Definition 16
- Function Commands - Definition 17
- Local Commands - Definition 18
- PF Key Assignments 19
- Multiple Command Input 21
- Issuing Commands by Magic Character 22
- Selection Windows and Wildcards 23

Command Logic Overview

Navigating and executing functions in Natural ISPF is done using commands issued from the command line of any Natural ISPF screen.

Alternatively, a command can be assigned to a PF key or a magic (special) character. Pressing the PF key or entering the magic character in the command line has the same effect as if the command were entered in the command line.

Natural ISPF commands fall into three main categories:

- Session commands
- Function commands
- Local commands

Note: Apart from these commands, a number of Editor commands are available from the command line of a `LIST`, `BROWSE` or `EDIT` session. An overview of Editor commands is provided in section ***Editor***.

The following subsections define these command types in more detail.

Command input is supported by a number of special features that include:

- Command redisplay (see section ***Useful Features***);
- Confirmation windows (see section ***Useful Features***).

Session Commands - Definition

With session commands, you can control your Natural ISPF session(s). For example, use the session command `SPLIT` to enter split-screen mode, `SUSPEND` to suspend the current Natural ISPF session and start a new one, `POP` to make a suspended session current, etc.

An overview of all available session commands is provided in the subsection ***Session Commands - Description*** of section *Command Reference*.

Function Commands - Definition

These commands address Natural ISPF objects (PDS members, Natural programs, Natural views, z/VSE files, job SYSOUT, etc.) and perform functions on them. Examples of function commands are `EDIT`, `BROWSE`, `PRINT`, `DELETE`, `LIST`.

Function commands can be issued in three different ways:

1. From the command field of an Entry Panel, with parameters in the input fields to identify the object to be addressed. If you leave the command line blank, the parameters entered in the input fields of an Entry Panel assume certain commands as default. For example, if you enter a PDS library name in the Dataset Name field of the PDS Objects Entry Panel and press ENTER, the assumed command is `LIST`. If you enter a PDS library name, followed by a member name in the Member field, the command defaults to `EDIT`. To list function commands relevant to the current Entry Panel, enter an asterisk (*) in the command field and press ENTER.
2. From the command field of any system screen, together with parameters to identify the object to be addressed. Used in this mode, function commands provide easy access to any object without having to leave the current screen. After the function has been performed, you can press PF3 to return to the screen from which the function command was issued.

   In this direct access mode, function commands take the format:

   ```
   COMMAND object-type object-parameters,function-parameters
   ```

   where:

Parameter-type	Meaning
object-type	Any Natural ISPF object type, for example: `P` for PDS member, `N` for Natural member, `MEM` for z/VSE member, etc.
object-parameters	Can consist of positional parameters and keyword parameters that correspond to the parameter fields on the Entry Panel of the object to be addressed.
function-parameters	Additional, function-specific parameters, for example a printer name in the `PRINT` command, a target source for `COPY`, a target node for `SUBMIT`, or target environment for `EXPORT` (PC or Con-nect), or a new name in the `RENAME` command.

   For a more detailed discussion of function command parameters, see the subsection ***Function Commands - Description*** in section *Command Reference*.

   Below is an example of the `LIST` command using this command syntax:

```
LIST P LIBRARY1 SCAN=ISP
```

The positional parameter `LIBRARY1` identifies the library to be listed, the keyword parameter `SCAN=ISP` lists only those members in the library which contain the string `ISP`;

Used in this mode, Natural ISPF assumes parameter values from the current environment. For example, if you issue the `LIST` command from the PDS facility, PDS is assumed as object type.

3. As line commands from lists of object names. It is common to select a required object from a list of object names. You can select an object for further processing by entering a line command in the input field preceding the object name on the list. Line commands are unique abbreviations of function commands and have the same effect as if the full function command were issued from the command line. Available line commands are described for each object type in the corresponding sections of this documentation.

   Note: Line commands can also be used as valid abbreviations of function commands entered in the command line of any screen.

Natural ISPF supports function command input with a powerful system of selection and prompt windows that help you select the right option or object if you leave out any required parameter from the command syntax. This makes the direct working mode using function commands easy even for less experienced users. Additionally, function commands assume default values from the environment in which they are issued. For example, the `LIST` command issued from the Main Menu displays a selection list of object types that can be listed; if you issue a `LIST` command from the PDS Objects Entry Panel, a window prompts you for the dataset name to be listed (see also the subsection ***Selection Windows and Wildcards***).

A full description of function commands, including examples, is contained in the section ***Command Reference***.

Local Commands - Definition

Local commands are commands that apply only to the object currently displayed in Editor format in `LIST`, `EDIT` or `BROWSE` mode. For example, when editing a Natural program, you can issue the `STOW` or `RUN` command, which are not available for any other object type. Local commands are described in detail for each object type in the section ***Natural Objects***, as well as in the section ***Command Reference***.

To obtain help information about available local commands in your current `EDIT`, `BROWSE` or `LIST` session, enter an asterisk (*) and press ENTER. A window opens that lists all available local commands.

PF Key Assignments

You can assign any Natural ISPF command, Editor main command, a sequence of commands separated by semi-colons (;), or an Editor line command preceded by a colon (:) to a PF key using the PROFILE option on the Natural ISPF Main Menu. Some standard functions are assigned to PF keys by default. These are listed below:

PF Key	Function (Command)
PF1	Enter the Natural ISPF online help system (HELP)
PF2	Enter split-screen mode (SPLIT)
PF3	Return to previous screen (END)
PF4	Suspend current session (SUSPEND)
PF5	Repeat previous Editor FIND command (RFIND)
PF6	Repeat previous Editor CHANGE command (RCHAN)
PF7	Scroll displayed data backwards (UP)
PF8	Scroll displayed data forward (DOWN)
PF9	Make other session active (SWAP)
PF10	Scroll displayed data left (LEFT)
PF11	Scroll displayed data right (RIGHT)
PF12	When in Editor, place cursor in command line (CURSOR)

If the PF-key line is displayed on your screen, you can use the FLIP command to choose between a display of PF1-PF12 or PF13-PF24.

If the PF-key line is not displayed on your screen, you can display it by issuing the session command KEYS ON.

You can modify any PF-key setting in your user profile in any of the following ways:

- Select the PROFILES option on the Natural ISPF Main Menu or issue the PROFILES command from any system screen to enter the profile maintenance facility;
- Issue the KEYS command without parameters to display the PF key modification screen in your user profile;
- Issue the KEYS command with parameters to directly assign a command to a PF key without entering the user profile. The command format is:

```
KEYS n string
```

where *n* is the number of the PF key and string is the command string to be assigned.

Examples:

- For example, if you issue the command:

```
KEYS 13 SEPARATE
```

  then every time you press PF13, the `SEPARATE` command is executed.
- The command:

```
KEYS 13 INITIAL
```

  resets PF13 to its initial value in the user profile.

Important

If the command string to be assigned consists of more than one command (for example: `SEPARATE;SPLIT`), you must use a double command delimiter to signal to Natural ISPF that this whole command sequence is being assigned. For example, to assign the command sequence `SEPARATE;SPLIT` to PF13, you must issue the command

```
KEYS 13 SEPARATE;;SPLIT
```

If you use only one command delimiter, the command `KEYS 13 SEPARATE` is executed first and then the command `SPLIT`.

You can also enter data in the command line prior to pressing a PF key assigned to a command. The command associated with the PF key is then concatenated ahead of the data entered in the command line to form the complete command.

For example, you may have the command `COMPRESS` assigned to PF16. If you enter a PDS library name in the command line and press PF16, the compress function is performed on the specified library. A common way to enter split-screen mode is to enter a line number in the command line and then press the PF key assigned to the `SPLIT` command (usually PF2).

Multiple Command Input

You can enter several commands in a single input operation if you separate them with the command delimiter specified in your user profile. They are then executed as if you had entered them individually. For example, the command sequence

```
SUSPEND;EDIT N MYLIB(MYPROG)
```

suspends the current session and starts a new edit session with Natural object `MYPROG` in the library `MYLIB`, if the semicolon (;) is the command delimiter specified in your user profile (see section *Profile Maintenance*).

Such command sequences can be assigned to PF keys for easy execution (see the subsection *PF Key Assignments*).

Multiple Editor Command Input

In the Natural ISPF Editor you can:

1. change data,
2. enter a line command, and
3. enter a main command.

   When you press ENTER, the commands are executed in the above order.

Example:

In the screen below, for example, if you:

1. insert the line `print #tamp` after line 40,
2. use the `R` line command to repeat line 40 (`WRITE #TEMP`) three times,
3. issue the main command `CHG TAMP TEMP ALL` to change all occurrences of `TAMP` to `TEMP`, and press ENTER.

```
EDIT-NAT:BRY(EDT-CMD)-Program->Struct-Free-47K ---------------- Columns 001 072
 COMMAND===> CHG TAMP TEMP ALL                                  SCROLL===> CSR
****** ****************************** top of data ******************************
 000010  DEFINE DATA LOCAL
 000020  1 #TEMP (A8)
 000030  END-DEFINE
 r30040  WRITE #TEMP
 ''''''  print #tamp
```

```
 000050  END
****** **************************** bottom of data ****************************
```

1. ... the line `print #tamp` is written,
2. line `40 WRITE #TEMP` is written three times and
3. all occurrences of `TAMP` are changed to `TEMP`:

```
EDIT-NAT:BRY(EDT-CMD)-Program->Struct-Free-47K ----------- Char  'tamp' changed
 COMMAND===>                                                   SCROLL===> CSR
****** ****************************** top of data ******************************
 000010  DEFINE DATA LOCAL
 000020  1 #TEMP (A8)
 000030  END-DEFINE
 000040  WRITE #TEMP
 000050  WRITE #TEMP
 000060  WRITE #TEMP
 000070  WRITE #TEMP
 ==chg>  print #temp
 000090  END
****** **************************** bottom of data ****************************
```

Issuing Commands by Magic Character

You can abbreviate any string used in Natural ISPF commands by assigning it to a magic character, usually a special character. If you enter this special character in the command line of any Natural ISPF screen, the effect is the same as if you had entered the associated string.

This feature makes entering long command sequences and multiple command input much more comfortable, for example when switching sessions, accessing other system functions or issuing operator commands. For example, if you assign the string `SUSPEND` to the exclamation mark (!), the command

```
!E NAT name
```

automatically starts a new Natural ISPF editing session with the specified Natural object. Or if you assign the stroke (/) to the string `OPERATOR_`, the command:

```
/C object=name
```

automatically executes the operator command `CANCEL` on the specified object.

Issuing command sequences by magic character can be an alternative to using PF keys if these are already assigned to other functions.

Note: Like any other command string, magic characters can be assigned to PF keys.

Selection Windows and Wildcards

When you enter function commands from the command line, Natural ISPF helps you choose the required object by providing selection lists and prompt windows when you leave out any required parameters from the command syntax.

Selection lists can be narrowed down by using certain wildcards for object parameters. You can enter wildcards in the parameter fields of object Entry Panels or as object parameters in function command syntax.

Two wildcard characters are available for all objects:

- An asterisk (*) selects object names with any character string;
- An underscore (_) selects object names with one character in the place of each underscore.

For Natural members, libraries, views and Con-nect documents, the following wildcards are also available:

- A less than sign (<) following a value lists all objects with lower values.
- A greater than sign (>) following a value lists all objects with greater or equal values.

(See examples below.)

Notes:

1. You can either use the asterisk, or a combination of asterisk(s) and underscore(s).
2. If you use either < or > as wildcard, these must be the last character in the parameter field and cannot be combined with asterisks or underscores. A single asterisk following < or > is accepted, but is redundant.
3. `*eoj` and `*cancel` are reserved words of Com-plete. If you use Natural ISPF with Com-plete and enter such a string as first input to be processed on a map, the input is ignored (`*eoj`) or Natural terminates (`*cancel`).

Examples:

- Enter a prefix followed by an asterisk (*) in the Member field to list all object names with that prefix. For example, in the Natural members facility, enter `ISP*` in the Member field and press ENTER to list all members in the specified library that start with `ISP`.

  The corresponding function command syntax issued from any system screen is:

  ```
  LIST N library(ISP*)
  ```

- Enter an asterisk (*) followed by a suffix in the Member field to list all object names with that suffix. For example, in the PDS members facility, enter `*INPL` in the Member field and press ENTER to list all members that end with `INPL` in the specified library.

  The corresponding function command syntax issued from any system screen is:

  ```
  LIST P library(*INPL)
  ```

- To list object names that contain a certain string in the name, you can use the notation *string* in the Member field. For example, `*ISP*` lists all objects that have the string `ISP` in their name. `A*B*` lists all object names that start with an A and also have a B in the name.
- To list object names with characters in certain places, use the underscore: for example, `A_ _E*` lists all object names with an `A` as first character and `E` as fourth character.
- The command:

  ```
  LIST DS *
  ```

  opens a window with a list of available library short names (see the subsection ***Library Definition*** in section *Profile Maintenance*).
- The command:

  ```
  LIST NAT C<
  ```

  lists all Natural members starting with `A` and `B`.
- The command:

  ```
  LIST NAT V>
  ```

  lists all Natural members starting with `V`, `W`, `X`, `Y` or `Z`.

4 Useful Features

- Handling and Navigational Features 26
- Functional Features 48
- Additional Features in Lists 83

This chapter describes special features that make work with Natural more comfortable.

The lists below present these features at a glance.

These features are described in more detail in the following subsections. For special features concerning command input, see section ***Command Logic***.

Handling and Navigational Features

The following topics are covered below:

- Working in Split-Screen Mode
- Multi-Session Operations
- Selecting a Function or Object
- ENTRY as Line Command
- Cursor-Sensitive String Selection - :C Directive
- Automatic Transfer of Commands
- Command Redisplay Feature
- Editor Sessions
- Editor Session with No Prefix
- Online Help Facility
- Online Technical and Site-Specific Information
- Confirmation Windows
- Using the Trace Function
- Break in Processing
- Automatic Screen Refresh

Working in Split-Screen Mode

Working with Natural in split-screen mode means dividing your screen horizontally into two sections and running a Natural session in each section.

You split your screen at the cursor position using the `SPLIT` session command (usually assigned to PF2). Alternatively, you can issue the `SPLIT` command with a line number as parameter. Your screen is then split at the specified line. If you only had one Natural session, another is automatically opened. If you had several sessions, the session last suspended is recalled to display.

You can change the portion of the screen devoted to each session by moving the cursor to where you wish to split the screen and repeating the `SPLIT` command.

The split-screen feature is useful for easy control of parallel sessions. For example, you could run a Natural program from an edit session in one screen session and immediately see the resulting output in a user workpool session in the other. Usually, only one session in split-screen is active.

To make the other session active, issue the `SWAP` command from the active session, or simply move the cursor to the inactive session and press ENTER.

If both sessions are edit sessions, both are active. This makes cross-session actions possible, for example, you can move or copy data from one session to the other. A common way to work with Natural is to run multiple sessions from your terminal with two sessions in split-screen mode.

Multi-Session Operations

Working in multi-session mode means starting several parallel Natural sessions. You can control up to 20 active Natural sessions from your terminal. You can work in two parallel sessions displayed on your terminal simultaneously using the split-screen and `SWAP` features.

You can suspend one session and start another at any time using the `SUSPEND` and `SPLIT` session commands.

Typical examples of multi-session operations are copying data from one edit session to another, and editing a Natural program in one session and checking the resulting output in another.

The following figure illustrates a screen with Natural in multi-session mode:

```
-------------------- NATURAL / NATURAL ISPF Main Menu ------------------------
>------------------------ NATURAL VIEW - ENTRY PANEL --------------------------
>>EDIT-PDS:FHI.JCL(ADAREP) ------------------------------------- Columns 001 072
 COMMAND===>                                                    SCROLL===> CSR
****** ****************************** top of data ******************************
000001 //HKADAREP JOB  SAG,CLASS=X,MSGCLASS=X
000002 //JOBLIB    DD   DSN=RZDBA.DB009.NEWLOAD,DISP=SHR
000003 //          DD   DSN=RZDBA.DB009.LOAD,DISP=SHR
000004 //REPORT    EXEC PGM=ADARUN
000005 //*******************************************************************
000006 //* DATABASE STATUS REPORT
000007 //*******************************************************************
000008 //DDASSOR1  DD   DSN=DB009.SYSF.ASSOR1,DISP=SHR
000009 //DDDATAR1  DD   DSN=DB009.SYSF.DATAR1,DISP=SHR
000010 //SYSUDUMP  DD   SYSOUT=X
000011 //DDDRUCK   DD   SYSOUT=3                      -> ADAREP MESSAGES
000012 //DDPRINT   DD   SYSOUT=*                      -> ADARUN MESSAGES
000013 //DDCARD    DD   DISP=SHR,DSN=NPRTAD.SYSF.DB009.CONTROL(CADAREP)
000014 //DDKARTE   DD   *                             <- ADAREP PARAMETERS
000015 ADAREP      NOFDT,FILE=66,67,68,70,74,75,76,77,78,79,84,85,86
000016 /*
****** **************************** bottom of data ****************************
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

The screen shows two suspended sessions, one containing the Natural ISPF Main Menu, the other a session with the `VIEWS` facility; two sessions are displayed in split-screen mode, the top session contains a PDS object editing session, the bottom session contains a list of Natural objects.

Natural provides a special feature here: the `SEPARATE` session command separates the current screen from its session, thus creating two sessions out of one. This feature is especially useful in conjunction with the `SPLIT` command. For example, after selecting an object from a list with a line command, you can issue the command sequence

```
SEPARATE;SPLIT
```

to display the object and the original list in separate sessions in split-screen mode.

Example:

If you enter the command `SEP;SPLIT 10` in the command line of the following screen, enter the line command `E` preceding the member `INPLISP` and press ENTER ...

```
S*>>LIST-PDS:FHI.ESM.JCL(*) --------------------- Row 0 of 41 - Columns 010 076
 COMMAND===> sep;split 10                                      SCROLL===> CSR
   MEMBER            VV.MM  CREATED  MODIFIED TIME  SIZE   INIT   TID   ID
** ******************************** top of list ******************************
   ADACOP            01.12  19930430 19940929 13:38    170     44       HKA
   ADAREP            01.04+ 19941017 19980910 13:36     16     14       FHI
   CATDEV            01.26  19911202 19960604 19:16     32     19       HKA
   CMPLOAD           01.02  19900830 19931213 10:34      6      6       HKA
   DELENV            01.16  19910904 19970227 13:06     23     19       HKA
   ERRLODU           01.04  19911202 19970221 17:01     16     19       HKA
   FHITST            01.01  19970919 19970919 13:51      1      1       FHI
   FNATBACK          01.03  19971124 19971124 15:37     31     31       FHI
   IBGVSE            01.30  19950222 19960909 18:30     27     21       HKA
   IEBGENPQ          01.15  19950310 19971104 13:36     34     26       FHI
   IEBGNIMP          01.04  19950310 19970711 14:14     30     26       MZC
   IEBGNOWN          01.04  19970717 19971113 14:10     19     19       FHI
   IEBGNPQ2          01.25  19950310 19971104 15:06     63     26       FHI
   INPL              01.20  19910418 19970221 16:17     47     17       HKA
 e INPLISP           01.12  19971126 19971203 16:24     35    119       FHI
   INPLSCAN          01.12+ 19980212 19980603 10:50     16     17       FHI
   IUPD              01.01  19970221 19970221 16:37     18     18       HKA
   LDDMCT            01.06  19920601 19930125 13:02     23     23       HKA
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

... the requested edit session with member `INPLISP` is displayed in a separate session in split-screen mode, up to Line 10:

```
S*>>>EDIT-PDS:FHI.ESM.JCL(INPLISP) ---------------------------- Columns 001 072
 COMMAND===>                                                    SCROLL===> CSR
****** ****************************** top of data ******************************
000001 //FHI  JOB FHI,MSGCLASS=X,CLASS=G,REGION=2500K
000002 //INPLISP EXEC PGM=NATBAT22,
000003 //   PARM=('DBID=9,FNR=40,FUSER=(,121),FSEC=(,16),IM=D,MT=0,MADIO=0',
000004 //         'INTENS=1,MAXCL=0,AUTO=ON,FDIC=(,122)')
000005 //STEPLIB DD  DSN=OPS.SYSF.PROD.LOAD,DISP=SHR
000006 //*       DD  DSN=OPS.SYSF.V5.ADALOAD,DISP=SHR
000007 //CMPRINT DD  SYSOUT=X
--------------------- NATURAL / NATURAL ISPF Main Menu ------------------------
OPTION  ===>
                                                              User ID  VMU
                                                              Time     14:34:09
 _ 1    NATURAL    - Development Functions                    Terminal DAEETCN8
 _ 2    NATURAL    - Development Environment Settings         Library  SYSISPHU
 _ 3    NATURAL    - Maintenance and Transfer Utilities       Node     148
 _ 4    NATURAL    - Debugging and Monitoring Utilities
 _ 5    NAT/NSPF   - Example Libraries

 _ 6    SAG        - SAG    Products
 _ 7    SYSTEM     - System Products
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Notes:

If the limit of 20 parallel sessions is reached and the user tries to open another session, the message: `No more sessions available` appears. If this message is received in a situation where fewer than 20 sessions are visible on the Activity Table (see description for the `ACTIVITY` command on the following page) consider the following:

1. Natural keeps a stack of sessions with 20 entries for each half (upper and lower) of the split screen.
2. For every menu level, one stack entry is created. This means that, if the user opens a session via the menu, the following entries are created:
   - Main Menu
   - Entry Panel
   - Object list
   - `EDIT` or `BROWSE` session

     Thus, for each “visible” session, four entries are required, so that the limit is reached with only five “visible” sessions.
3. The number of stack entries can be reduced by using direct commands.

The following commands are available to control multi-session mode of work:

Command	Explanation
ACTIVITY	Displays list of all your Natural sessions. The current session appears highlighted. You can select another session using the P line command, terminate a session by entering a hyphen (-) in the line command field, or SWAP sessions using the S line command. In the Name column of the list, you can enter a short name for the session for easy identification in a later POP command. Alternatively, you can enter the name of a PF key in the Name field. You can then press this PF key at any time to call the associated session (existing PF key definition is overridden).
POP	Calls the indicated session to display. The popped session appears in the section of your screen in which you entered the POP command. The required session is indicated: ■ by placing the cursor on the session header line, or: ■ by adding the session name as a command parameter (you can assign short names to sessions using the ACTIVITY command), or: ■ by adding any string from the session header line as a parameter If an unqualified POP command is issued, a selection window with available sessions is opened.
SEPARATE	Separates the current screen from its session to split the session into two. Useful in conjunction with a subsequent SPLIT command to display two screens from the same original session.
SPLIT	Splits the screen to display two Natural sessions. If only one session is active, a new one is created. The screen is split at the line specified in the command parameters or at cursor position. The new session is started in the lower part of your screen. You can change the size of the screen portions devoted to each session by repeating the SPLIT command with an appropriate line number or the cursor position indicating the required split position.
SUSPEND	Suspends the current session and starts a new one: the suspended session is eliminated from display except the top line.
SWAP	When working in split-screen mode, makes the other session active: the cursor is automatically moved to the other session. When working in full-screen mode, calls another session to display in full-screen mode in wrap-around fashion.
UNZOOM	Reverses the previous ZOOM command.
ZOOM	Enlarges the current session to full screen. If any suspended sessions exist, their top lines are eliminated from display. In split-screen mode, the current session is enlarged to full screen. If you are in split-screen mode and have the top lines of suspended sessions displayed, two ZOOM commands are required (one to remove the header lines, one to enlarge the current session).

Additionally, the Editor provides some line commands that allow you to copy or move lines of data from one edit session to another. You can mark the line or lines you wish to move or copy in one edit session and mark the target line in another edit session.

The available line commands to mark the source lines are:

Command	Explanation
C	Copies this line
CC	Marks the first and last line of a block of data to be copied
M	Moves this line
MM	Marks the first and last line of a block of data to be moved

The available line commands to mark the target area are:

Command	Explanation
A	Inserts data after this line
B	Inserts data before

Selecting a Function or Object

Natural makes it easy for you to select a function or object by providing an extensive system of prompt windows.

- To select a function, simply enter an asterisk (*) in the command line of the current screen. This opens a window with available functions. Select one by entering it in the input field of the window;
- To select a function for a particular object type, simply enter the object type in the command line of any system screen. This opens a window with all available functions for the object type.

  Note: When requesting a list of functions for Natural objects, use the shortest possible abbreviation N. If you use NAT or NATURAL, it is interpreted as a logon request to Natural.

- To select an object for a particular function from the Main Menu, simply enter the function command in the command line of the Main Menu. This opens a window with available object types for the function.

See also the section *Command Logic*.

ENTRY as Line Command

The ENTRY command displays the Entry Panel for the specified object type:

```
ENTRY object-type
```

The ENTRY command is also available as line command from a list of objects. When issued as a line command, the Entry Panel of the object type is displayed, with the parameter fields filled with the values of the selected object. This is especially useful when working with objects that have long names (for example, datasets, LMS elements, Con-nect documents), and you wish to create a new object with a name similar to the selected object.

Example:

You have requested a list of all elements in the LMS library $ASF.LMSLIB.T using the command

```
LIST LMS :D:$ASF.LMSLIB.T(*)
```

and wish to create the new member NOP.M.NAT. Simply enter the EN line command for the member NOP.P.NAT as follows:

```
LIST-LMS::D:$ASF.LMSLIB.T(*) -------------------- Row 0 of 13 - Columns 026 076
 COMMAND===>                                                SCROLL===> CSR
   TYP ELEMENT-NAME                  S V-C HIGH-VERSION     CREATED    MODIF-DA
** ******************************** top of list ******************************
   (J) DOC                           V   1 003              1991-05-29 1991-05-
   (J) E.JOB.TEST.001                V   1 001              1991-03-27 1991-08-
   (J) JOB                           V  +3 003              1990-11-29 1991-01-
EN (J) NOP.P.NAT                     V   1 §                1990-11-29 1991-01-
   (J) OSOSOSOSOSOSOYSOS             V   1 0045005          1991-07-18 1991-07-
   (J) YYY                           V   1 003              1991-07-18 1991-07-
   (J) ZZZHUNZEL                     V   1 999              1991-12-19 1992-11-
   (S) DOC                           V  +3 003              1990-11-29 1991-07-
   (S) DOC.NEW                       V   1 §                1990-11-29 1991-06-
   (S) DOC-S                         V   1 001              1991-07-16 1991-07-
   (S) DOC-VERS                      V   1 LONGVERSION01    1991-07-09 1991-07-
   (S) NEW-MEMBER                    V   1 001              1991-06-17 1991-07-
   (S) ZZZ                           V   1 003              1991-07-18 1992-11-
** ****************************** bottom of list ******************************




Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right :s
```

The LMS Elements - Entry Panel is displayed with the parameter fields filled:

```
------------------------ LMS ELEMENTS - ENTRY PANEL ---------------------------
COMMAND ===>


File Name  => :D:$ASF.LMSLIB.T
Element    => NOP.P.NAT
Type       => J                        ( S,M,J,D,H,P,R,C,X or combination   )
Version    =>
Rec.length =>                          ( If different from default (80/133) )
Password   =>                          ( If library is password protected   )
Scan for   =>
Edit macro =>
Node       => 31









Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right :s
```

All you need to do is change the member name to `NOP.M.NAT` and press ENTER to start the required edit session with the new member.

Cursor-Sensitive String Selection - :C Directive

In the Editor and any other Natural Session

Natural provides a feature which makes input of commands in the command line more comfortable.

In any session (`LIST`, `EDIT` or `BROWSE` mode), command input can consist of typing in command keywords plus any word selected from the data displayed. The word is selected by typing `:C` in the command line and placing the cursor on the first character of the required word. When you press ENTER, the `:C` notation is substituted by the selected string, and the completed command is executed.

Example of :C Notation:

```
>>EDIT-PDS:MBE.SYSF.SOURCE(NOJC02)----------------------------- columns 001 072
 COMMAND ===> edit :C
00010 //MOVE A TO B
000020 //INCLUDE MYPROG
```

With the command

```
EDIT :C
```

in the command line, move the cursor to the string `MYPROG` and press ENTER. The command executed is

```
EDIT MYPROG
```

Note: You might find it useful to assign the command `SUSPEND;E :C` to an available PF key.

In the Online Help System

When you are scrolling help texts in the online help system, you can enter `HELP :C` in the command line, place the cursor on a word and press ENTER to see the help text on that word, if available. Assigning the `HELP :C` directive to a PF key makes this quick cross reference to other help texts even more comfortable.

Automatic Transfer of Commands

To make using Natural even easier, commands are accepted in many situations, although they are syntactically not quite correct. Such commands are transferred to another object depending on the context, thus saving the user additional entries or steps. Two examples follow:

From PDS to Datasets

You can access partitioned and sequential data sets by using two Natural facilities, the PDS objects facility and the Datasets facility.

If you wish to maintain PDS members using function command syntax from outside the PDS objects facility, you must address Object Type `P`. If you wish to access datasets with function command syntax, you must address Object Type `D`.

However, if you are in the PDS facility, you can use a command pertaining to datasets. The command is transferred automatically (and invisibly to you) to the Datasets facility and executed there. The command is:

```
COMPRESS
```

This feature provides added convenience when, for example, you wish to save a modified PDS member and a message tells you that the dataset is full.

From Active Jobs to Job Information

If you select a job from a list of active jobs with the `B` line command (`BROWSE`), you are automatically transferred to the job information facility.

Command Redisplay Feature

In any Natural Screen

With the `LAST` command, you can retrieve the last 10 (ten) commands you entered. A window opens with a list of the commands. You can select any command for reexecution by placing the cursor on it and pressing ENTER. You can modify the command by overtyping it before execution. This also applies to Editor command strings.

If a command is executed again, it is always put on top of the last command buffer. This keeps the commands used most in the buffer. Additionally, you can also delete commands from the last buffer. Commands not used again can be deleted; this avoids automatic deletion of commands likely to be reused. You can delete the command from the buffer by setting the entire line to blanks and pressing ENTER.

Only commands consisting of two words or more and entered via the keyboard are stored and appear in the window: commands executed by PF key, magic character or by selection from a prompt window are not stored.

In the Editor

Additionally, any Editor command entered in the command line of an Editor screen (`LIST`, `BROWSE`, `EDIT`) remains in display after execution if preceded by an ampersand (&). This applies to Editor commands only.

Example:

You may find this feature useful, for example, in combination with the commands `RFIND` (to scan for all occurrences of a search string) or `XSWAP` (to exchange displayed lines and excluded lines).

Editor Sessions

The Software AG Editor is integrated in Natural. Certain commands issued from a Natural screen automatically start an Editor session: `EDIT`, `BROWSE`, `LIST`. This means that lists of objects, and objects displayed in edit or browse mode constitute Editor sessions and you can use any appropriate Editor command.

In `LIST` or `BROWSE` mode, you can therefore use any Editor command that does not change the data. This includes not only scroll commands (`UP`, `DOWN`, `BOTTOM`, `TOP`, `LEFT`, `RIGHT`, etc.), but also other commands such as `FIND`, `CREATE`, `HEX`, etc., corresponding Editor line commands, as well as the C/CC line commands.

This means, for example, that you can copy lines from a list into another member in a cross-session copy operation, or store a list in a member as data that can be edited.

Example:

You wish to store a list of members from Natural library `MYLIB` as a PDS member. From the Main Menu, issue the command:

```
L N MYLIB(*)
```

This displays the required list on your screen. In the command line of the list, enter the command:

```
CREATE P USER.LIB(LIBLIST)
```

and mark the first and last lines of the block to be stored with a `CC` line command as follows:

```
  LIST-NAT:MYLIB(*) ------------------------------- Row 0 of 7 - Columns 010 076
   COMMAND===> create p user.lib(liblist)                        SCROLL===> CSR
     MEMBER            PGMTYPE     SM S/C VERSION   USERID   DATE     TIME  VV.MM
  ** ******************************** top of list ******************************
     ISPNHVKN          Subprogram  S  S/C 3.1 0006  UKSJU    20030319 16:02
  CC ISPNIN-N          Subroutine  S  S/C 3.1 0006  JWO      20020904 16:56
     ISPNIN-P          Program     S  S/C 8.2 0001  BLI      20110112 15:00 01.09
     ISPNIN-1          Map         R  S/C 8.2 0001  BLI      20101208 16:59
     ISPNLDRP          Program     S  S/C 4.1 0004  FHI      20060320 10:04
     ISPNLS-L          Local          S/C 8.2 0001  BLI      20101210 18:48
  CC ISPNLS-P          Program     S  S/C 8.2 0001  BLI      20110131 18:27 01.25
  ** ****************************** bottom of list ******************************









 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
       Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

When you press ENTER, the marked list is created as member `LIBLIST` in PDS library `USER.LIB` (a message confirms successful creation).

Editor Session with No Prefix

You can work with an Editor session in `EDIT` or `BROWSE` mode without the prefix area (line numbers) displayed. The data area is then increased by 7 columns. This feature is controlled by the `PREFIX` session command.

The command format is:

```
PREFIX  [ON]
        [OFF]
```

The default value is `ON`. When the prefix area is `OFF`, you can issue line commands by typing them in the data area preceded by the escape character.

Online Help Facility

Natural includes a comprehensive online help system which provides information about the Natural facilities and helps you select the appropriate function, command or item from any system screen.

The online help system consists of active help and passive help:

- **Active Help**

  Lists selectable options in windows and is invoked by entering an asterisk (*) in input fields or instead of object parameters in function command syntax (see also the subsection ***Selection Windows and Wildcards*** in section *Command Logic*);

- **Passive Help**

  Displays general information on the selected item and is invoked

  - at **screen level** by entering the `HELP` command in the `COMMAND==>` or `OPTION==>` field and pressing ENTER;
  - at **field level** by entering a question mark (?) in the appropriate field and pressing ENTER;
  - at **message level** by entering the `HELP` command followed by message number in the `COMMAND==>` or `OPTION==>` field.

The following subsections describe how to use the active and passive help systems.

Active Help

Active help always offers you a selection list of Natural commands, functions or objects. If you select an item from the list, Natural prompts for missing parameters/options for the selected function until the command can be executed.

Active help is usually invoked by entering an asterisk (*) in the command line and pressing ENTER. The type of information displayed depends on the type of screen where you invoke active help.

The following subsections describe four examples of active help (in a Menu, Entry Panel, `LIST` and Editor session) in more detail.

Natural Menu

If you invoke active help in a menu, it lists all available function commands at your site. If you select a function from the list, you are prompted for the object (e.g. Natural or PDS) and then for missing parameters.

Below is an example of an active help screen invoked from the Natural ISPF Main Menu:

```
--------------------- NATURAL / NATURAL ISPF Main Menu ------------------------
OPTION  ===> +---------------------------------------------------+
             ! ENTER FUNCTION:                                   ! ID  VMU
            !   1 L  LIST              17 ET EXTENTS          !     14:11:05
 _ 1    NATU !   2 B  BROWSE            18 FL FOLLOW           ! nal DAEETCK7
 _ 2    NATU !   3 E  EDIT              19 ST STATUS           ! ry  SYSISPHU
 _ 3    NATU !   4 D  DELETE              20 CC COND.CODES        !     148
 _ 4    NATU !   5 R  RENAME              21 RL RELEASE           !
 _ 5    NAT/ !   6 SB SUBMIT              22 HL HOLD              !
             !   7 PL PLAY                23 DI DIFFERENCE        !
 _ 6    SAG  !   8 PR PRINT               24 PG PURGE             !
 _ 7    SYST !   9 CP COPY                25 CH CHANGE            !
             !  10 CM COMPRESS            26 DF DEFINITION        !
 _ 8    NSPF !  11 ZP ZAPS                27 RU RUN               !
             !  12 XT EXTERNS             28 XE EXECUTE           !
 _ 9    NSPF !  13 A  ALLOCATE            29 EX EXPORT            !
             !  14 CT CATALOG             30 DS DESCRIPTION       !
 _ HELP HELP !  15 U  UNCATALOG           and others...           !
 _ NHLP HELP !  16 I  INFORMATION         Select ==> __           !
 _ END  EXIT +---------------------------------------------------+


Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

To display an explanation of a listed function, enter the function's number or code in the `Select ==>` field and press PF1. This invokes the passive help system.

Natural Entry Panel

If you invoke active help from an Entry Panel, a list of all available function commands for the selected object is displayed. If you select a function from the list, it checks whether the fields in the Entry Panel already contain a sufficient object identification; if not, you are prompted for the missing parameters.

Below is an example of an active help screen invoked from the Natural View - Entry Panel:

```
------------------------ NATURAL VIEW - ENTRY PANEL --------------------------
COMMAND === +--------------------+
            ! ENTER FUNCTION:    !
            !   1 L  LIST        !
   View Nam !   2 B  BROWSE      !
   Dbid     !   3 CP COPY        !                      (For selection list)
   Fnr      !   4 DF DEFINITION  !                      (For selection list)
   Record F !   5 DS DESCRIPTION !
   Start va !   6 DW DOWNLOAD    !
   End valu !   7 UP UPLOAD      !
   Max Reco !  Select ==> __     !
   Password +--------------------+
```

To display an explanation of a listed function, enter the function's number or code in the `Select ==>` field and press PF1. This invokes the passive help system.

Line Command Help in LIST Session

Invoking active help in a list usually displays the same list as invoked from an Entry Panel.

Natural Editor screen

If you invoke active help from an `EDIT`/`LIST`/`BROWSE` screen, all available local commands for the selected object and function are displayed. If you select a function from the list, the function is executed.

Use PF7 / PF8 for scrolling if more than 10 commands are available, or PF3 to abort. If you press ENTER without any selection, a list of all function commands is displayed.

Do not forget, if you need information about Editor commands just enter: `HELP EDITOR`.

Below is an example of an active help screen invoked from an edit session for a Natural program:

```
EDIT-NAT:NSPFEXAM(API-002P)-Program->Struct-Free-45K ---------- Columns 001 072
 COMMAND===> *                                                  SCROLL===> CSR
****** ****************************** top of data ******************************
 000010 DE +-------------Local command help-------------+
 000020 LO !                                             !
 000030 *  !  1  SM                6  TYPE               !
 000040 *  !  2  STOW              7  STRUCT             !
 000050 *  !  3  CHECK             8  OUTPUT             !
 000060 1  !  4  CAT               9  IMPORT             !
 000070 1  !  5  RUN               10 PLAY               !
 000080 1  !                         more...             !
 000090 *  !  Select  __ or press enter for global help !
 000100 *  +-----------------------------------------+
 000110 *
 000120 1 #ERR-NUMBER (N4)
 000130 1 #ERR-TEXT   (A75)
 000140 1 #ERR-PARM   (A75)
 000150 *
 000160 * local data
 000170 *
 000180 1 #LGT       (N2)
 000190 1 #IN-DSN    (A44)
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

To display an explanation of a listed function, enter the function's number in the Select field and press PF1. This invokes the passive help system.

Passive Help

Passive help texts are available on three levels:

- Screen level;
- Message level;
- Field level.

The following subsections describe the three levels of passive help in more detail.

Screen-level Help

The hierarchy of the help screens reflects the hierarchy of Natural system screens. This means that you can invoke the online help facility from any system screen and you immediately see the help text defined for that screen. Invoke a screen help by issuing the `HELP` command (usually assigned to PF1) from any system screen, or by typing a question mark (?) in the command line and pressing ENTER. This displays a help text with general information on the screen from which you invoked the online help.

The help text displayed can be one of two types:

- User-defined help;
- Screen-dependent help text from the online help facility.

The user-defined help can be linked to a menu customized to your installation and is maintained in a separate library by the system administrator (see also the *Natural ISPF Administration Guide*). If you issue the `HELP` command from any menu, the user help library is searched before the system help.

You can enter the online help facility at the top end of the hierarchy by issuing the `HELP` command (usually assigned to PF1) from the Natural ISPF Main Menu. This displays the main Help Menu on your screen with a list of selectable items:

```
 Z*HELP-----------------------Table-of-contents------------------>>> continued
 COMMAND ==>                                                          MAIN

                            NATURAL-ISPF HELP
      Please select a topic by entering a numerical code

 PART I: Working with NATURAL ISPF

       1  Help system
       2  Command Logic
       3  NATURAL ISPF Commands
       4  Useful Features

 PART II: Common Functions

       5  User Workpool
       6  Recovery Files
       7  Profile Maintenance
       8  NATURAL ISPF Editor

 PART III: NATURAL

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

The Help Menu can consist of several screen pages. Use the `UP` and `DOWN` commands (usually assigned to PF7 and PF8) to scroll the menu.

To select an item, enter its number in the command line at the top of the menu and press ENTER. This displays a help text for the selected item.

The `HELP` command is also available with a number of parameters to allow you to call help texts on any Natural item from any system screen.

```
HELP  [INDEX]
      [object]
      [:C]
```

where:

Parameter	Meaning
INDEX	Lists all objects for which there is a help text. You can specify any object in a HELP command to display the related help text.
object	The object for which a help text is required. This can be any object listed by the HELP INDEX command. Use the asterisk wildcard (*) to generate selection lists of objects with the same prefix (see examples below).
:C	Is substituted by the string marked by the cursor (see the subsection ***Cursor-Sensitive String Selection***). If you use the :C directive, you need not enter the HELP command keyword.

Examples:

Command	Meaning
HELP LIST	Displays the help text for the command LIST.
HELP L*	Displays a selection list of all items and commands starting with L for which there is a help text. You can select any item from the list to display the associated help text.
HELP FEATURES	Displays help text for special Natural features.
HELP EXAMPLES	Displays an overview of example programs, macros and command scripts supplied in the Natural Examples Library.
HELP :C	With cursor on string "edit", executes the command HELP EDIT.

A help text may consist of more than one page. If this is the case, you are notified by the message Continued.... Use PF7 and PF8 to browse though the help pages. A special local command is also provided here. The command UP L or BACK scrolls up to the next higher level in the screen hierarchy. Alternatively, you can enter L in the command line and press PF7.

Some help screens provide direct access to help texts on a related item or on another level in the help screen hierarchy. You can enter the selected item in the command field at the top of the help screen and press ENTER to see the associated help text.

Alternatively, you can enter HELP :C in the command line, place the cursor on a word in the help text and press ENTER to display the help text on that word. Assigning the HELP :C directive to a PF key makes this cross reference even more comfortable (see the subsection ***PF Key Assignments*** in section *Command Logic*).

To print a help text as displayed on the screen, enter the PRINT command and press ENTER.

Note: Words which appear in the help text in reverse video (on color terminals, yellow) have separate help texts defined for them. Place the cursor on the word and press the help key to display the corresponding help text.

To leave the online help facility, issue the `END` command (usually assigned to PF3).

Message-level Help

The online help facility also allows you to display information on error messages. This is done using the following command from any Natural screen:

Command	Meaning
`HELP` *nnnn*	Displays a help text for error message Number *nnnn*. This can be any existing error number of Natural, or any other related Software AG product installed at your site.

If a Natural or Natural error occurs, a special feature is provided that enables you to issue a `HELP` command without parameters. This displays a help text on the last error.

Field-level Help

You invoke a field help from a system screen by typing a question mark (?) in the selected input field and pressing ENTER. This opens a window with an explanation of the field and how to use it. Press ENTER to close the window.

Online Technical and Site-Specific Information

Natural provides some technical information online relevant to the system administrator for system identification and debugging purposes, as well as a facility with which the system administrator can provide site-specific information.

The `TECH` command opens a window with information that includes the following information:

- Current versions of Natural ISPF, the SAG Editor and Entire System Server
- Values of the `ASIZE` and `SSIZE` parameters
- Library, current date and time, terminal ID, TP monitor and operating systems

The `TECH ALL` command shows, in addition to above described `TECH` information, details about installed hotfixes and corrections.

The `UINFO` command displays site-specific information maintained by the system administrator. This information may consist of several screens. Navigation among Information screens is done by PF key, scroll commands or topic selection, as directed in the screens.

Confirmation Windows

When you wish to perform certain functions on a Natural object using function commands (e.g. DELETE, COMPRESS), a window is opened in which Natural asks you to confirm the function by entering the object name or Y. You can back out the request simply by leaving the input field in the window blank and pressing PF3.

You can suppress the confirmation feature with the command

```
CONFIRM  OFF
```

for example when deleting objects from a list using multiple D line commands in a single input operation. The message CONFIRM OFF is displayed in the message line. The function is then performed without confirmation. The confirmation feature is automatically reactivated when you invoke another system screen, or you can issue one of the following commands:

```
CONFIRM LONG
CONFIRM SHORT
```

where LONG specifies confirmation by name input and SHORT specifies confirmation by entering Y or N in the confirmation window.

The default is CONFIRM SHORT and can be modified in your user profile using the PROFILE option on the Natural Main Menu (see the subsection ***User Defaults*** in the section *Profile Maintenance*). See also the description of the CONFIRM session command in the section ***Command Reference***.

Using the Trace Function

Natural provides a feature that allows you to trace Natural activity while CPU time is being consumed. After a command is issued and a predefined time interval has elapsed before the next screen output, a window opens informing you of Natural activity. The message is updated after each interval of the specified length until the function is completed.

The interval after which the trace window appears is specified in your user profile. You can modify this interval using the TRACE command.

Examples of the TRACE command:

Command	Meaning
TRACE 10	If a function takes more than 10 seconds to complete, the trace window appears after 10 seconds of elapsed time. The window is updated every 10 seconds.
TRACE 00	Trace function is disabled.
TRACE OFF	Trace function is disabled.

For example, if the trace interval is set at 3 (TRACE 3 command), and you issue a LIST command with the SCAN option in the PDS facility (that is, only those members are listed which contain the

string specified with the SCAN keyword), a message similar to the following appears in a window if after 3 seconds the function is not completed:

```
PROCESSING   4 MEMBERS
```

The message is updated every 3 seconds until the function is complete and the requested list is displayed.

Break in Processing

The BREAK function allows you to abort Natural activity if it takes too long or if it has been started by mistake.

In addition to the TRACE value (see above), you can define a BREAK value in your user profile. This causes Natural to prompt you whether to abort or continue with the current function. This feature is available with the LIST function.

For example, if you set the BREAK value at 3 and you issue a LIST command that reads a large number of data, you are prompted to interrupt processing if after 3 trace windows the function has not completed:

```
PROCESSING   104 MEMBERS
BREAK PROCESS(Y/N) _
```

You can abort the function by entering Y in the input field. N continues the function.

The BREAK value can be modified either directly in your user profile or using the BREAK session command:

Command	Meaning
BREAK 2	You are prompted to abort the function every 2 trace windows.
BREAK 00	Break function is disabled.
BREAK OFF	Break function is disabled.

Automatic Screen Refresh

Refreshes the display of the current screen.

The command format is:

```
REFRESH nn
```

The screen is refreshed automatically every *nn* seconds until the user presses the attention interrupt key, to return to normal command input mode. This is especially useful if a user wants to monitor active jobs or browse the console/log without pressing ENTER.

This function is not supported in all TP environments (i.e. IMS-TM).

Example:

For example, if you enter the command `REFRESH 2` in the command line of the following console screen and press ENTER ...

```
BROWSE-CON:/NODE=148/TYPE=ALL --------------------------------- Columns 001 076
 COMMAND===> REFRESH 2                                      SCROLL===> CSR
   JOB03635  -KHKLOAD  ENDED.  NAME-                     TOTAL TCB CPU TIME=
    TOTAL ELAPSED TIME=   1.2
   JOB03635  .HASP395 KHKLOAD  ENDED
   JOB03640  .HASP373 KHKLOAD  STARTED - INIT  1 - CLASS G - SYS DAEF
   JOB03640  IEF403I KHKLOAD - STARTED - TIME=16.34.46
   STC03689  +ESY007I INACTIVE USER          ESYID 0335 HAS BEEN PURGED
   STC03689  +ESY007I INACTIVE USER          ESYID 0336 HAS BEEN PURGED
   STC03689  +ESY007I INACTIVE USER          ESYID 0337 HAS BEEN PURGED
   STC02112  ACF01200 INVALID PASSWORD/AUTHORITY FOR ID HGS FROM
   STC02112  ACF01012 PASSWORD NOT MATCHED
   STC02346  NETT004I: 06 - DIS-CONN       Admin REQ From Node PCBTO    Receive
   STC02346  NETP107W: Trecv    API Call Failed, R15=00000004 R0=00000008 For P
     157.189.161.138
   STC02346  NETP102W: Error Code=TERELESE (Connection Release Received   )
   STC02346  NETP001W: Error During Receive RC=(08) Link=LNKPCBTO
   STC02346  NET0107I DAEFNODE: Link LNKPCBTO disconn. from node PCBTO
   STC02085  ESY007I INACTIVE USER SAGEJE   ESYID 0719 HAS BEEN PURGED
   JOB03642  .HASP100 ECKLOD   ON INTRDR                         FROM STC02077
    DAEFCO
** ****************************** bottom of list ******************************
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

... the screen is refreshed every 2 seconds:

```
BROWSE-CON:/NODE=148/TYPE=ALL ---------------------------------- Columns 001 076
COMMAND===>                                                      SCROLL===> CSR
  JOB03644  -KAS      ADAFRM              00   1788   3682    .00    .00
   10210   4     0     0     0     0
  JOB03642  -ECKLOD   LD       ADARUN     00    311    776    .00    .00
   3315   1     0     0     0     2
  JOB03642  IEF404I ECKLOD - ENDED - TIME=16.36.17
  JOB03642  -ECKLOD   ENDED.  NAME-                    TOTAL TCB CPU TIME=
   TOTAL ELAPSED TIME=    .3
  JOB03642  .HASP395 ECKLOD   ENDED
   .HASP309    INIT  2 INACTIVE ******** C=G
  STC03754  IEC130I SYSOUT   DD STATEMENT MISSING
  STC03754  +WER045C NSNSERV ,NSNSERV ,         - END SORT PH
  STC03754  +WER177I NSNSERV ,NSNSERV ,        -TURNAROUND SORT PERFORMED
  STC03754  +WER055I NSNSERV ,NSNSERV ,        -INSERT         6, DELETE
  STC03754  +WER246I NSNSERV ,NSNSERV ,        -FILESIZE 42 BYTES
  STC03754  +WER054I NSNSERV ,NSNSERV ,        -RCD IN         0, OUT
  STC03754  +WER169I NSNSERV ,NSNSERV ,        -TPF LEVEL 3B
  STC03754  +WER052I NSNSERV ,NSNSERV ,        -END SYNCSORT - NSNSERV,NSNSERV
   DIAG=CC00,8C08,227C,E49D,CD7E,8C09,226C
  STC03479  TMN001C DAEFC0 uses 61.26% CPU --> from TMON
** ****************************** bottom of list ******************************
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Functional Features

The following topics are covered below:

- Versioning
- Recovery
- Copying Objects
- Printing Objects
- Natural Interface to External Environments
- Message Switching
- Using the Calculator
- Using the Macro Facility

- Executing Command Scripts

Versioning

Natural can keep previous versions of the following object types after they have been edited and saved (`SAVE` command) or stowed (`STOW` command, for Natural programs):

- Natural members (but not for maps and data areas)
- PDS members
- z/VSE members
- CA Librarian members
- LMS elements

Note: This section describes versioning of Natural, z/OS and z/VSE objects only. Versioning of LMS elements is described in the subsection ***LMS Library Elements*** in section *BS2000 Objects*. Versioning for CA Librarian members is described in the subsection ***Previous Versions*** in section *CA Librarian Members*.

For a previous version of an object to be available, versioning must be enabled, or one of the following conditions must be true:

- The member must already have previous versions, or
- The library is specified in the versioned library list maintained by the system administrator.

Versioning is enabled at the user level in the user defaults using the `PROFILE` option on the Natural Main Menu, and/or can be activated or deactivated for the session using the `VERSIONS ON/OFF` command.

How many versions of a member are to be stored is specified by the system administrator in the NSPF Parameters by selecting the `CONFIG` option from the Administrator Menu and the NSPF option from the Configuration Menu. This value can be overridden for a member from the Information screen for the member. The system administrator also specifies the database number and file number where versions are kept.

Notes:

1. If the maximum number of versions is reached, a subsequent `SAVE` or `STOW` command automatically deletes the oldest version not on `HOLD` (see the `HOLD` command below);
2. If you rename a member, all versions are renamed;
3. If you replace a member using the `COPY` command, all previous versions are deleted.
4. The command `VERSIONS OFF` will affect only members that have not yet been versioned. If `VERSIONS OFF` is issued, a member with existing versions will be still versioned unless all versions of the member are deleted.

Retrieving Previous Versions

Previous versions are separate object types that can be addressed using the following object type notations in function commands:

Notation	Meaning
NV	Natural object versions
PV	PDS object versions
VV	z/VSE member versions
MV	all object versions

You can retrieve previous versions of a member in any of the following ways:

1. If you have administrator status, using the VERSIONS option on the Administrator Menu.
2. From the object's Entry Panel, issue the LIST command with the library and member names specified. This generates a list of versions for the member.

   Note: You must explicitly enter the LIST command; if you leave the command line blank, the command defaults to EDIT, and an edit session with the last version is started.

3. From any system screen using the LIST or BROWSE function command in any of the following formats:

   ```
   LIST N/P/LIB/LMS LIBRARY(MEMBER)
   LIST MEM LIBRARY.SUBLIBRARY(MEMBER.MEMBER-TYPE)      /* for z/VSE
   ```

   for a list of versions for the member of the specified object type, or:

   ```
   LIST NV/PV/LV/LMV/VV/MV LIBRARY(MEMBER)
   LIST VV LIBRARY.SUBLIBRARY(MEMBER.MEMBER-TYPE)      /* for z/VSE
   ```

   for a list of versions for the member of the specified object type, or:

   ```
   LIST MV
   ```

   for a list of all members of all object types for which there are previous versions, or:

   ```
   BROWSE NV/PV/LV/LMV/VV  LIBRARY(MEMBER)
   BROWSE VV LIBRARY.SUBLIBRARY(MEMBER.MEMBER-TYPE)      /* for z/VSE
   ```

   to display a version of the member. A window opens with the object parameters of the last version. You can modify any parameter, for example the date/time parameter to see an earlier version.

Note: Versioning of LMS elements is described in detail in the subsection *LMS Library Elements* in section *BS2000 Objects*. Versioning of CA Librarian members is described in the subsection *Previous Versions* in section *CA Librarian Members*.

Example 1: Listing previous versions of a PDS member

The following screen lists previous versions of a PDS member obtained using the command:

```
LIST PV FHI.JCL(COPYNSPF)
```

```
LIST-PV:FHI.JCL(COPYNSPF) ------------------------ Row 0 of 2 - Columns 008 076
 COMMAND===>                                                   SCROLL===> CSR
    VV.MM          MODIFIED TIME     SIZE   ID       CHANGE REASON
** ******************************** top of list ******************************
    01.07          19980603 10:07        27 FHI
    01.06          19980603 09:55        27 FHI
** ****************************** bottom of list ******************************
```

The lists for previous versions of z/VSE members has the same format (for an example, see below).

Meaning of column headings:

Column	Meaning
VV.MM	Version and modification level. If a version is in HOLD status, this field is highlighted.
MODIFIED	Date of modification
TIME	Time of modification.
SIZE	Number of lines.
ID	User ID of user who saved this version.
CHANGE REASON	Brief indication for reason of the modification from one version to the next (see the subsection *Reason for Change*).

You can select any version with the B (BROWSE), CH (CHANGE), D (DELETE), DI (DIFFERENCE), or HL (HOLD) line command. These functions are described in a separate subsection below.

Example 2: Listing previous versions of a Natural member

The following screen illustrates a list of previous versions of a Natural member obtained using the command:

```
LIST NV MYLIB(MYPROG)
```

```
LIST-NV:MYLIB(MYPROG) ---------------------------- Row 0 of 3 - Columns 011 076
 COMMAND===>                                                  SCROLL===> CSR
   NUM VV.MM          MODIFIED TIME     ID       CHANGE REASON
** ******************************** top of list ******************************
     0 01.02          19980722 15:07:30 FHI
    -1 01.01          19980722 15:07:05 FHI
    -2 01.00          19941124 18:58:59 MZC
** ****************************** bottom of list ******************************
```

Meaning of column headings:

Column	Meaning
NUM	Version number: zero (0) indicates the most recent version.
VV.MM	Version and modification level. If a version is in HOLD status, this field is highlighted.
MODIFIED	Date of modification.
TIME	Time of modification.
ID	User ID of user who saved this version.
CHANGE REASON	Brief indication for reason of the modification from one version to the next (see the subsection *Reason for Change*).

You can select any version with the B (BROWSE), CH (CHANGE), D (DELETE), DI (DIFFERENCE), or HL (HOLD) line command. These functions are described in a separate subsection below.

Example 3: Listing previous versions of a z/VSE member

The following screen illustrates a list of previous versions of a z/VSE member obtained using the command:

```
LIST VV NATPROC.IV131.SYSTEM.LIBRARY.TEST1(UEDTB1)
```

Where TEST1 is the name of the sublibrary that contains versions of member UEDTB1.

```
LIST-VV:NATPROC.IV131.SYSTEM.LIBRARY.TEST1(UEDTB1. Row 0 of 4 - Columns 008 076
 COMMAND===>                                                   SCROLL===> CSR
    VV.MM          MODIFIED TIME     SIZE   ID       CHANGE REASON
** ********************************* top of list ******************************
    01.03          19980827 13:49:41     32 MBE      release version
    01.02          19980827 13:47:10     32 MBE
    01.01          19980827 13:45:41     32 MBE
    01.00          19980827 13:41:02     32
** ******************************** bottom of list ******************************
```

Meaning of column headings:

Column	Meaning
VV.MM	Version and modification level. If a version is in Hold status, this field is highlighted.
MODIFIED	Date of modification.
TIME	Time of modification.
SIZE	Number of lines.
ID	User ID of user who saved this version.
CHANGE REASON	Brief indication for reason of the modification from one version to the next (see the subsection *Reason for Change*).

You can select any version with the B (BROWSE), CH (CHANGE), D (DELETE), DI (DIFFERENCE), or HL (HOLD) line command. These functions are described in a separate subsection below.

Available Functions for Previous Versions

Relevant for object types NV, PV, VV and MV

The functions available for previous versions are BROWSE, CHANGE, DELETE, DIFFERENCE and HOLD. All commands can be executed as line commands from lists of previous versions or as function commands with appropriate parameters from any Natural screen.

The following table provides an overview of these functions. More detailed information and examples follow.

Function	Line Command	Meaning
BROWSE	B	Display specified version (no modification possible).
CHANGE	CH	Add/modify change reason of a specified version.
DELETE	D	Delete specified version.
DIFFERENCE	DI	Display difference between current version and specified version.
HOLD	HL	Hold current version, thus preventing it from being automatically deleted.

BROWSE

You can select a previous version from a list of versions with the B line command, or you can issue the BROWSE function command from any system screen.

Example:

For example, the command:

```
BROWSE PV MY.OWN.LIB(MEM)
```

requests a display of a previous version of PDS member MEM in the library MY.OWN.LIB. The command opens a prompt window in which you can specify which version you require by date and/or time:

```
+-----------------------------------------------+
! BROWSE-PDSVER:                                 !
! Dataset Name  ===> MY.OWN.LIB                  !
! Member        ===> MEM                         !
! Date          ===> 93/08/02-11:30              !
! Volume        ===>                             !
! Password      ===>                             !
! Node          ===> 148                         !
+-----------------------------------------------+
```

The current data and time is the default, meaning the current (or latest) version is selected. If you do specify the exact date or time, the version closest to the specification is selected:

```
BROWSE-PV:MY.OWN.LIB(MEM)-V01.03-93/07/29-11:14 --------------- Columns 001 072
 COMMAND===>                                                     SCROLL===> CSR
****** ****************************** top of data ******************************
000001 This is the third version of a PDS member.
000002 This is the third version of a PDS member.
000003 This is the third version of a PDS member.
****** **************************** bottom of data*****************************
```

Note that the header line gives an indication as to the version number and modification level, as well as the date and time the version was created.

CHANGE

The `CHANGE` command is available for any previous version of a member. It is available as the line command `CH` only. The `CHANGE` command allows you to add/modify the `CHANGE REASON` of a version. A window opens in which you can modify/enter the change reason:

```
LIST-NV:MYLIB(MYPROG) ---------------------------- Row 0 of 3 - Columns 011 076
 COMMAND===>                                                 SCROLL===> CSR
   NUM +-----------------------------------------------------------------------+
** *** !                                                                       !
     0 !  Enter reason for change: ___________________________________________  !
ch  -1 !                                                                       !
   -2 +-----------------------------------------------------------------------+
** ****************************** bottom of list ******************************
```

DELETE

You can delete any version of an object by selecting it with the `D` line command from a list of previous versions, or by issuing the `DELETE` function command with the appropriate object type and object parameters.

Example:

For example, the command:

```
DELETE PV MY.OWN.LIB(MEM)
```

Opens a prompt window similar to the one described for `BROWSE` above, and you can select the version to be deleted by specifying the version date and/or time.

If you delete the current version of an object, the previous version becomes the current version. When the object is listed (for example, `LIST PDS`), the modification date and time are refreshed to the date and time of the previous version. In lists of Natural members (`LIST NAT`), however, the `DATE` and `TIME` fields of the "new" current version reflect the date and time of the delete operation.

A Note on Storage:

Storage occupied by previous versions is only freed after a `DELETE` operation if the oldest or most recent versions are deleted.

DIFFERENCE

The `DIFFERENCE` function is available for any previous version of a versioned object. It displays the current version together with any changes made during the period between the selected

version and current version. Changes are indicated by highlighting and a corresponding remark in the prefix area.

Example:

The following figure illustrates the result of the `DIFFERENCE` command on a previous version of Natural member `MYPROG` in library `MBE`:

```
DIFFERENCE-NV:MBE(MYPROG)-Ver<-2>-93/08/02-16:31:08 ----------- Columns 011 076
 COMMAND===>                                                    SCROLL===> CSR
** ******************************** top of list ******************************
      000010 * NSPF - GET MODEL VARIABLES ( CALLNAT TO ISP--MVN )
    Old>0010 * NSPF - GET MODEL VARUABLES ( CALLNAT TO ISP--RVN )
      000020 DEFINE DATA
      000030 GLOBAL USING ISP---GL
    Old>0030 GLOBAL USING ISP----G
      000040 LOCAL
      000050 1 #REQUEST(A8)
      000060 1 #RETURN(A8)
    New>0070 1 #NEWFIELD(A8)
      000080 END-DEFINE
      000090 MOVE ##NAT-REQUEST TO #REQUEST
     000100 CALLNAT 'ISP--MVN' #REQUEST #RETURN ##ISPF-ERC ##ISPF-ERT ##SES-DA
   Old>0100 CALLNAT 'ISP--RVN' #REQUEST #RETURN ##ISPF-ERC ##ISPF-ERT ##SES-DA
      000110   ##ED-SESNUM(O)
      000120 MOVE #RETURN TO ##TEMP-A20
    Del>---- END-SUBROUTINE
      000130 END
** ****************************** bottom of list******************************

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right :s
```

- The lines with no remark in the prefix area are those of the current version.
- The highlighted lines with `Old` in the prefix area indicate the lines which have been changed. The modified line in the current version is directly above the `Old` line.
- The highlighted lines with `Del` in the prefix area indicate the lines that have been deleted and no longer exist in the current version.
- The lines with the highlighted prefix and the remark `New` did not exist in the earlier version and have been added in the current version.

HOLD

The `HOLD` command is available for the current version of an object. Placing a member in `HOLD` status means that it is not counted as an existing version and is not automatically deleted as further versions of the member are created.

Example: Set version number to n

The HOLD command accepts a function parameter VV=*n* in order to set the version number to n instead of incrementing it by 1.

```
HOLD N  MBE(MYPROG), VV=7
```

sets the version number to 7 if the current version is less than seven.

As a result of the HOLD command, the version number (VV value) is increased by 1 or set to the value passed as VV parameter, and the modification level (MM value) is set to 00. Further HOLD operations on the member further increase the version number.

Example: Issue HOLD for current version (with prompting)

The HOLD command can be issued for the current version as line command from a list of versions, or from any system screen as function command. For example, the command

```
HOLD NV MBE(MYPROG)
```

opens the following prompt window:

```
+-----------------------------------------------+
! HOLD-NATVER:                                   !
! Library       ===> MBE                        !
! Member        ===> MYPROG                     !
! Date          ===> 93/08/03-11:53             !
+-----------------------------------------------+
```

Only the current version is selected. Press ENTER to confirm.

Example: Issue HOLD for current version (without prompting)

The command

```
HOLD N MBE(MYPROG)
```

holds the current version of Natural member MYPROG without prompting.

Reason for Change

When updating a member with versioning active, you can indicate a reason for changing the member in either of the following ways. In either case, the reason given appears in the field headed `CHANGE REASON` in the list of versions for the member (see the examples of lists of versions above).

1. When saving the new version of the member, add the reason as parameter to the `SAVE` or `STOW` command, for example:

   ```
   STOW 'Referenced program names changed'
   ```

2. If the library is in the list of versioned libraries, ask your system administrator to add the string `REASON` to the library name. Every time you then issue the `SAVE` or `STOW` command, a window opens prompting you to add the reason for the change.

If you are authorized to change the `CHANGE REASON`, you can also use the `CH` line command.

Editing and Saving Previous Versions

Earlier versions of a member are available in `BROWSE` mode only (no modifications possible, see the description of the `BROWSE` command above). When displaying an earlier version of a member in Editor format, you can use the Editor `CREATE` command to save the version under a different name in the object library.

You must mark the first and last lines of the block in the member you wish to save with the Editor line command `CC` and enter the command:

```
CREATE newname
```

in the command line. Press ENTER to create a new member with the specified name. You can edit and maintain the new member as any other member.

Alternatively, you can use multi-session mode to copy all or part of an earlier version into a new or existing member using the Editor `CC` line command to mark the block to be copied, and the Editor `A`, `B` or `O` line commands in the target member to mark the place where the text is to be entered.

Another alternative is to delete all versions created after the version you want to reactivate. This automatically makes the required version the current version.

Note: If you update a versioned Natural, PDS or z/VSE member with an editor outside of Natural, all versioning data for the member is lost.

Recovery

Natural provides a comfortable recovery facility for lost files after an abnormal termination or system crash. You must have `RECOVERY ON` as a default either in your user profile or specified for your current edit session.

In your user profile, you can also specify the number of modified lines required before an automatic backup is performed (see the section ***Profile Maintenance***).

If you lose files for any reason, Natural notifies you with a message at your next logon, asking you to list the recovery files. This automatic check for recovery files can be switched off in the User Defaults section of the Profile. This significantly reduces the time required for Natural initialization (see the `NO RECOVER` option in section ***Profile Maintenance***). If you issue the `RECOVERY` command (an implicit `LIST REC`), you are presented with a list of recovery files. You can enter either of the following line commands in the input field preceding any file name:

Command	Meaning
`D`	Delete the file.
`E`	Re-edit the recovered file. The file appears as recovered at the last checkpoint.

If more than one file is to be recovered, you can re-edit one file. After saving it, issue the `END` command (usually assigned to PF3) to return to the list of recovered files and you can re-edit the next file.

See also the subsection ***Recovery Files*** in section *Common Objects*.

Copying Objects

Natural provides a `COPY` command which allows you to copy a selected object in a number of ways:

1. Copy and store an object under a different name in the same object library, for example:

   ```
   COPY N NATLIB(PROG1),NATLIB(PROG2)
   ```

   copies Natural member `PROG1` from library `NATLIB` to Natural member `PROG2` in the same library.
2. Copy and store an object under the same or a different name in another library of the same object type, for example:

```
COPY N NATLIB(PROG1),NEWLIB(PROG2)
```

copies Natural member PROG1 from library NATLIB to Natural member PROG2 in the library NEWLIB.

3. Copy and store an object on another computer (if you have a multi-CPU environment), for example:

```
COPY N NATLIB(PROG1),P PDSLIB(PROG2) NODE=155
```

copies the Natural member PROG1 from the library NATLIB as the member PROG2 to the PDS library PDSLIB on Node 155.

4. Copy and store multiple members to another library, for example

```
COPY P PDSLIB(ISP*),NEWLIB
```

copies all PDS members in the library PDSLIB that start with ISP to the PDS library NEWLIB. The same functionality is also available for other object types (Natural objects, z/VSE members, LMS elements, etc.).

Note: In Examples 1, 2 and 3, the copy operation is not performed if the member PROG2 already exists in the target library. In Example 4, for each member name starting with ISP and already occurring both in source and target libraries, the user will be prompted to decide whether or not the member is to be replaced.

5. Copy and store an object as another object type; if the target object already exists, it is overwritten, for example:

```
COPY P PDSLIB(MYMEM),N NATLIB(PROG1),REP
```

copies the PDS member MYMEM from the library PDSLIB to the Natural library NATLIB as the member PROG1. The same functionality is also available for other object types, for example copying a Natural member to a z/VSE member, or copying a BS2000 LMS element to a Natural object.

When using function command syntax as in the above examples, you are prompted to select valid options in selection windows should you omit any required parameter, making the COPY command easy to use even without knowledge of the complete command syntax.

For example, the command

```
COPY PDS,NAT
```

prompts you first for the PDS member to be copied, then for the Natural object as destination.

If the new object name already exists in the target library, you can specify whether the existing member is to be retained or deleted (overwritten) by entering `Y` (overwrite) or `N` (retain) in the target specification window.

The `COPY` command can also be issued from the source object's Entry Panel, with object parameters entered in the input fields. Prompt windows then prompt you to specify the target object type and identifiers.

Under z/OS, if the target dataset of a `COPY D <dataset name>` command does not exist, you are prompted for a file allocation.

Note: Not every object can be stored as any other object type. For example, a Natural program cannot be copied as a job SYSOUT file. Most object types can be source objects, but use as target is restricted to those objects that can be edited, except for module CSECTs, recovery files, menus, Natural error messages, BS2000 job variables and Predict descriptions. The selection windows displayed while executing the `COPY` command list valid values only.

Printing Objects

Objects that can be edited and saved can also be printed in a number of ways.

1. Issue the `PR` line command for an object from a list of object names;
2. Issue the `PRINT` function command from the object's Entry Panel with identifiers in the parameter input fields to print the object;
3. Issue the `PRINT` function command from any system screen in the format

   ```
   PRINT object-type object-parameters,function-parameters
   ```

   where the object parameters depend on the object type (see the description of the required object type).

   Function parameters can consist of the following:

Parameter	Meaning
`ASIS`	Valid from an Editor session only: prints the whole Editor session, including header, PF key line, etc. For more details see the subsection *Printing from an Editor Session*.
`CONTROL / CC`	Honors any ASA or machine code control characters. No additional headers are printed.
`printer-name`	Name of the printer. This printer overrides the printer specified in your user profile.
`NOCONTROL / NO`	Deactivates automatic carriage control when printing Natural objects or job SYSOUT files.

Parameter	Meaning
WORKPOOL	Writes the output to the user workpool.

Or the following keyword parameters can be used:

Parameter	Meaning	
PRINTER	Name of the printer.	
DRIVER	Specify the name of a printer control character table as defined in the Natural NTCC macro, or under Com-plete you can specify the name of a logical output driver routine, which can perform additional output formatting during printing.	
FORM	Specify a printout form specification.	
NAME	Specify a list name for the printout.	
DISP	Specify the disposition of the printout:	
	D or DEL	Delete after printing (default).
	H or HOLD	Hold printout, do not print.
	K or KEEP	Keep after processing.
	L or LEAV	Leave in spool queue after printing.
COPIES	You can specify the number of additional printouts.	
PS	Specify the number of lines per page you want for this printout.	
SUPPRESS	If you enter Y, Natural does not print header information or generate form feeds.	
NOM	NOM=YES indicates that the extended Entire Output Management interface is used for printing. For further information, see the subsection *Natural ISPF Parameters* in section *System Configuration* of the *Natural ISPF Administration Guide*.	

Example:

```
PRINT NAT T1M1, PRINTER=DAEPR1 COPIES=3 PS=20
```

In addition to printing Natural objects you can also print help texts currently displayed on the screen. Just enter the PRINT command from your help screen.

Note: Carriage control when printing Natural objects or SYSOUT files is automatic (see the NO function parameter).

The user workpool can be used as destination for various types of output: macro output, output of Natural programs, output from Natural facilities. For more information, see the subsection ***User Workpool*** in section *Common Objects*.

When printing to the user workpool is selected, the interpretation of carriage control characters is automatically switched off.

Example:

The command:

```
PR N NSPF120(MYPROG),NO
```

prints Natural object MYPROG in library NSPF120 at the default printer without automatic carriage control.

The hardcopy device at which the output is printed is selected according to the following hierarchy:

- The printer specified in the function parameters;
- The printer specified in your user profile.

If an asterisk wildcard (*) is specified as printer name, a window opens after you issue the PRINT command prompting you for the printer name and other printer parameters. The data entered in this window become the defaults for the next PRINT command prompting in your Natural session:

```
-------------------------- NATURAL ISPF MAIN MENU -----------------------------
OPTION  ===> PRINT N NSPF120(MYPROG),*
                                                          Userid   BRY
             +---------Print parameters----------+        Time     12:21:35
   0    PROF !                                  !        Terminal DAELC630
   1    NATU ! Enter printer  : ________        !        Library  BRY
   2    VIEW ! Lines per page : 60              !  Views Node     148
   3    ERRO ! Interpret CC   :                 ! es
   4    PRED ! Suppress Header:                 !
   5    WORK ! Listname       : ________        !
   6    CONT ! Form           : ________        !
             ! Disp           : ____  (D/H/K/L) !
   7    JOBS ! Copies         :         (0-255) ! z/OS)
   8    PDS  ! Log.-Driver    : ________        !
   9    DATA +-----------------------------------+
   10   JOBS       - Display JOBS status and data (VSE)
```

The input fields are described below.

- If no printer is specified in your user profile, the group profile is searched for a printer: for example, if your user ID is ABC, profile AB* is searched, then A*, and then *;
- If no printer is specified in your user group profile, the printer specified in your TP environment is used;
- If no printer is specified in your TP environment, the printer assigned to NATURAL PRINTER 2 is used (see the Natural profile parameter PRINTER);

If none of the above is found, you are notified by an error message.

Meaning of the input fields:

Field	Meaning	
Enter printer	Enter printer name (this can also be Workpool).	
Lines per page	Number of lines to be printed on a page before a page break.	
Interpret CC	Specifies whether ASA or machine control characters, if they appear in the printout, are honored (Y or N).	
Suppress Header	If you enter Y, Natural does not print header information or generate form feeds.	
Listname	Specify a name of the printout.	
Form	Specify a printout form.	
Disp	Specify the disposition of the printout:	
	D or DEL	Delete after printing (default).
	H or HOLD	Hold printout, do not print.
	K or KEEP	Keep after processing.
	L or LEAV	Leave in spool queue after printing.
Copies	Specify number of copies (maximum is 255).	
Log.-Driver	Specify the name of a profile form, that is, the name of a printer control character table as defined in the NTCC macro (see the Natural *Parameter Reference* for more information); or: Specify the name of a logical output driver routine to perform additional output formatting during printing. **Note:** If you want to use the logical output drivers under Software AG's TP monitor Com-plete, ask your administrator to activate the special USPOOL interface with APPLYMOD 22.	

Printing from an Editor Session

1. Issue the PRINT command from an Editor session with the object in EDIT or BROWSE mode to print the member;
2. Use the Editor line command P to print the selected line, or mark the first and the last lines of a block of text with the Editor PP line command to print the block from the member;

Note: When issuing the PRINT command from an Editor session, the current boundary settings (BNDS Editor command) are respected: only the data within the set boundaries are printed.

When issuing a PRINT command from an Editor session, you can use the special parameter ASIS as follows:

```
PRINT,ASIS
```

- If you have a printer defined in your Natural profile, the whole Editor session is printed as it would appear on your screen.
- If you have an asterisk defined as printer in your Natural profile, you are prompted for the printout parameters:

```
BROWSE-NAT:NSPFEXAM(IDB-ADRP)-Program->Struct ---------------- Columns 001 072
 COMMAND===> print,asis                                         SCROLL===> CSR
****** *** +-----------Print parameters------------+ **************************
 000010 * E !                                      ! a window:
 000020 *   ! Lines per page    : 60               ! pplication logic
 000030 *   ! Take linesize from                   ! or messages only
 000040 *   !   edit session    : X                !
 000050 DEF !   or screensize   :                  !
 000060 LOC ! Print prefix      : X                !
 000070 LOC ! Enter printer     : DAEPRT12         !
 000080 1 A ! Listname          : ________         !
 000090   2 ! Form              : ________         !
 000100   2 ! Disp              : L___   (D/H/K/L) !
 000110   2 ! Copies            :          (1-255) !
 000120   2 ! Log.-Driver       : ________         !
 000130   2 ! Print via NOM     :          (Y/N)   !
 000140 1 K +-------------------------------------+
 000150   2 PERSONNEL-ID
 000160   2 NAME
 000170   2 ADDRESS-LINE  (1:2)
 000180   2 COUNTRY
 000190 1 #FUNK  (A1)
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Meaning of the input fields:

Field	Meaning
`Lines per page`	Number of lines to be printed on a page before a page break.
`Take linesize from edit session`	Mark this field if the line length of the printout is to correspond to the length of the edit session (this might be more than 80).
`or screensize`	Mark this field if the line length of the printout is to correspond to the line length of your screen.
`Print prefix`	Mark this field if you want the prefix area (line numbers, labels, etc.) printed.
`Enter printer`	Enter printer name (this can also be `Workpool`).
`Listname`	Specify a name of the printout.
`Form`	Specify a printout form.
`Disp`	Specify the disposition of the printout:

Field	Meaning	
	D or DEL	Delete after printing (default).
	H or HOLD	Hold printout, do not print.
	K or KEEP	Keep after processing.
	L or LEAV	Leave in spool queue after printing.
Copies	Specify number of copies (maximum is 255).	
Log.-Driver	Specify the name of a profile form, that is, the name of a printer control character table as defined in the NTCC macro (see the Natural *Parameter Reference* for more information); or: Specify the name of a logical output driver routine to perform additional output formatting during printing. **Note:** If you want to use the logical output drivers under Software AG's TP monitor Com-plete ask your administrator to activate the special USPOOL interface with APPLYMOD 22.	
Print via NOM	Enter Y to use the extended interface between Natural and Entire Output Management.	

Natural Interface to External Environments

Natural interfaces to the PC environment and to Software AG's Office System Con-nect. You can transfer Natural objects to PC files and Con-nect cabinets, and you can transfer PC files and Con-nect documents to Natural object libraries (PDS, Natural, CA Panvalet, CA Librarian, etc.).

To perform these functions, Natural provides the EXPORT and IMPORT function commands.

EXPORT

You can issue the EXPORT command in any of the following ways:

- As a function command from any Natural screen, with parameters specifying the external environment and identifiers for the object to be transferred; if any parameter is missing, selection windows prompt you for valid items;
- As a command from an edit session with a Natural object, with a parameter specifying the destination; destination Con-nect is specified as CONNECT or CNT, destination PC is specified as PC;
- As a line command (EX) from a list of Natural objects; selection windows prompt you for destination and target identifiers.

IMPORT

The IMPORT command is a local command issued from an edit session with any Natural object that can be edited.

The following subsections illustrate some examples of using the Natural interface to the PC and Con-nect environments.

DOWNLOAD / UPLOAD

In addition to the EXPORT and IMPORT commands, which usually handle text only, the DOWNLOAD and UPLOAD commands also handle binary data. Currently they are available for Natural objects and sources as well as data areas and maps, views and PDS members.

All PC files created with the Natural DOWNLOAD command can be processed by the UPLOAD command as described in the section *Command Reference*.

Exporting a Natural Object to Con-nect

To export a Natural object (PDS member, Natural program, View, sequential dataset, SYSOUT file, workpool entry, library list) to Con-nect, send it to another user or both, enter the following command in the command line and press ENTER:

```
EXPORT object-name object-parameters, TARGET=CNT target-parameters
```

Or, if you are in an Editor session, enter the following command in the command line and press ENTER:

```
EXPORT CNT
```

A window opens.

Example:

The command:

```
EX P MY.ONLY.SOURCE(MYMEM),TARGET=CNT NAME=MYDOC SEND=MYBOSS
```

stores the PDS member MYMEM from the library MY.ONLY.SOURCE as document MYDOC in Con-nect and sends a copy to the user MYBOSS.

Natural notifies you with a message of the successful export operation. If a new document is created, it can be maintained within Con-nect as any other Con-nect document (see the Con-nect documentation).

Exporting Part of an Object to Con-nect

To export only part of a Natural object, open an EDIT session with the object and mark the required block of data with two CC Editor line commands, then issue the command EXPORT CNT. The following window opens:

```
EDIT-NAT:NSPFHELP(LOGOFF)-Text->Report-Free-45K --------------- Columns 001 072
 COMMAND===> export cnt                                          SCROLL===> CSR
****** ****************************** top of data ******************************
000010 &H Session Command: LOGOFF
000020 &U MAIN
000030 +------------------Export document to CON-NECT-------------------+
CC0040 !                                                                !
000050 !  Cabinet:         GHH_____                                     !
000060 !  Password:                                                     !
000070 !                                                                !
000080 !  Create document: ________________________________             !
000090 !   and/or Send to: ____________________ ____________________    !
000100 !                   ____________________ ____________________    !
000110 !  Description/Subject                                           !
000120 !  ______________________________________________________________ !
000130 !  ______________________________________________________________ !
000140 !  ______________________________________________________________ !
000150 !  ______________________________________________________________ !
000160 +-----------------------------------------------------------------+
CC0170 The LOGOFF command can be concatenated with any valid NATURAL command.
000180 Examples:
000190 &L
000200   #LOGOFF;FIN
000210   #LOGOFF IMM;FIN
```

You can store the object as a document in Con-nect, send it as a message to one or more Con-nect users, or both.

Meaning of the input fields:

Field	Keyword	Meaning
Cabinet	CABINET	Required. Name of the Con-nect cabinet in which the object is to be stored as a document.
Password	PASSWORD	Required, if Con-nect password is needed to access the cabinet.
Create document	NAME or DESTINATION	Document name under which the object is to be stored. Required for filing the Natural object as a Con-nect document. (Not required for the SEND function.)
and/or Send to	SEND	To send the Natural object as a message to Inbaskets of other Con-nect users, enter at least one recipient. You can specify up to four user IDs. (Not required when only storing object as document.)
Description/Subject		Optional description of object. Required for sending the Natural object as a message to other Con-nect users.

Note: If you specify document name, up to four user IDs as well as the subject, the Natural object is both filed as a document in the specified cabinet and sent to the Inbasket of the specified users.

Having specified the Con-nect destination, press ENTER. Natural notifies you with a message of the successful export operation. If a new document is created, it can be maintained within Con-nect as any other Con-nect document (see the Con-nect documentation).

See also the description of the `EXPORT` command as described in the section *Command Reference*.

Importing a Con-nect Document

To import a Con-nect document into Natural, you can start an edit session with the Natural Editor for the required object type. When the Editor screen is displayed on your screen, enter the command:

```
IMPORT CNT
```

in the Editor direct command line and press ENTER. This opens a window in which you must specify the Con-nect document to be transferred:

```
EDIT-PDS:MBE.COMN.SOURCE(DOCUMENT)-----------------------------columns-001-072-
 COMMAND===> import cnt                                        SCROLL===> CSR
****** ****************************** top of data ******************************
''''''
''''''
''''''
''''''
''''''      Cabinet    ________
''''''      Password
''''''      Name       ________________________________
            Mark for Notes/Encl. _
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
****** **************************** bottom of data ****************************
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Meaning of the input fields:

Field	Keyword	Meaning
Cabinet	CABINET	Name of the Con-nect cabinet from which the document is to be copied.
Password	PASSWORD	If required, Con-nect password to access the cabinet.
Name	NAME	Document name.
Mark for Notes/Encl.		Mark this field if the document is to be imported with cover notes and enclosures.

Specify the Con-nect document to be transferred in the window and press ENTER.

If you are creating a new Natural object (your Editor screen is blank), the document is loaded into the data area of the Editor screen. If you wish to include the document in an existing object, Natural asks you to position the document with the message Import is pending. You must mark the line after which you wish to insert the document with the A line command, or the line before which you wish to insert the document with the B line command.

Any document transferred to Natural can be edited, stored and maintained as any other Natural object.

Using the Natural Interface to the PC Environment

If Entire Connection is installed at your site, you can use a PC to emulate a mainframe terminal. You can transfer files between the PC and Natural using the IMPORT and EXPORT function commands.

After logging on to Natural from the PC, you must issue the Natural terminal command

```
%+
```

from the Natural Main Menu. The terminal command %+ sets the Natural system variable *DEVICE to PC (see the Entire Connection documentation for more information).

The following subsections describe how to export Natural objects and how to import PC files in more detail.

Exporting a Natural Object to the PC

To export (download) a Natural object (PDS member, Natural program, view, sequential dataset, SYSOUT file, workpool entry, library list) to the PC, enter the following command in the command line and press ENTER:

```
EXPORT object-name object-parameters, TARGET=PC DESTINATION=<ilename.ext
```

If you omit the DESTINATION parameter, a window opens at the bottom of the screen, that prompts you to specify the name under which the object is to be stored on the PC. Enter a name for the new file and press ENTER.

Note: To download only part of a Natural object, open an EDIT session with the object and mark the required block of data with two CC Editor line commands, then issue the command EXPORT PC.

You are informed of the successful download operation with a message. You can check the result of your download operation by returning to MS-DOS and issuing the DIR command or the TYPE <filename> command.

To export Natural objects in binary format, Natural views or z/OS load modules, use the DOWNLOAD command as described in section *Command Reference*.

Examples:

- The command:

  ```
  EX P MY.ONLY.SOURCE(MYMEM),PC
  ```

  opens a window in which you can enter the PC file name under which the PDS member MYMEM in library MY.ONLY.SOURCE is to be downloaded.

- The command:

  ```
  EX P MY.ONLY.SOURCE(MYMEM),TARGET=PC DESTINATION=MYFILE.NCD
  ```

  downloads the PDS member MYMEM from library MY.ONLY.SOURCE to the PC and stores it in MYFILE.NCD without prompting.

See also the description of the EXPORT command in section *Command Reference*.

Importing a PC File into Natural

To import (upload) a PC file into Natural, you must start an edit session with the Natural Editor for the required object type. (Do not forget to signal to Natural that you are working from a PC by entering the percent sign followed by the plus sign: %+).

When the Editor screen is displayed, enter the command:

```
IMPORT  PC, NAME=myfile.ncd
```

in the Editor command line and press ENTER. If the NAME parameter is omitted, a window opens in which you are prompted for the PC file name to be uploaded. Enter the file name in the window and press ENTER.

If you are creating a new Natural object (your Editor screen is blank), the file is loaded into the data area of the Editor screen. If you wish to include the file into an existing object, Natural asks you to position the document with the message Import is pending. You must mark the line after which you wish to insert the document with an A line command, or the line before which you wish to insert the document with a B line command.

Any PC file transferred to Natural can be edited, stored and maintained as any other Natural object.

To import Natural objects in binary format, Natural Views or z/OS load modules, use the UPLOAD command as described in the section *Command Reference*.

Message Switching

Messages can be sent to TSO, TIAM or Com-plete users with the SEND command issued from any screen. This opens the following window:

```
---------------------- NATURAL OBJECTS - ENTRY PANEL -------------------------
COMMAND ===> send


   Library       ===> NSPFEXAM
   Member        ===> *
   +-------------------------------------------------------------------------+
   !                                                                         !
   !  Send to : ________   ________   ________   ________   ________        !
   !  Node    : 148                                                          !
   !  Text    :                                                              !
   !  ______________________________________________________________________ !
   !                                                                         !
   +-------------------------------------------------------------------------+









Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Meaning of the input fields:

Field	Meaning
`Send to`	Specify up to five destination user IDs.
`Node`	Node number of destination.
`Text`	Message text.

Using the Calculator

The `CALC` session command invokes the ISPF calculator:

```
EDIT-PDS:MBE.COMN.SOURCE(ISPIDB) ---------------- +---------------------------+2
 COMMAND===> calc                                 !                           !
000022                                            !             PF-Keys       !
000023 Incompatibility:. The previous version who !             1 Help        !
000024                   prefix an suffix, this v !             2 Command     !
000025                                            !             5 Print       !
000026 Examples -       DATA-FORMAT ******C       !             7 Up          !
000027                  DATA-FORMAT = /*,*/    /* !             8 Down        !
000028                                            !             9 Mode(NHD)   !
000029                                            !               NUMERIC     !
000030 Get Data                                   !             10 Left       !
000031 --------                                   !             11 Right      !
000032 Syntax   -       GET-DATA   ( USING <local !             12 Clear      !
000033                             <var-name> ..  ! =========== ------------ !
000034                  END-GET                   !           0               !
000035                                            !             <--Input---- !
000036 Where    -      <local-name> =  Name of lo !             <  Examples: !
000037                 <var-name>   =  var-name(< !             <  12*33+5   !
000038                 <index-definition> =   (n) !             <  -         !
000039                 <index-definition> =   (n) !             <  126 / 2   !
000040 Explanation     GET-DATA will restore vari !             <  +452      !
000041                 value of SAVE-DATA, from t !             <  *A0 + H10 !
000042                 the fields are taken from  !             <----------- !
000043                 their  name explicity.     +---------------------------+
```

The calculator appears as a window on your screen and consists of two parts:

The bottom half of the calculator window is the input area into which you can enter the problem to be computed. You can enter numbers in the same line or in consecutive lines, as long as the numbers are preceded by the appropriate operator. Valid operators are:

- `+` for addition (default);
- `-` for subtraction;
- `*` for multiplications;
- `/` for divisions.

Numbers can be entered as numerical digits or in hexadecimal format. If the data entered in the input line contains more than one operator, these will be evaluated from left to right, disregarding the arithmetic processing order.

After you have entered a computing problem in the input area and pressed ENTER, the numbers reappear in the top half of the calculator in the order you entered them, preceded by the operator. The solution appears directly under the line separating the top from the bottom half. Note that you can modify values on the calculator to make corrections to the problem. The solution is immediately updated after you press ENTER.

When the calculator is displayed on your screen, you can perform certain functions using the following PF keys:

PF Key	Explanation
PF1	Display help text on the input area and paper area of the calculator.
PF2	Use the calculation result as NSPF command: a window opens with the problem solution in it. You can use the solution to form an NSPF command which is executed when you press ENTER. This is useful for example, if you want to find a hexadecimal result in a dump.
PF3	Close the calculator window, content is kept.
PF5	Print the contents of the calculation (destination can also be the workpool).
PF7	Scroll up list of the numbers in top half.
PF8	Scroll down list of the numbers in top half.
PF9	Toggle display among numerical digits, hexadecimal format, and numerical digits with two places after the decimal.
PF10	Move the calculator window to the left.
PF11	Move the calculator window to the right.
PF12	Clear the calculator for the next computation.

Using the Macro Facility

Natural provides a facility with which source can be generated automatically, relieving you of much routine editing work, and eliminating editing errors. This feature is useful when writing programs with a similar structure but with different content.

The macro feature is an extension of the Natural language and consists of special processing statements executed when the program is compiled, as well as variables in source lines substituted by valid values at compilation time. These special processing lines and variables are distinguished in the source by typing the macro character as first character in the line.

As a simple example of macro processing lines and variables, if you define the following macro:

```
0010 § MOVE 'PERSONNEL' to #FILE-NAME (A32)
0020 § MOVE 'NAME'      to #KEY       (A32)
0030 READ §#FILE-NAME by §#KEY
```

the following source line is generated at compilation time:

```
READ PERSONNEL BY NAME
```

By varying the values, a variety of source lines can be created from the same "skeleton" macro. You can use this macro feature in Natural in a variety of ways:

1. **Macro objects:**
   Macro objects are a special type of Natural object and can be accessed and maintained as any other Natural object (specify `TYPE=MACRO`). They contain macro processing lines and macro variables and can be used to generate Natural programs, or they can be referenced from other Natural objects to generate text lines;
2. **Inline macros:**
   Other Natural objects (e.g. Natural programs, PDS members, z/VSE members, etc.) can use macro variables in their source. They can also reference a macro object using the special `INCLUDE-MACRO <name>` statement. The variables are substituted and the specified macro object is executed, and the generated lines are included in the source when the object is compiled (Natural) or submitted (non-Natural);
3. **Edit macros:**
   When starting an edit session with a Natural object (Natural program, PDS member, z/VSE member, LMS element etc.), you can specify a macro object to be used as a model for the new edit session. The specified macro is executed and the generated lines written to the new edit session. The lines generated in this way are protected, but you can reserve some places in the macro object in which you can add your own code after the macro is invoked as an edit macro.

The generated output of objects that use the macro facility is written to the user workpool, where it can be checked and further handled (for full details and examples, see the section *Macro Facility* in the *Natural ISPF Programmer's Guide*).

Executing Command Scripts

This subsection explains the commands which can be used in, or in conjunction with, command scripts:

- PLAY Function Command
- RECORD Session Command
- PAUSE Session Command
- Scripts in the User Workpool
- CONTINUE Session Command
- Example:
- MACPARM Session Command
- MESSAGE Session Command

- REMARK Session Command
- Generating Scripts using a Macro

You will find examples of how to generate scripts using a macro under ***Generating Scripts using a Macro***.

PLAY Function Command

Natural commands can be written and stored in a member. Such a member is called a command script. The script can be executed with the `PLAY` function command. The commands are then executed sequentially. Scripts executed by the `PLAY` command are written to the User Workpool (see the subsection ***Scripts in the User Workpool***).

Command scripts can be stored as any of the following object types:

- Natural object
- PDS member
- Workpool output file (see below)
- z/VSE member
- Macro
- BS2000 LMS element

Example:

Assume Natural member `MYPROCA` in library `MYLIB` contains the following lines:

```
ED MYLIB(PROGA);STOW;END
ED MYLIB(PROGB);STOW;END
ED MYLIB(PROGC);STOW;END
```

The command:

```
PLAY MYLIB(MYPROCA)
```

edits and stows the Natural programs `PROGA`, `PROGB` and `PROGC`.

- A script that has been interrupted by an included `PAUSE` command (see below) can be cancelled by the session command `PLAY OFF`. This command also deletes the workpool entry (see the subsection ***Scripts in the User Workpool***).
- If an invalid command is detected in a command script during execution, and the script does not contain a `CONTINUE` statement, the script is stopped automatically and the invalid command is displayed in the Natural command line. You can correct the faulty command and reexecute it by pressing ENTER. You can then continue the script by issuing the `PAUSE` session command (see below).

- If one of the played commands results in an error message (for example, if program PROGB in the above example does not contain correct Natural source, the same logic applies as for invalid commands.
- A command script held in a PDS, z/VSE or LMS member may not contain line numbers within the member's lines (for example, in columns 73 to 80). Thus, when editing such a command script, it is recommended that you switch off the number mode of the Editor, either with the **profile settings** or by issuing one of the Editor commands AUTOREN OFF or UNREN. You should be aware that the line numbers could have been inherited unintentionally when copying lines from one EDIT session to another with the **line commands** mentioned in this documentation. Line numbers contained in script lines could result in unpredictable error messages during command script execution.
- It is possible to use nested PLAY commands in a command script. The contents of a member to be played is always entered at the top of the ##PLAY workpool session.

Example:

Assume the member EX1 contains the following lines:

```
EDIT NAT MYPROG
PLAY NAT EXPLAY(CHG-AB)
EDIT NAT MYPROG1
PLAY NAT EXPLAY(CHG-AB)
```

and assume member CHG-AB contains the following lines:

```
CHANGE 'A' 'B' all
STOW
END
```

The command:

```
PLAY EX1
```

starts an edit session with Natural member MYPROG, plays the commands in CHG-AB, and only then is the command EDIT MYPROG1 processed.

- In the COMMAND field in your user profile, you can specify a command that is executed every time you start Natural. You can specify a PLAY command in this profile field to start a command script when you log on (see also the description of the COMMAND field in the ***User Defaults*** sub-section of section *Profile Maintenance*).
- It might be useful when writing a command script to start it with RECORD ON and finish it with RECORD OFF. This causes all messages to be recorded in the User Workpool (see the RECORD session command below).

RECORD Session Command

You can record Natural commands using the RECORD session command. After you have issued the RECORD session command, all ensuing commands are recorded until you issue the RECORD OFF session command. The recorded commands can be found in the User Workpool in member ##RECORD. This member can be PLAYed.

If a command issued during the recording session causes a message, the message is also recorded in the User Workpool member ##RECORD, preceded by two asterisks (**). The message is ignored when the member is executed with the PLAY command.

PAUSE Session Command

In some cases it is useful to interrupt a script being executed, for example in order to manipulate some data manually, and then continue executing the script. This can be done by writing the session command PAUSE in the script. When the script is executing, it stops at the place the PAUSE command was entered. To continue the script, simply issue the PAUSE command manually.

Example:

Assume the member MYPROCB contains the following lines:

```
EDIT MYJOB
PAUSE
SUB
FOLLOW
CAN
```

The command:

```
PLAY MYPROCB
```

starts an edit session with the member MYJOB and then stops in order to allow modification of the JCL. If you then issue the PAUSE command from the Natural command line, the JCL is submitted, job status messages are displayed (FOLLOW command) and the edit session is cancelled.

Note: The PAUSE command must always be the last command or the only command in a script line.

Scripts in the User Workpool

A script which is executed by the PLAY command is stored in the User Workpool in an entry named ##PLAY. When a script is interrupted by a PAUSE command or an error, the lines not yet executed are in the workpool member ##PLAY and can be modified.

CONTINUE Session Command

The CONTINUE command can be used in command scripts to gain more flexible control in error situations. If no CONTINUE statement is in the command script, the script is set to PAUSE mode after an error.

If a CONTINUE statement (which can be compared to a label) is in the script, the following actions are taken:

1. RECORD ON is set internally if not activated by the user.
2. The command causing the error and the message is recorded.
3. All lines of the script until the next CONTINUE command are deleted and execution of these lines is skipped.
4. Processing continues with the next CONTINUE statement. All following statements are executed.
5. Termination resets RECORD to its previous value and informs the user if an error has occurred.

Example:

```
KEYS 3 PAUSE
HELP VERIFY
MESSAGE 7480
TECH
........
CONTINUE
REMARK PROCESSING WILL CONTINUE HERE AFTER ERROR
KEYS 3 END
```

The above script modifies the user profile. By using the CONTINUE command it makes sure that, after execution of the script, PF3 is reset to its default value from the user profile, even if errors have occurred during execution of the script.

MACPARM Session Command

The `MACPARM` command is used in command scripts to put data on the Natural stack which is read by a macro using an input statement later in the command script.

This avoids prompting by the macro for parameters, when using macros in command scripts. The `MACPARM` command must be the only command in a source line.

The command format is:

```
MACPARM p1
```

Explanation of parameters:

Parameter	Meaning
p1	Maximum length of this parameter is 50 bytes and it can contain blanks.

Examples:

The commands:

```
MACPARM LS PDS JW(A*)
PLAY MAC MAC1
```

pass the command `LS PDS JW(*)` to macro `MAC1`.

Another useful example can be found in the member `VERIFY` in the **Natural Example Library**.

MESSAGE Session Command

The `MESSAGE` command can be used in command scripts to display a text during execution of a script on the screen and to interrupt the active command script. The `MESSAGE` command must be the only command in a source line.

The command format is:

```
MESSAGE p0,p1,...pn
```

Explanation of parameters:

Parameter	Meaning
p0	Must be a 4-digit error message number. First, the user library SYSISPFU is searched for the message text. If it does not exist, it is taken from the system library SYSISPS1.
p1,...*pn*	Optional parameters which are used to replace variable parameters (:1: . . . . . :*n*:) in the text. Parameters must usually be separated with your parameter delimiter, usually a comma (,) and can contain blanks.

Examples:

The command:

```
MESSAGE 6812,MYPROG
```

results in the following message, if no text for this number is available in the user library SYSISPFU:

```
Member MYPROG not found
```

The command:

```
MESSAGE 6809,Please enter some text
```

results in the following message, if no text for this number is available in the user library SYSISPFU:

```
Please enter some text
```

Another useful example can be found in member VERIFY in the **Natural Example Library**.

REMARK Session Command

The REMARK command is used in command scripts to document the command script.The REMARK command must be the only command in a source line.

The command format is:

```
REMARK text
```

Example:

```
REMARK The following command extracts all members
REMARK including the string Adabas
LIST PDS JW(*) SC=Adabas
```

Generating Scripts using a Macro

A command script can be generated by a macro. With this mechanism, scripts can be created dynamically.

Example: Generate prompt for CHANGE

For example, executing the following macro with the `PLAY` command generates a prompt for a `CHANGE` command to be used on a member, with a choice of a `STOW` or `SAVE` command after the change is made:

```
§ RESET #MEMBER(A8) #FROM(A16) #TO(A16) #STOW(A1)
§ INPUT(AD=MI) 'Change' #FROM 'To' #TO 'in member' #MEMBER
§            / 'Stow?' #STOW
EDIT NAT §#MEMBER
CHANGE '§#FROM'  '§#TO' all
§ IF #STOW NE ' ' DO
STOW
§ DOEND
§ ELSE DO
SAVE
§ DOEND
END
```

Example: Installation Verification

The following macro generates a command script which verifies Natural installation. It also provides you with a working example of how the commands `CONTINUE`, `MACPARM`, `MESSAGE` and `REMARK` function:

```
§ DEFINE DATA LOCAL
§ 1 #SUB-SYSTEM (A3)
§ 1 #SUB-SYSTEM-INDEX(N1)
§ 1 #SUBSYS-ARRAY (4)
§   2 #SUBSYS-LONG (A10) INIT <'NATURAL', 'z/OS', 'VSE', 'BS2000'>
§   2 REDEFINE #SUBSYS-LONG
§     3 #SUBSYS-SHORT (A3)
§ END-DEFINE
§ IF *DATA GT 0
§   INPUT   #SUB-SYSTEM
§   EXAMINE #SUB-SYSTEM TRANSLATE INTO UPPER
§   EXAMINE #SUBSYS-SHORT(*) FOR #SUB-SYSTEM INDEX #SUB-SYSTEM-INDEX
```

```
§ END-IF
§ IF #SUB-SYSTEM-INDEX EQ 0
§   #SUB-SYSTEM-INDEX := 1    /* NATURAL is default
§ END-IF
§ #SUB-SYSTEM := #SUBSYS-SHORT(#SUB-SYSTEM-INDEX)
KEYS 3 PAUSE
HELP VERIFY
MESSAGE 7480,§#SUBSYS-LONG(#SUB-SYSTEM-INDEX)
TECH
END
§ IF #SUB-SYSTEM-INDEX EQ 1
PLAY NAT VERIFYO
§ ELSE
MACPARM §#SUB-SYSTEM-INDEX
PLAY MAC VERIFYS
§ END-IF
CONTINUE
REMARK Processing will continue here after error
REMARK and pf3 will be reset to its initial profile value
KEYS 3 INITIAL
RECORD OFF
```

Additional Features in Lists

The following topics are covered below:

- ALL Command for Lists
- LAYOUT Command for Lists
- RELIST Command for Lists
- SORT Command for Lists

ALL Command for Lists

In most of the Natural `LIST` sessions, the command `ALL` is available to execute a command or command sequence for all objects shown in the list. This is done internally by generating a command script and executing (playing) it.

Examples:

Command	Function
`ALL ED`	The command `EDIT` is executed for all members of the list. If an `END` command usually assigned to PF3 is entered, the current `EDIT` session is terminated and an `EDIT` session for the next member from the list is opened.
`ALL PG *`	All objects from the list (probably jobs) are purged.
`ALL 'ED *;STOW;END'`	All members are edited and compiled.
`ALL ED +`	A window opens for entering the command string. For further information see the example below.

The command string must be included in quotes, if it contains command or parameter delimiters like semicolon (;) or comma (,). The following special characters can be used inside the command string:

Character	Meaning
`+`	A plus sign entered as last character indicates that command string is not yet complete. A window opens where further data can be entered.
`*`	An asterisk indicates position where the full object name is inserted into the command.
`:1:`	A number in colons indicates position where the first part of the object name is inserted into the command.
`:2:`	As above but second part.
`:3:`	As above but third part.

Notes:

1. The last three notations are not available with most of the Natural objects because the full object name which identifies an object consists of one word only. However, there are exceptions like z/VSE Librarian members which are identified with name and type. Also LMS member names consist of several components.
2. For the commands to operate on selected objects listed, use the Editor command `EXCLUDE` or the Editor line commands `X` or `XX` to reduce the set of displayed lines. A subsequent `ALL` command operates only on the displayed lines, not on the excluded ones.

Example: ALL Prompt Window

The prompt window opens, if it is requested by a plus sign as last character of the command string or when Natural detects that more than a simple command has been entered. Simple commands consist of only one Natural function code or function code and wildcard character (*):

```
ALL ED
ALL PG *
```

Simple commands are executed immediately, and if the wildcard character (*) is not entered, it is automatically generated at the end of the command string, so `ALL ED` and `ALL ED *` are treated identically.

The following is an example of the prompt window that opens after entering the command:

```
ALL ED +
```

```
LIST-NAT:NSPFHELP(*)/SCAN=&V211 ---------------- Row 0 of 113 - Columns 010 076
 COMMAND===> ALL ED +                                          SCROLL===> CSR
   MEMBER            PGMTYPE     SM S/C  NUM FIRST FOUND
** ******************************** top of list ******************************
+------------------------List command propagator-------------------------+
! ENTER without any changes will start processing.                       !
! Command       :  EDIT *_____________________________________________    !
!                  ______________________________                        !
! Generate only :  _                                                     !
!                                                                        !
+------------------------------------------------------------------------+
   BD-FCBTY          Text           S      1 &V211 I
```

You can abort processing by pressing PF3. If you mark `Generate only`, then the command script is generated only to the Natural workpool. Execution is not started. The generated script can be executed later with the `PAUSE` command or can be saved in a member for execution with the `PLAY` command.

For example, if you modify the `Command` field of the above window to contain:

```
EDIT *;PLAY NAT SCRIPTS(REPLA-01);SAVE;END
```

and press ENTER twice, all members listed will be edited, modified according to the Editor commands contained in the Natural member `REPLA-01` of the library `SCRIPTS` and updated without any further manual activity.

LAYOUT Command for Lists

This feature applies to any list of objects displayed in Natural.

When a list of Natural objects is displayed, you can change the layout of the list for better viewing according to your own criteria. This is done by issuing the local command `LAYOUT` from the list. This opens the following window:

```
LIST-NAT: +----------List Layout definition-----------+ f 531 - Columns 010 076
 COMMAND= !  Select fields and sort sequence:         !         SCROLL===> PAGE
   MEMBER !                                           !    DATE     TIME  VV.MM
** ****** !     Field-name                     AD Seq ! **********************
   ISP--- !  1_ MEMBER                          _  __ !    20000406 14:23
   ISP--- !  __ PGMTYPE                         _  __ !    20030109 15:35
   ISP--R !  __ SM                               _  __ !    20070326 15:17
   ISP--R !  __ S/C                              _  __ !    20020904 16:57
   ISP--R !  __ VERSION                          _  __ !    20070615 14:31
   ISP--S !  __ USERID                           _  __ !    20070615 14:39
   ISP--Z !  __ DATE     TIME                    _  __ !    20020904 16:56
   ISP--Z !  __ VV.MM                            _  __ !    20030324 11:28
   ISP--Z !                                            !    20020904 16:56
   ISP--Z !                                            !    20020904 16:56
   ISP--Z !                                            !    20020904 16:56
   ISP-AC !  Enter (Y) to delete layout    _           !    20020904 16:57
   ISP-AC !  Entr-PF3--PF7--PF8--                      !    20020904 16:56
   ISP-AC !  Down End  Up   Down                       !    20020904 16:57
   ISP-AC +-----------------------------------------+    20020904 16:56
   ISP-AIVP          Program     R  S/C 3.1 0006  JWO      20020904 16:57
   ISP-AS-C          Copycode    S  S   3.1 0002  ISP242   20000406 14:23
   ISP-AS1C          Copycode    S  S   3.1 0002  ISP242   20000406 14:23
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Meaning of the column headings in the window:

Column	Meaning
`Field name`	These are the column headings for the list. You can change their order from left to right by assigning sequence numbers to them. For example, enter `1` before `MEMBER` to display it as the first column on the left, enter `2` before `PGMTYPE` to display it as the second column, etc. If you do not assign a sequence number to a column heading, the heading does not appear in the display. Note that some sequence numbers are not modifiable due to internal processing reasons.
`AD Seq`	These two fields allow you to define the sorting hierarchy. In the `AD` field, enter `A` for ascending order or `D` for descending order. In the `Seq` field, enter a sequence number for the sorting hierarchy. For example, enter `1` in the `Seq` field of the `USERID` line and `2` in the `DATE/TIME` line to sort the list first according to user ID and then for each user according to date and time. You can specify up to 5 fields to define the sorting hierarchy.

- The defined layout can optionally be stored in your user profile. Once you have defined the desired layout and attempt to close the window with PF3, another prompt window asks you whether the layout is to be stored or not. `Y` stores the layout in your user profile, `N` keeps the layout for the current session only.
- You can delete a defined layout by issuing the `LAYOUT` command from the respective list and entering `Y` in the field prompted ENTER (`Y`) to delete layout. The list is then displayed in the standard format when regenerated.

Example:

Assuming you have generated a list of Natural objects using the command:

```
LIST NAT NSPF82(ISP-*)
```

and from the list you have issued the `LAYOUT` command. You are interested in listing members by type, user ID and date. You could fill the prompt screen as follows:

```
LIST-NAT: +----------List Layout definition-----------+ f 531 - Columns 010 076
 COMMAND= !  Select fields and sort sequence:         !         SCROLL===> PAGE
   MEMBER !                                           !    DATE     TIME  VV.MM
** ****** !     Field-name                     AD Seq ! **********************
   ISP--- !  1_ MEMBER                          A  4_ !    20000406 14:23
   ISP--- !  2_ PGMTYPE                         A  1_ !    20030109 15:35
   ISP--R !  __ SM                              _  __ !    20070326 15:17
   ISP--R !  __ S/C                             _  __ !    20020904 16:57
   ISP--R !  __ VERSION                         _  __ !    20070615 14:31
   ISP--S !  3_ USERID                          A  2_ !    20070615 14:39
   ISP--Z !  4_ DATE     TIME                   D  3_ !    20020904 16:56
   ISP--Z !  __ VV.MM                           _  __ !    20030324 11:28
   ISP--Z !                                           !    20020904 16:56
   ISP--Z !                                           !    20020904 16:56
   ISP--Z !                                           !    20020904 16:56
   ISP-AC !  Enter (Y) to delete layout    _          !    20020904 16:57
   ISP-AC !  Entr-PF3--PF7--PF8--                     !    20020904 16:56
   ISP-AC !  Down End  Up   Down                      !    20020904 16:57
   ISP-AC +-------------------------------------------+    20020904 16:56
   ISP-AIVP          Program     R  S/C 3.1 0006  JWO      20020904 16:57
   ISP-AS-C          Copycode    S  S   3.1 0002  ISP242   20000406 14:23
   ISP-AS1C          Copycode    S  S   3.1 0002  ISP242   20000406 14:23
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

If you close the window and repeat the `LIST` command, the list is displayed in the new format: the columns from left to right are `MEMBER (1)`, `PGMTYPE (2)`, `USERID (3)` and `DATE/TIME (4)`. The other columns are not displayed. The data is sorted first by `PGMTYPE (1)`, then `USERID (2)`, `DATE/TIME (3)` and `MEMBER (4)`. `DATE/TIME` is in descending order, the other columns in ascending order.

An example of the resulting list is illustrated on the next page.

This list shows the result of the procedure described above, with a `FIND MAP` command issued to display `MAP` type objects:

```
LIST-NAT:NSPF82(ISP-*)M ------------------------- Row 0 of 66 - Columns 010 076
 COMMAND===>                                                   SCROLL===> PAGE
   MEMBER            PGMTYPE     USERID   DATE     TIME
** ******************************** top of list ******************************
   ISP-POP1          Map         BLI      20110119 14:50
   ISP-TE11          Map         BLI      20110117 15:15
   ISP-TEC1          Map         BLI      20110117 15:00
   ISP-SPA1          Map         BLI      20081029 11:44
   ISP-SPR1          Map         BLI      20081029 11:43
   ISP-RLO1          Map         FHI      20060621 10:46
   ISP-BC5M          Map         FHI      20031113 08:58
   ISP-MEC1          Map         FHI      20031113 08:58
   ISP-BC-M          Map         FHI      20031113 08:57
   ISP-ACM1          Map         JWO      20020904 16:56
   ISP-ACT1          Map         JWO      20020904 16:56
   ISP-BPS1          Map         JWO      20020904 16:56
   ISP-BR-M          Map         JWO      20020904 16:56
   ISP-BRK1          Map         JWO      20020904 16:56
   ISP-BRR1          Map         JWO      20020904 16:56
   ISP-BRV1          Map         JWO      20020904 16:56
   ISP-BR5M          Map         JWO      20020904 16:56
   ISP-CAB1          Map         JWO      20020904 16:56
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

RELIST Command for Lists

In most of the Natural `LIST` sessions, the command `RELIST` is available to rebuild the list to reflect changes from line commands like `DELETE`, `RENAME` or `EDIT`.

This command does not apply to so-called refreshable lists like list of jobs, which are rebuilt

Example:

This example shows a list of PDS members after execution of the `DELETE` line command on a number of members:

```
LIST-PDS:BRY.COMN.SOURCE(H*) ------------------------ >>> Member H11000 deleted
 COMMAND===>                                                SCROLL===> CSR
   MEMBER            VV.MM  CREATED  MODIFIED TIME  SIZE   INIT   TID   ID
** ******************************** top of list ******************************
   HELP              01.04  19970202 19970202 15:19     71     51       BRY
   HELP1             01.99  19970203 19970208 17:27    153     63       BRY
   HPSPOOL           01.03  19980112 19980112 16:55     44     10       GW
   H11000   *Deleted 01.01  19970318 19970318 14:42     20     20       BRY
   H11002   *Deleted 01.01  19970318 19970318 14:43     29     29       BRY
   H1100210 *Deleted 01.01  19970318 19970318 14:44     19     19       BRY
   H11004   *Deleted 01.01  19970318 19970318 14:44     12     12       BRY
   H11006   *Deleted 01.01  19970318 19970318 14:45     15     15       BRY
   H11007   *Deleted 01.01  19970318 19970318 14:45     21     21       BRY
```

Below you can see the result of the `RELIST` command:

```
LIST-PDS:BRY.COMN.SOURCE(H*) --------------------- Row 0 of 3 - Columns 010 076
 COMMAND===>                                                SCROLL===> CSR
   MEMBER            VV.MM  CREATED  MODIFIED TIME  SIZE   INIT   TID   ID
** ******************************** top of list ******************************
   HELP              01.04  19970202 19970202 15:19     71     51       BRY
   HELP1             01.99  19970203 19970208 17:27    153     63       BRY
   HPSPOOL           01.03  19980112 19980112 16:55     44     10       GW
** ****************************** bottom of list ******************************
```

SORT Command for Lists

When displaying lists of objects, you can use the Editor `SORT` command with special parameters:

```
SORT column-header a/d
```

where:

Parameter	Meaning
`column header`	String of the column header according to which the items in the list are to be sorted.
`a`	Ascending order (default).
`d`	Descending order.

Examples:

Command	Function
`SORT DATE`	Sorts the list in ascending order according to date, that is, the item edited last is at the bottom of the list. The `SORT` command with no parameters assumes the first column is to be taken.
`SORT :C`	If you use the `SORT` command in lists frequently, you should define a PF key with this command. Then, to perform the sort, just move the cursor to the relevant column header and press the assigned PF key.

5 Common Objects

- Natural Objects 92
- Natural Views 113
- Natural Error Messages Member: ERRMENU 124
- Predict Descriptions 127
- Con-nect Documents 129
- User Workpool 134
- Recovery Files 143
- Container Files 145

Natural Objects

This subsection explains all the functions you can perform on Natural objects. It also describes how to write program output to the user workpool, where it can be handled further.

If the object consists of job control, you can use the Natural ISPF macro facility. You can use all types of macro statements. Macro expansion is performed at submission time (see the `SUBMIT` command below).

When creating a new object, you can also use the Edit macro feature to automatically create text lines which can then be modified. For details, see the section *Macro Facility* in the *Natural ISPF Programmer's Guide*).

To enter the Natural objects maintenance facility, select the Natural option from the Natural ISPF Main Menu. The Natural Objects Entry Panel appears:

```
----------------------- NATURAL OBJECTS - ENTRY PANEL ------------------------
 COMMAND ===>


    Library       ===>
    Member        ===>
    Type          ===>            ( Blank,P,S,N,C,M,G,L,A,H,T,O,4,8,Z,3,5,7,9 )
    Status        ===>            ( Blank,S,C,OS,OC )
    Scan for      ===>
    Edit macro    ===>
    Set number    ===>









 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
       Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Specify the Natural object you wish to maintain in the input fields and enter a function command in the command line. The meaning of the input fields is explained in the following table:

Field	Meaning
`Library`	Natural library name. The library used last is displayed in this field. Select any other library by overtyping this name. You can use the wildcards (* _ < >) to list accessible libraries. (See the subsection ***Selection Windows and Wildcards*** in section *Command Logic*.). See also *Example: LIST (4)*.
`Member`	Name of required member. You can use the wildcards (* _ < >) to list members. (See the subsection ***Selection Windows and Wildcards*** in section *Command Logic*.)
`Type`	Type of member; possible options: `A` - Parameter area `C` - Copy code `G` - Global area `H` - Help routine `L` - Local area `M` - Map `N` - Subprogram `O` - Macro object `P` - Program `S` - Subroutine `T` - Text `Z` - Recording `3` - Dialog `4` - Class `5` - Processor `7` - Function `8` - Adapter `9` - Resource For example, you can enter an asterisk (*) in the `Member` field and P in the `Type` field to list all Natural members which are a program. A combination of up to five types is possible. For example, the notation `PSNM` lists all programs, subroutines, subprograms and maps according to the other selection criteria entered in the Entry Panel.
`Status`	Status of object. Use this field for selection criteria when listing members. Possible options: `C` - List members which have a cataloged object

Field	Meaning
	`OC` - List members which have a cataloged object only `OS` - List members which have an uncataloged source only `S` - List members which have a source
`Scan for`	Selection criterion for listing Natural members: all members as specified in the above fields are listed which contain the string entered here.
`Edit macro`	Name of macro object to be used as a model for the member. The specified macro is executed and loaded into the Editor. See section *Macro Facility* in the *Natural ISPF Programmer's Guide* for more details. Used with `LIST`, the list will contain all objects that use the specified macro as model.
`Set number`	Enter a set number to list members in the set created for the library. Alternatively, enter an asterisk (*) to list sets created for the library. Sets are created using Predict cross referencing for Natural objects (see the *Predict Reference* documentation).

Notes:

1. When editing a macro object, you must not use the Natural statement `END`.
2. You can access the Predict Cross-Reference menu directly from any Natural ISPF screen to maintain sets using the command: `NAT L X`.

Function Commands

The available function commands are:

Command	Parameter Syntax
`BROWSE`	*library*(*member*)
`CATALOG`	*library*(*member*)
`COMPARE`	*library*(*member*)
`COPY`	*library*(*member*), *object-type* *object-parms* `NODE=`*id*, `REP`
`DELETE`	*library*(*member*)
`DESCRIPTION`	*library*(*member*)
`DOWNLOAD`	*library*(*member*)
`EDIT`	*library*(*member*) `TYPE=`*t* `MACRO=`*name*
`EXECUTE`	*library*(*member*)
`EXPORT`	*library*(*member*), *target-environment*
`FORMAT`	*library*(*member*)
`HOLD`	*library*(*member*)
`INFORMATION`	*library*(*member*)
`LIST`	*library*(..*) `TYPE=`*t* `ST=`*s* `SC=`*string* `MACRO=`*name* `SET=`*n*

Command	Parameter Syntax
PLAY	*library*(*member*)
PRINT	*library*(*member*),*printer-name* CC NO
RENAME	*library*(*member*)
RUN	*library*(*member*)
SUBMIT	*library*(*member*),TARGET=*node-id*
UNCATALOG	*library*(*member*)
UPLOAD	*library*(*member*)
UNLOCK	*library*(*member*)

Notes:

1. The library parameter can be optional, depending on where the command is issued. If you specify only the member, the current library is assumed.
2. When you issue a CATALOG, RUN or SUBMIT command for a Natural program that includes inline macros, a macro expansion is performed before the program is compiled if the macro expansion function is enabled with the MACRO ON command or in your User Defaults profile (see also the section *Macro Facility* in the *Natural ISPF Programmer's Guide*).
3. The COPY command only works for Natural sources. If you wish to copy compiled objects, you must use the Natural SYSMAIN utility.
4. Before using the CATALOG command, it is recommended that you enable or disable the macro facility using the MACRO ON/OFF command as appropriate. If you issue a CATALOG command with MACRO ON for an object that does not use the macro facility, resources are wasted as the object is checked for the macro character.
5. If you issue any of the above function commands from outside the Natural facility and NAT is not the default object type specified in your profile, you must specify the object type N before the object parameters.

A full description of these commands is contained in section *Command Reference*. The object parameters correspond to the input fields on the Natural Objects Entry Panel.

Below are some examples of the INFORMATION, LIST, FORMAT and COMPARE functions using command syntax.

Example: INFORMATION

An information screen similar to the following is displayed as a result of the command:

```
INFORMATION N MYLIB(MYPROG)
```

The screen shows information on Natural member `MYPROG` in library `MYLIB`. The data provided are self-explanatory.

```
---------- NATURAL PROGRAM INFORMATION----MYLIB(MYPROG)-Subprogram ------------
COMMAND ===>

                     SOURCE      OBJECT       GDA USED:  ISP----G
    ORIGIN LIBRARY:  MYLIB       SYSNSPF      SUBROUTINES    ( From  1       )
    USERID:          BLI          ''          INTERPRET-L-COMMAND-2   ISP-SUBR
    TERMINAL ID:     DAEFTC46    DAEFTC11     MAKE-WINDOW             ISP-WINS
    DATE SAVED:      2011-01-31  2011-02-07   CHECK-MACRO-TYPE        ISPNIN-N
    TIME SAVED:      18:27:42    13:18:57
    OP SYSTEM:       MVS/ESA      ''
    TP SYSTEM:       COMPLETE     ''
    NAT VERSION:     8.2 0001     ''
    CODEPAGE:                    IBM01140



    SOURCE SIZE IN SOURCE AREA:     29766     MAXIMUM NO. OF VERSIONS  : 50_
    OBJECT SIZE IN DATSIZE:         10224     ACTUAL  NO. OF VERSIONS  : 22
    OBJECT SIZE IN BUFFER POOL:     35608     CURRENT VERS.MODIF LEVEL : 01.25
    OBJECT SIZE OF GLOBAL DATA:     16304
    OBJECT SIZE OF OPT-CODE:
    OPT STRING:
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

You can modify the `MAXIMUM NO. OF VERSIONS` for the object to override the default set by the system administrator (see also the subsection ***Versioning*** in section *Useful Features*).

Example: LIST (1)

The following display is the result of the command:

```
LIST N MYLIB(IS*) P
```

The list shows all Natural programs starting with `A` in the library `MYLIB`.

```
   LIST-NAT:MYLIB(IS*)P ---------------------------- Row 0 of 3 - Columns 010 076
    COMMAND===>                                                   SCROLL===> CSR
   MEMBER            PGMTYPE     SM S/C VERSION   USERID   DATE     TIME  VV.MM
** ********************************* top of list ******************************
   ISP-TECH          Program     S  S/C 8.2 0001  BLI      20110124 17:57 01.26
   ISP-TEC1          Program     S  S/C 8.2 0001  BLI      20110117 15:00
   ISP-TEC2          Program     S  S/C 8.2 0001  BLI      20110117 15:15
** ****************************** bottom of list ******************************
```

Meaning of the column headings:

Column	Meaning
MEMBER	Name of member
PGM TYPE	Type of member
SM	Natural mode. Possible values: R - Reporting S - Structured
S/C	Status of member. Possible values: S - Source C - Cataloged object S/C - Source and cataloged object
VERSION	Release version of Natural for member
USERID	ID of user who last modified member.
DATE	Date of last modification.
TIME	Time of last modification. If DATE/TIME of source and object is different, the save date of the source is shown in the list. It is highlighted to indicate that a time stamp difference exists between source and object.
VV.MM	Version number and modification level of the current version of the member. When a member is modified for the first time with versioning active, this field contains 01.01. With each modification with versioning on, the MM value is increased by one. This field can also contain: (blank) - No previous versions exist. The HOLD command for the member increases the VV value by one and resets the MM value to 00.

Note: You can change the layout of this list according to your needs. For detailed information, see the subsection *LAYOUT Command for Lists* in section *Useful Features*.

Example: LIST (2)

The following figure is the result of the command:

```
LIST N SYSISPE(EX*) TYPE=PGO SC=FILE-NAME
```

This list contains all program-type, global area-type and macro-type objects in the Natural library `SYSISPE` whose names start with `EX` and which contain the string `FILE-NAME`.

```
LIST-NAT:SYSISPE(EX*)PGO/SCAN=FILE-NAME----------- Row 0 of 9 - columns 010 071
 COMMAND===>                                                  SCROLL===> CSR
   MEMBER            PGMTYPE     SM S/C  NUM FIRST FOUND
** ****************************** top of list ******************************
   EXF1              Macro       S  S/C    5   1 #FILE-NAME(A32)
   EXF2              Macro       S  S/C    7   1 #FILE-NAME(A32)
   EXF3              Program     S  S/C    2   1 #FILE-NAME(A32) INIT <'AU
   EXF4              Program     S  S/C    2   1 #FILE-NAME(A32) INIT <'PE
   EXF6              Macro       S  S/C    1   MOVE     'NOFILE'   TO #OPT
   EXF9              Program     S  S/C    2   1 #FILE-NAME (A32) INIT <'AUT
   EXTG              Global      S  S/C    1   **DF            A  32 1#FILE-
   EXT1              Macro       S  S/C    7   * #FILE-NAME(A32)
   EXT2              Program     S  S/C    1   MOVE 'PERSONNEL' to #FILE-N
** *************************** bottom of list ****************************









Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

The meaning of the information in the right hand columns is:

Column	Meaning
`NUM`	Number of occurrences of specified string in this member
`FIRST FOUND`	Line with the first occurrence of specified string

The lists appear in Natural ISPF Editor format in browse mode. This means you can use all available Editor browse commands (`UP`, `DOWN`, `BOTTOM`, `TOP`, `FIND`, `LOCATE`).

When selecting a member from a list generated using the `Scan for` option with `EDIT` or `BROWSE`, the cursor is positioned to the first occurrence of this string in the member. The `RFIND` command places the cursor on the next occurrence of the string.

Example: LIST (3)

The following figure is the result of the command:

```
LIST N NSPFWORK(*) SET=*
```

The list contains all sets created for the Natural library `NSPFWORK`:

```
LIST-SET:NSPFWORK -------------------------------- Row 0 of 4 - Columns 001 072
 COMMAND===>                                                   SCROLL===> PAGE
   Nr Count Description
** ****************************** top of list ****************************
    4  1015 SELECT IS*
    7     4 DA-AREA ISP-PR-L (*) REF PROG * (*) BLOCK *
    8     2 PROG ISP-HL1N (*) REF PROG * (*) WITH * VIA *
    9     6 PROG ISP-FILN (*) REF PROG * (*) WITH * VIA *
   11    20 VIEW SYSTEM2 REF PROG * (*) USAGE *
** ***************************** bottom of list ****************************









Enter-PF13--PF14--PF15--PF16--PF17--PF18--PF19--PF20--PF21--PF22--PF23--PF24---
      Help  e :c  Save; Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Meaning of the column headings:

Column	Meaning
`Nr`	Set number assigned by Predict.
`Count`	Number of objects in the set.
`Description`	Information on object name and type according to which the set was created. The above list shows, for example, that set number 4 contains 1015 objects whose names start with `ISP`.

The available line command from the list of sets is `L` for `LIST`. This lists the objects in the selected set. The following figure illustrates the list of objects for set 1:

```
LIST-NAT:NSPF821(*)/SET=18 ---------------------- Row 0 of 19 - Columns 010 076
 COMMAND===>                                                  SCROLL===> CSR
   MEMBER            PGMTYPE     SM S/C VERSION   USERID   DATE     TIME  VV.MM
** ******************************** top of list ******************************
   ISP-COP1          Program     S  S/C 8.2 0001  BLI      20101114 14:43
   ISP-COP2          Program     S  S/C 8.2 0001  BLI      20110223 19:20
   ISP-COP3          Program     S  S/C 8.2 0001  BLI      20110120 18:53
   ISP-COP4          Program     S  S/C 8.2 0001  FHI      20110117 15:20
   ISP-ENVP          Program     S  S/C 8.2 0001  FHI      20101031 14:56
   ISP-LCPR          Program     S  S/C 8.2 0001  BLI      20110318 11:47
   ISP-PLYP          Program     S  S/C 8.2 0001  BLI      20101030 14:03
   ISP-PUTG          Program     S  S/C 8.2 0001  BLI      20101115 18:26
   ISP-RERN          Subprogram  S  S/C 8.2 0001  FHI      20101121 13:52
   ISP-RSTP          Program     S  S/C 8.2 0001  BLI      20110223 17:53
   ISP-SOUT          Program     S  S/C 8.2 0001  BLI      20110321 19:40
   ISP-STAS          Program     S  S/C 8.2 0001  BLI      20101115 19:21
   ISP-SUSU          Subprogram  S  S/C 8.2 0001  BLI      20110120 18:59
   ISP-WINS          Subroutine  S  S/C 8.2 0001  FHI      20110116 09:52
   ISP-WLOG          deleted
   ISPFERR           Program     S  S/C 8.2 0001  BLI      20101203 15:08
   ISP0600N          Subprogram  S  S/C 8.2 0001  FHI      20101018 17:10
   NAT00012          deleted
Enter-PF13--PF14--PF15--PF16--PF17--PF18--PF19--PF20--PF21--PF22--PF23--PF24---
      Help  Relis §End  !Br : t;fin !inf  Up    Down  Susp; Left  Right Exc :
```

Note that you can access this list of objects directly from the Natural Objects Entry Panel using the `Set number` field, or using the command:

```
LIST N NSPFWORK(*) SET=18
```

You can maintain the objects on this list as any other Natural object.

Example: LIST (4)

The following figure is the result of the command:

```
LIST N SYSISP*(A*)
```

The list contains all Natural libraries beginning with the string `SYSISP` for which you are authorized. For example:

```
Z*LIST-NLI:SYSISP* ------------------------------ Row 0 of 13 - Columns 010 050
 COMMAND===>                                                   SCROLL===> PAGE
   Library          Description
** ******************* top of list ******************
   SYSISPDB         INCORE DATABASE FOR CUSTOMER
   SYSISPE          NSPF example library
   SYSISPF          N-ISPF NATURAL EXAMPLES
   SYSISPFU         N-ISPF USER DATA
   SYSISPFX
   SYSISPH1         ISPF help texts
   SYSISPI          NSPF INTERFACE MODULES
   SYSISPIU         User information for ISPF
   SYSISPR          ISPF recordings
   SYSISPSC         ISPF INTERNAL TABLES COM-PLETE
   SYSISPST         ISPF tables for testmode
   SYSISPS1         ISPF system tables/menus
   SYSISPX
** ***************** bottom of list ****************




Enter-PF13--PF14--PF15--PF16--PF17--PF18--PF19--PF20--PF21--PF22--PF23--PF24---
      Help  e :c  Save; Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

You can now select a library with the line command `L` (for `LIST`). This displays all members in the library beginning with the letter `A`.

Notes:

1. The entries in the Description column of the above figure are taken from Natural Security (if installed).
2. Note that even if you wish to list libraries only, you must still enter the asterisk in parenthesis (*) for listing all members in the command syntax. If you enter only the library prefix (`SYSISP*`), it is interpreted as member prefix in the current Natural library.

Example: FORMAT

This function command applies only to Natural objects of type map. The map layout is displayed in a Natural ISPF Editor session. Modifiable fields (`AD=A` and `AD=M`) are replaced by a special filler character (_). Variable output fields (`AD=O`) are replaced by a period (full stop) (.).

```
FORMAT-NAT:NSPF211(ISPTIN-1)-Map ------------------------------ Columns 001 072
 COMMAND===>                                                    SCROLL===> CSR
****** ****************************** top of data ******************************
000001 DSName  : ..........................................
000002 Member  : ..........
000003 Language: _________
000004 Status  : _   (Test/Production)
000005           _   (Active/Inactive)
000006           _   (Enabled/Disabled)
000007 User    : ____
000008 Level   : ___
000009 Comment : ____________________________________________________
****** **************************** bottom of data ****************************
```

Example: COMPARE

The `COMPARE` function compares Natural sources stored in the Natural system file.

For example, to compare the source of the member `ISPJ---U` in the library `NSPFEXAM` with its source in the library `NSPF141`, enter the library and member name in the input fields and `CR` in the command line of the Natural Objects Entry Panel:

```
----------------------- NATURAL OBJECTS - ENTRY PANEL -------------------------
 COMMAND ===>


   Library       ===>
   Member        ===>
   Type          ===>            ( Blank,P,S,N,C,M,G,L,A,H,T,O,4,8,Z,3,5,7,9 )
   Status        ===>            ( Blank,S,C,OS,OC )
   Scan for      ===>
   Edit macro    ===>
   Set number    ===>









 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
       Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

The member entered in the Entry Panel is assumed to be the old source.

Press ENTER.

The following window opens in which you can enter the location of the new source and several compare options:

```
----------------------- NATURAL OBJECTS - ENTRY PANEL -------------------------
 COMMAND ===> CR


    Library       ===> NSPFEXAM
    Member        = +------COMPARE-NATURAL:NSPFEXAM(ISPJ---U)-------+
    Type          = !                                               ! ,H,T,O )
    Status        = ! Location of new source                        !        )
    Scan for      = !   Library                 NSPF141             !
    Edit macro    = !   Member                  ISPJ---U            !
    Set number    = !                                               !
                    ! Compare options                              !
                    !   Ignore comments         Y                   !
                    !   Ignore indentation      Y                   !
                    !   Display  differences    Y                   !
                    !   Show all differences    Y                   !
                    !   Number of sync lines        2               !
                    ! Enter to perform , PF3 to exit                !
                    +-----------------------------------------------+



 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
       Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

The meaning of the input fields is explained in the following table:

Field	Meaning
Location of new source:	
`Library`	Enter the name of the Natural library containing the source to be compared. The name of the library last used is displayed. You can select another library by overtyping it. Enter the asterisk wildcard (*) and press ENTER to list Natural libraries.
`Member`	Enter the name of the newer member to be compared. If Natural ISPF detects that this member was saved before the old member, member names are swapped and a message is displayed.
Compare options:	
`Ignore comments`	Enter `Y` to ignore comments while comparing sources.
`Ignore indentation`	Enter `Y` to ignore differences coming from indentation caused by the `STRUCT` Editor command.
`Display differences`	Enter `Y` to list all differences found in the source lines of a compared object. Otherwise, a message simply indicates whether the compared sources are identical or not.

Field	Meaning
Show all differences	Enter Y to display differences completely. Otherwise a short form of listing is used to print different ranges of more than 6 lines in the following way: FIRST LINE SECOND LINE ... LAST LINE -1 LAST LINE
Number of sync lines	Enter the number of synchronization lines. The default is 2. This parameter influences the compare mechanism. At least this number of consecutively equal lines must be found before the program assumes to have found an equal portion of code.

When you have made all entries, press ENTER. The successful compare displays an edit session containing the source differences. In our example, all options have been set to Y:

```
COMPARE-NAT:NSPFEXAM(ISPJ---U)-Subprogram->Struc >>> Old and new member swapped
 COMMAND===>                                                  SCROLL===> CSR
   OLD      NEW OLD=NSPF141(ISPJ---U) NEW=NSPFEXAM(ISPJ---U)
** ******************************** top of list ******************************
   0110 == 0110 1 #FUNCTION(A2)

   0120 >       1 #SES-DATA(A128)
         < 0120 PARAMETER USING ISPJ---L
         < 0130 PARAMETER

   0130 == 0140 1 #ERROR-CODE(N3)
        ...
   0190 == 0200   2 #JOB-PREFIX (A8)

   0200 >       LOCAL USING ISPJ---L

   0210 == 0210 LOCAL
        ...
   0260 == 0260   VALUE 'LS'

   0270 >           MOVE #SES-DATA TO #SES-DATA-J    /* GET SESSION DATA

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Last save date of old and new source are compared and old and new member are always set to reflect the sequence of their last save dates. In the example above, the message “Old and new member swapped” appears in the upper right corner of the screen. This means that Natural ISPF has detected that the “new” member (in library NSPF141) was saved before the “old” member (in library NSPFEXAM) and that the member names have been exchanged.

The format of the above screen is explained on the following page.

Output of the COMPARE program:

Column Heading				Explanation
OLD		NEW		
NNNN	... ==	*MMMM*	*SOURCE LINE*	All lines are equal in old and new source up to this line. *NNNN* and *MMMM* are equal
NNNN NNNN	> > <	*MMMM MMMM* *MMMM MMMM*	*SOURCE LINE SOURCE LINE SOURCE LINE SOURCE LINE SOURCE LINE*	Old lines *NNNN* have been modified to the new lines *MMMM*. The number of lines in OLD and NEW may be different. To perform this modification, delete the lines marked with > and add the ones marked with <.
NNNN *NNNN*	== ... ==	*MMMM* *MMMM*	*SOURCE LINE* *SOURCE LINE*	Such a block of three lines indicates that the *NNNN*-lines are equal to the *MMMM*-lines. NNNN and *MMMM* may be different.
NNNN NNNN NNNN	> > >		*SOURCE LINE SOURCE LINE SOURCE LINE*	Only lines specifying line numbers in the OLD column marked with > have been deleted.
NNNN *NNNN*	== ... ==	*MMMM* *MMMM*	*SOURCE LINE* *SOURCE LINE*	Another block of equal lines. The number of lines is equal as well (of course).
	< < <	*MMMM MMMM MMMM*	*SOURCE LINE SOURCE LINE SOURCE LINE*	Only lines specifying line numbers in the NEW column, all marked with < , have been inserted.
NNNN	== ...	*MMMM*	SOURCE LINE	These last lines indicate that the rest of the source is equal from line number *NNNN* in the OLD version and line *MMMM* in the new version.

Differences between Old and New coming from different indentation as a result of executing STRUCT are ignored. Single equal lines within a block of modified lines are also ignored, i.e. the whole block including the single equal lines are marked as modified. It takes at least two non empty lines (lines containing only an asterisk (*) are considered to be empty) to cause an output of a block of equal lines.

Line Commands

From lists of Natural sets

Select a set from a list of Natural sets with the line command `L` (for `LIST`). This lists the members in the set (see also the **third example of the `LIST` command**).

From lists of Natural libraries

Select a library from a list of Natural libraries with the line command `L` (for `LIST`). This lists the members in the library (**see also the fourth example of the `LIST` command**).

From lists of Natural members

To select a member for further maintenance from a list of Natural objects type a line command in the input field preceding the member name and press ENTER. Each line command is an abbreviation of a function command (but note the `LIST` command for a member):

Line Command	Corresponding Function Command
B	BROWSE
CP	COPY
CR	COMPARE
CT	CATALOG
D	DELETE
DS	DESCRIPTION
DW	DOWNLOAD
E	EDIT
EX	EXPORT
FR	FORMAT
HL	HOLD
I	INFORMATION
L	LIST previous versions of the member
PL	PLAY
PR	PRINT
R	RENAME
RU	RUN
SB	SUBMIT
U	UNCATALOG
UP	UPLOAD
XE	EXECUTE

Line Command	Corresponding Function Command
UL	UNLOCK

Line commands can be used as valid abbreviation for function commands entered in the command line of any screen.

Local Commands

When displaying Natural objects in Editor format, you can use the following local commands:

In LIST mode:

CATALL

From a list of Natural objects, you can catalog multiple objects using the CATALL local command. The following are examples of the CATALL command:

Command	Meaning
CATALL	Catalogs all objects in the list
CATALL ISP*	Catalogs all objects in the list whose names start with ISP.

When a CATALL command is issued, a window opens on your screen showing the name of the program being cataloged. After the cataloging process, those objects for which an error was detected are indicated by the message *ERROR in the message field, and the nature of the error is displayed in the statistical data fields.

ALL, LAYOUT, RELIST, SORT

You can use the commands ALL, LAYOUT, RELIST and SORT. For detailed information, see the sub-sections in section ***Useful Features***.

The following figure shows the result of the CATALL EX* command issued from a list of Natural objects:

```
LIST-NAT:SYSISPE---------------------------------------- >>> 2 errors detected
 COMMAND===> catall ex*                                        SCROLL===> CSR
   MEMBER            PGMTYPE     SM S/C  NUM FIRST FOUND
** ***************************** top of list ****************************
   EXF1              Macro       S  S/C    5   1 #FILE-NAME(A32)
   EXF2              Macro       S  S/C    7   1 #FILE-NAME(A32)
   EXF3              Program     S  S/C    2   1 #FILE-NAME(A32) INIT <'AU
   EXF4     *ERROR   Program     ERROR 2 AT LINE 20
   EXF6              Macro       S  S/C    1   MOVE     'NOFILE'   TO #OPT
   EXF9              Program     S  S/C    2 1 #FILE-NAME (A32) INIT <'AUT
   EXTG              Global      S  S/C    1 **DF            A  32 1#FILE-
   EXT1              Macro       S  S/C    7   * #FILE-NAME(A32)
   EXT2     *ERROR   Program     ERROR 2 AT LINE 20
** *************************** bottom of list ***************************





Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Note: Before using the `CATALL` command, it is recommended that you enable or disable the macro facility using the `MACRO ON/OFF` command as appropriate. If you issue a `CATALL` command with `MACRO ON` for an object that does not use the macro facility, resources are wasted as the object is checked for the macro character.

In EDIT mode:

You can use the following Natural commands as local commands from the Editor command line when editing a Natural object:

Command	Meaning
`CHECK`	Checks syntax of the Natural program being edited. *
`IMPORT`	Edit mode only: imports a PC file or Con-nect document into the Natural member (see the section ***Useful Features***).
`SM OFF`	Sets structured mode off.
`SM ON`	Sets structured mode on.
`STOW 'text'`	Stores the Natural program in source and object form. ** When stowing a program after modification with versioning active, you can specify a reason for the change with *text* parameter.
`STRUCT`	Performs structural indentation of Natural source statements and identifies any structural inconsistencies (not applicable for macro-type programs).
`TYPE <t>`	Specifies Natural program type, where *t* can stand for any of the following:

Command	Meaning
	C - Copy code H - Help routine N - Subprogram O - Macro object P - Program S - Subroutine T - Text 4 - Class

* If the Natural object is a macro object, the CHECK command also checks the processing statements and variables to be substituted. The command does not check that the lines generated by the macro object are valid Natural source (see the section *Macro Facility* in the *Natural ISPF Programmer's Guide*).

** If you issue a STOW command for a Natural program that includes inline macros, a macro expansion is performed before the program is compiled, if the macro expansion function is enabled with the MACRO ON command or in your User Defaults profile (see also the section *Macro Facility* in the *Natural ISPF Programmer's Guide*).

For more information on Natural commands, see the Natural documentation.

You can also use a special COPY command which may be useful when editing Natural programs (use Editor target line commands A, B or O to mark the place where the data are to be copied). You can copy other Natural objects or other object types into the edit area. The following object types can be copied:

Object type	Meaning
BF	BS2000 file
D	Dataset (sequential)
DJ	Job (z/VSE)
FIL	z/VSE file
J	Job (z/OS)
LIB	CA Librarian member
LMS	LMS element
LMV	LMS element version
MAC	Macro object
MEM	z/VSE member

Object type	Meaning
N	Natural object
O	Output file in workpool
P	PDS member
PAN	CA Panvalet member
S	Job SYSOUT (z/OS)
V	Database view

Examples:

Command	Meaning
COPY *mapname*	Generates a Natural INPUT statement for the Natural map *mapname* and copies the map's variable definition into the current program at the marked place.
COPY VIEW *viewname*	Copies the definition of view *viewname* into the current program.
COPY *data-area-name*	Generates a data definition source from the Natural data area and copies it into the current program at the marked place.
COPY MACRO *name*	Performs a macro expansion of the macro object *name* and copies the result into the current member at the marked place.

If you issue the COPY command without parameters, you are prompted for object type and object name.

OUTPUT

Starts an edit session with the output of the current program in the user workpool (only valid after a RUN command issued from the edit session).

REGENERATE / REG

Available for Natural programs written using the Edit macro option. Reexecutes the specified macro object and writes the result in protected lines in the current edit session. Any defined user code remains in place. For details, see the section *Macro Facility* in the *Natural ISPF Programmer's Guide*.

Previous Versions

Previous versions of Natural objects can be retrieved and maintained (see also the line command `L` for `LIST`). They are separate objects in Natural ISPF, accessible via the Natural Objects Entry Panel, or using function commands with object type `NV`. To activate the versioning feature, you must issue the command `VERSIONS ON` before starting your edit session. For details, see the subsection ***Versioning*** in the section *Useful Features*.

"Write-To-Workpool" Feature

The Write-To-Workpool option is a simple yet powerful feature useful for checking the output of Natural programs. Using the Editor, you can write a Natural program and include a statement defining a printer for the program output. The Write-To-Workpool feature allows you to define the workpool as destination printer, for example:

```
DEFINE PRINTER(1) OUTPUT 'WORKPOOL'
```

When a write to printer 1 is performed (using a `WRITE`, `PRINT` or `DISPLAY` statement), the program output is written to the user workpool. Several reports can be written to the workpool by defining the workpool as destination for multiple printers (`DEFINE PRINTER(2)`, etc).

You can use the `WORKPOOL` option from the Natural ISPF Main Menu to display and maintain the output (see the subsection ***User Workpool***). Note that each time you run the program (`RUN` command), the existing output of the program in the user workpool is overwritten with the new output.

Using the split-screen feature, the Natural programmer can edit a program in one screen subsection and immediately see the resulting output in the other screen subsection by issuing the `RUN` command from the edit session. Checking and debugging programs is thus made very convenient.

The following is an example Natural program illustrating the use of the Write-To-Workpool feature.

```
*
*
DEFINE PRINTER(1) OUTPUT 'WORKPOOL'
*
READ (100) AUTOMOBILES BY MAKE STARTING FROM 'C'
   WRITE(1)  MAKE COLOR MODEL HORSEPOWER WEIGHT
             NUMBER-OF-CYLINDERS SERIAL-NUMBER
END
```

If you issue the `RUN` command for this program, it reads the file `AUTOMOBILES` and writes the contents of the specified fields to the user workpool, where the program output can be accessed (see the subsection ***User Workpool***).

The figure below illustrates two Natural ISPF sessions in split-screen mode, with the Natural program in the upper session and the program output in the lower session:

```
EDIT-NAT:NSPF211(EXW1)-Program->Report-Free-46K ----------- >>> Source EXW1 run
 COMMAND===>                                                  SCROLL===> CSR
****** ****************************** top of data ******************************
000010 * DEMO: WORKPOOL
000020 *
000030 DEFINE PRINTER(1) OUTPUT 'WORKPOOL'
000040 *
000050 READ (20) AUTOMOBILES BY MAKE STARTING FROM 'C'
000060    WRITE(1) MAKE COLOR MODEL HORSEPOWER WEIGHT
000070             NUMBER-OF-CYLINDERS SERIAL-NUMBER
EDIT-OUT:EXW1 ------------------------------------------------- Columns 001 072
 COMMAND===>                                                  SCROLL===> CSR
****** ****************************** top of data ******************************
000001 Page      1                                                  94-12-27  1
000002
000003 CHRYSLER       green      DODGE CORONET CUSTOM  255   4150   6 035549448
000004 CHRYSLER       GREEN      DODGE CHALLENGER SIX  150   3160   6 J92314635
000005 CHRYSLER       BROWN      PLYMOUTH ROAD RUNNER  330   3695   6 L32433047
000006 CHRYSLER       YELLOW     DODGE CHALLENGER SIX  150   3160   6 N58644909
000007 CHRYSLER       WHITE      NEWPORT ROYAL         175   4210   6 J90372307
000008 CHRYSLER       WHITE      DODGE MONACO          190   4310   6 089730037
000009 CHRYSLER       WHITE      PLYMOUTH FURY II      175   4040   6 L15260038
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Concurrent Editing of Natural Objects

When you save/stow a Natural program, Natural ISPF runs a check to see if the same program has been modified by another user or another session whilst you were editing. If this is the case, you are notified by a message and the save/stow operation is not executed.

You can use the `BROWSE` command to inspect the Natural object and you can decide whether you wish to override it with your latest modifications or not. If you wish to override it with your latest modifications, you can either:

- use the `REPLACE` command for the existing object, or
- delete the existing Natural object and then save/stow the version with your latest changes.

Natural Views

The VIEWS option on the Natural ISPF Main Menu allows you to select a data view, list its field names (view definition) and access the database to display the field contents. Natural programmers can use this facility to check their files in one session while editing programs in other sessions.

How to reference views and record fields in Natural programs is described in the Natural documentation.

To enter the views facility, select the VIEWS option from the Natural ISPF Main Menu. This displays the Natural View Entry Panel on your screen:

```
S*>>---------------------NATURAL-VIEW--ENTRY-PANEL---------------------------
COMMAND ===>


   View Name     ===>
   Dbid          ===>                                  (For selection list)
   Fnr           ===>                                  (For selection list)
   Record Fields ===>                                          (Field list)
   Start value   ===>
   End value     ===>
   Max Records   ===>  100
   Password      ===>









Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

You can specify the object you wish to display in the input fields and enter a function command in the command line.

Meaning of the input fields:

Field	Meaning
`View name`	Enter a view name. You can use the wildcards (* _ < >) to list available views. (See the subsection ***Selection Windows and Wildcards*** in section *Command Logic*.)
`Dbid`	Enter database ID of view. The value 0 (zero) is a valid database ID.
`Fnr`	Enter file number of view.
`Record fields`	Enter field names for content display. Adabas short names are also accepted. The first field must be a descriptor or superdescriptor.
`Start value`	Enter value with which to start the list of records.
`End value`	Enter value with which to end the list of records.
`Max records`	Enter maximum number of records to be listed (default is 100).
`Password`	Enter the Adabas security password for the `BROWSE` function (required if the file is protected by Adabas security).

Function Commands

The following function commands are available for view handling:

Command	Parameter Syntax
`BROWSE`	*`view-name`*, `PASSWORD=`*`pswd`*
`COPY`	*`view-name`*, *`object-type object-parameters`*
`DEFINITION`	*`view-name`*
`DESCRIPTION`	*`view-name`*
`DOWNLOAD`	*`view-name`*
`LIST`	*`view-name`* (or ..*) `DBID=`*`id`* `FNR=`*`n`*
`UPLOAD`	*`view-name`*

For an explanation of these commands, see the section ***Command Reference***. The object parameters correspond to the input fields on the Natural View Entry Panel.

Notes:

1. The `COPY` command copies the generated data definition statements. (For information on generated data definition statements, see the subsection ***Local Commands***).
2. If a view is not explicitly specified in the `View Name` field, the default function command is `LIST`.
3. If a view is specified, but no record fields are entered, the default function command is `DEFINITION`.
4. If a view and one or more record fields are specified, the default command is `BROWSE`.
5. The `DESCRIPTION` function is also available as a line command for view elements. See **the example for the `DEFINITION` command**.

If you issue any of the above commands from outside the views facility, the object type parameter `V` must precede any object parameters in the command syntax.

Examples of function commands using full command syntax are described in the following subsections.

Example: LIST

The following figure illustrates a list of views displayed using the command:

```
LIST V *
```

The views are listed according to name, database ID, file number and type:

```
LIST-VIW:* ------------------------------------- Row 0 of 92 - Columns 034 055
 COMMAND===>                                                   SCROLL===> CSR
   VIEW NAME                                  DBID FNR TYPE
** ********************* top of list *********************
   ACCOUNTING                                  148  34 P
   ACTIVE-JOBS                                 148  29 P
   ADDRESS-SPACE                               148  21 P
   ALLOCATIONS                                 148  22 P
   ARCHIVE                                     148 211 P
   AUTOMOBILES                                   0   2 A
   CATALOG                                     148   8 P
   CATALOG-UPDATE                              148  10 P
   COMMAND                                       1   1 C
   COMMON-DATA                                 148  33 P
   CONSOLE                                     148  35 P
   CONSOLE-LOG                                 148  25 P
   CONTAINER                                   147 199 A
   COPY-FILE                                   148  37 P
   DEVICE-NAMES                                148  30 P
   DICTIONARY                                  255 253 A
   EMPLOYEES                                     0  53 A
   EMPLOYEES-VS                                254   1 V
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Use the `DBID=` option and/or the `FNR=` option in the command syntax to display views on a specific database and/or with a specific file number. To select any view for further handling, use any line command described in the subsection *Line Commands*.

Example: BROWSE

The following prompt window opens as a result of the command:

```
BROWSE V AUTOMOBILES
```

```
-------------------------- NATURAL ISPF MAIN MENU -----------------------------
  OPTION  ===> BROWSE V AUTOMOBILES
                                                                Userid   BRY
                                                                Time     15:11:53
     0    PROFILE    - Profile maintenance                      Terminal DAEFTC30
     1    NATURAL    - Work with NATURAL objects                Library  BRY
    +-----------------------------BROWSE-VIEW:------------------------------+
    !                                                                        !
    ! View Name     ===> AUTOMOBILES                                        !
    ! Record Fields ===> MAKE MODEL WEIGHT                                  !
    ! Start value   ===> F                                                  !
    ! End value     ===> H                                                  !
    ! Max Records   ===> 20                                                 !
    ! Password      ===>                                                    !
    +-------------------------------------------------------------------------+
```

The fields in the window correspond to the parameter fields on the Natural View Entry Panel. The screen on the following page shows database records listed as a result of entering the parameters as shown in the window on this page:

```
BROWSE-VIW:AUTOMOBILES-----------------------------------------Columns 001 043
COMMAND===>                                                   SCROLL===> CSR
   MAKE           MODEL                WEIGHT
** *************** top of list **************
   FERRARI        365 GTB/4             2640
   FERRARI        365 GTB/4             2640
   FERRARI        DINO 246 GT           2380
   FERRARI        DINO 246 GT           2380
   FERRARI        365 GTB/4             2640
   FERRARI        DINO 246 GT           2380
   FERRARI        DINO 246 GT           2380
   FERRARI        DINO 246 GT           2380
   FERRARI        365 GTB/4             2640
   FERRARI        DINO 246 GT           2380
   FERRARI        DINO 246 GT           2380
   FERRARI        DINO 246 GT           2380
   FERRARI        365 GTB/4             2640
   FERRARI        DINO 246 GT           2380
   FERRARI        DINO 246 GT           2380
   FERRARI        365 GTB/4             2640
   FERRARI        DINO 246 GT           2380
   FERRARI        365 GTB/4             2640
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Cursor
```

The display shows the first 20 automobiles in a list starting with the letter `F` and ending with `H`, with information on make, model and weight.

- Alternatively, you can enter an asterisk (*) in the `Record Fields` field in the prompt window and press ENTER. The following window opens:

```
-------------------------- NATURAL ISPF MAIN MENU -----------------------------
 OPTION  ===> BROWSE V AUTOMOBILES
                                                                Userid   BRY
                  +-----------------BROWSE-VIEW:------------------+   15:11:53
   0    PROFILE    ! Select fields:                              ! l DAEFTC30
   1    NATURAL    !                                               !   BRY
  +--------------- ! _ 1   CAR-DESCRIPTION                        ! -----+
  !                ! _ 2 D MAKE                             A 14   !      !
  ! View Name      ! _ 2 D MODEL                            A 20   !      !
  ! Record Fields  ! _ 2 D BODY-TYPE                        A 15   !      !
  ! Start value    ! _ 2 D NUMBER-OF-CYLINDERS              N 2.0  !      !
  ! End value      ! _ 2 D HORSEPOWER                       N 3.0  !      !
  ! Max Records    ! _ 2   PISTON-DISPLACEMENT              N 5.0  !      !
  ! Password       ! _ 1   CAR-DETAIL                              !      !
  +--------------- ! _ 2   WEIGHT                           N 5.0  ! -----+
                   ! _ 2 D COLOR                            A 10   !
   10   SYSTEM     ! Entr-PF3--PF7--PF8--                          !
   11   ADMIN      ! Down End  Top  Down                           !
   NEWS CHANGES    +-----------------------------------------------+
   END  EXIT       - Exit NATURAL ISPF
   HELP HELP       - Display help information

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right TOP
```

Select one or more fields with any character and press PF3 to display field contents.

- If you select only 1 field here, a `HISTOGRAM` statement is generated for Adabas files. In addition to the field contents, the variable `*NUMBER` is also displayed:

```
BROWSE-VIW:AUTOMOBILES -------------------------- Row 0 of 58 - Columns 016 026
 COMMAND===>                                                   SCROLL===> CSR
   MAKE           NMBR
** ******* top of list ******
   ALPINE                   6
   AMERICAN MOTOR           3
   AMERICAN MOTOR          85
   AUDI                     7
   AUDI-80                  1
   BMW                     20
   CHRYSLER               167
   CITROEN                  8
   DAIMLER-BENZ             1
   DATSUN                  11
   DE TOMASO               11
   DKW                      1
   FERRARI                 30
   FIAT                    28
   FORD                   133
   GENERAL MOTORS         281
   GOLF                     3
   ISO                      6
   JAGUAR                  22
   LAMBORGHINI              7
```

Example: DEFINITION

The following view definition display is the result of the command:

```
DEFINITION V EMPLOYEES
```

```
DEFINITION-VIW:EMPLOYEES-(254,1) ------------------------------ Columns 001 074
 COMMAND===>                                                    SCROLL===> CSR
   T L  DB  Name                              F Leng  S D Remarks
** ****************************** top of list ******************************
     1  AA  PERSONNEL-ID                      A    8    D
            HD=PERSONNEL/ID
   *             CNNNNNNN
   *             C=COUNTRY
   G 1  AB  FULL-NAME
     2  AC  FIRST-NAME                        A   20  N
     2  AD  MIDDLE-NAME                       A   20  N
     2  AE  NAME                              A   20    D
     1  AF  MAR-STAT                          A    1  F
            HD=MARITAL/STATUS
   *             M=MARRIED
   *             S=SINGLE
   *             D=DIVORCED
   *             W=WIDOWED
     1  AG  SEX                               A    1  F
            HD=S/E/X
     1  AH  BIRTH                             N  6.0
            HD=DATE/OF/BIRTH
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Meaning of the column headings:

Column	Meaning
T	Field type
L	Level
DB	Database field name
Name	Field name
F	Field format. For example: A - Alpha field B - Binary field N - Numeric field
Leng	Field length
S	Special option. Possible values: blank - No option defined F - Fixed length N - Nulls suppressed
D	Descriptor. Possible values:

Column	Meaning
	D - Field descriptor S -Superdescriptor
Remarks	Comments

From the definition screen, you can use the DS line command to edit the description of a view element. An example is illustrated on the next page.

The following screen appears as the result of the line command DS entered for view element FIRST-NAME.

```
EDIT-PRD:EL:PERSONNEL(FIRST-NAME) ---------------------------- Columns 001 072
 COMMAND===>                                                  SCROLL===> CSR
****** ****************************** top of data ******************************
000001           ====================================
000002           Elem. field additional description
000003           ====================================
000004 A person's first name is required only if the surname (NAME) is not
000005 unique.
****** **************************** bottom of data ****************************









Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right :s
```

You can edit and save the description as required.

Example: DESCRIPTION

The following view description display is the result of the command:

```
DESCRIPTION V AUTOMOBILES
```

```
EDIT-PRD:FI:AUTOMOBILES ------------------------------------- Columns 001 072
 COMMAND===>                                                  SCROLL===> CSR
****** ****************************** top of data ******************************
000001           ====================================
000002              File additional description
000003           ====================================
000004 This is the famous automobiles file.
****** **************************** bottom of data ****************************









Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right :s
```

You can edit the description as required and save it.

Line Commands

The following line commands are available from a list of views generated using the `LIST` command:

Line Command	Corresponding Function Command
B	BROWSE
CP	COPY
DF	DEFINITION
DS	DESCRIPTION
DW	DOWNLOAD
UP	UPLOAD

Line commands can also be used as valid abbreviations of function commands entered in the command line.

The `DS` line command is also available for view elements displayed in the view definition, see **the example for the `DEFINITION` command**.

Local Commands

Besides Editor browse commands, you can use one local command from the Editor command line when displaying a view definition. The command:

```
GENERATE
```

generates data definition statements for a Natural source.

Example:

The following view definition is displayed as a result of the command:

```
DEFINITION V EMPLOYEES-VS
```

```
DEFINITION-VIW:EMPLOYEES-VS-(254,1) --------------------------- Columns 001 074
 COMMAND===>                                                    SCROLL===> CSR
   T L  DB  Name                            F Leng  S D Remarks
** ******************************** top of list ******************************
     1  AA  PERSONNEL-ID                    A    8    D
            HD=PERSONNEL/ID
   *             CNNNNNNN
   *             C=COUNTRY
   G 1  AB  FULL-NAME
     2  AC  FIRST-NAME                      A   20  N
     2  AD  MIDDLE-NAME                     A   20  N
     2  AE  NAME                            A   20    D
     1  AF  MAR-STAT                        A    1  F
            HD=MARITAL/STATUS
   *             M=MARRIED
   *             S=SINGLE
   *             D=DIVORCED
   *             W=WIDOWED
     1  AG  SEX                             A    1  F
            HD=S/E/X
     1  AH  BIRTH                           N  6.0
            HD=DATE/OF/BIRTH
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

If you now issue the `GENERATE` command from the command line, the following data definition statements are generated:

```
DEFINITION-VIW:EMPLOYEES-VS-(254,1) --------------------------- Columns 001 074
 COMMAND===>                                                    SCROLL===> CSR
   T L  DB  Name                             F Leng  S D Remarks
** ****************************** top of list ******************************
   1 EMPLOYEES-VS-VIEW  VIEW OF EMPLOYEES-VS
     2 PERSONNEL-ID
     2 FULL-NAME
       3 FIRST-NAME
       3 MIDDLE-NAME
       3 NAME
     2 MAR-STAT
     2 SEX
     2 BIRTH
     2 FULL-ADDRESS
       3 ADDRESS-LINE
       3 CITY
       3 ZIP
       3 FILLER
       3 COUNTRY
     2 TELEPHONE
       3 AREA-CODE
       3 PHONE
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Note: If you use the `COPY` command for a view, the generated data definition statements are copied.

Natural Error Messages Member: ERRMENU

The `ERROR` option on the Natural ISPF Main Menu allows you to edit the long and short texts of Natural error messages and user-defined messages.

To enter the error facility, select the `ERROR` option from the Natural ISPF Main Menu. The Natural Error Messages Entry Panel appears:

```
------------------- NATURAL ERROR MESSAGES - ENTRY PANEL ----------------------
COMMAND ===>


   Library       ===>
   Type          ===>                          (N,U)
   Number from   ===>
   Number to     ===>
   Language code ===>
   Scan for      ===>
```

You can specify the Natural error message you wish to maintain in the input fields of the Natural Error Messages Entry Panel and enter a function command in the command line.

The meaning of the input fields is explained in the following table:

Field	Keyword	Meaning
Library	-	Natural library name. The library used last is displayed in this field. Select any other library by overtyping this name.
Type	TYPE	Possible values are: N - Natural error message U - User-defined message (default) If the Library field is filled, message type U is assumed.
Number from	FROM	Starting value for the range of error messages to be listed, or the number of the error message to be browsed or edited.
Number to	TO	Ending value for the range of error messages to be listed. Required only when invoking the LIST function.
Language code	LANGUAGE	If this field is not filled, the value of *LANGUAGE is used.
Scan for	SCAN	Selection criterion for listing Natural error texts: all error messages as specified in the above fields are listed whose short text contains the string entered here.

Function Commands

The available function commands are:

Command	Parameter Syntax
BROWSE	*library*(*error-number*) TYPE=*x* LANG=*l*
DELETE	*library*(*error-number*) TYPE=*x* LANG=*l*
EDIT	*library*(*error-number*) TYPE=*x* LANG=*l*
LIST	*library*(*error-from*) TO=*error-to* TYPE=*x* SCAN=*string* LANG=*l*

Note: If you issue any of the above function commands from outside the Natural Error Messages facility, you must specify the object-type parameter E before the object parameters.

Line Commands

The following line commands are available from a list of messages generated using the LIST command:

Line Command	Corresponding Function Command
BR	BROWSE
D	DELETE
E	EDIT

The BROWSE and EDIT functions display short and long error texts in one edit session. This enables you to maintain both text types on one screen. The short text is always displayed in Line 1. Lines 2 to 21 contain the long text. You cannot save the message when more than 21 lines are in the Editor session.

Example of EDIT screen:

```
EDIT-ERR:(3022)/TYPE=N ---------------------------------------- Columns 001 072
 COMMAND===>                                                    SCROLL===> CSR
****** ****************************** top of data ******************************
000001 Invalid command.
==msg> Start of long error message
.TX    Invalid command.
000004 .
000005 .
.EX    The command entered was invalid:
000007 .
000008    - invalid command code;
000009 .
000010    - access-only user attempted to issue update command;
000011 .
000012    - non-ET-logic user issued BT command;
000013 .
000014    - CLU user issued ET command.
000015 .
000016 .
000017 .
000018 .
000019 .
.AC    Check program and correct error.
000021 .
```

Local command NEXT

The `NEXT` command is available with the `EDIT`/`BROWSE` functions. In `BROWSE` mode, the next error text found is displayed. In EDIT mode, the current text is saved, if it has been modified, and the next error text is read from the system file for editing.

Predict Descriptions

Natural ISPF allows you to edit the long description of any Predict object type. To access the Predict description facility, select the Predict option from the Natural ISPF Main Menu. The Predict Description Entry Panel appears:

```
-------------------- PREDICT DESCRIPTION - ENTRY PANEL -----------------------
COMMAND ===>


   Library       ===>
   Member        ===>
   Type          ===>                    ( DA,EL,FI,KY,MO,PR,RL,RP,SY,US,VE)
   Object Name   ===>
   File name     ===>









Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right :s
```

Select a description by specifying the corresponding object in the parameter fields and pressing ENTER.

Note: The selected object must already be defined in Predict.

Meaning of the input fields:

Field	Meaning
`Library`	Name of Natural library: for objects of type `PR` (program), you can specify the implementation pointer.
`Member`	Name of Natural program: for objects of type `PR` (program), you can specify the implementation pointer.
`Type`	Predict object type (for example, `PR` for program, `DA` for database, `FI` for file, `EL` for file element, `VE` for verification).
`Object Name`	Name of the Predict object. For object type `EL` (file element), the element name.
`File Name`	For object type `EL` only (file element). Enter the file name here and an element name in `Object Name` field.

Function Commands

Predict descriptions are separate objects in Natural ISPF with object type PRD. This means you can start an edit session with a Predict description from any system screen using the following function commands:

Command	Object Parameter Syntax
BROWSE	*library*(*member*)
BROWSE	*object-name* TYPE=*t* FILE=*file-name*
EDIT	*library*(*member*)
EDIT	*object-name* TYPE=*t* FILE=*file-name*
LIST	*object-name* TYPE=*t* FILE=*file-name*

Note: If you issue any of the above function commands from outside the Predict description facility, you must specify the object-type parameter PRD before the object parameters.

Example: EDIT

```
E PRD AUTOMOBILES TYPE=FI

L PRD A* TYPE=PR
```

See also the function command DESCRIPTION in section *Command Reference*.

Con-nect Documents

The Natural ISPF interface to Con-nect enables you to access Con-nect documents for which you are authorized. Con-nect documents can be listed from Con-nect cabinets and folders in Editor format. Information can be displayed for a document, and documents can be added, deleted and maintained using the Editor in the same way as any other Natural ISPF object.

Additionally, using the Natural ISPF multi-session and split-screen features, you can easily transfer data to and from Con-nect documents.

To enter the Con-nect document maintenance facility, select the Con-nect option from the Natural ISPF Main Menu. The Con-nect Documents Entry Panel appears:

```
--------------------- CON-NECT DOCUMENTS - ENTRY PANEL ------------------------
COMMAND ===>


   Cabinet       ===>
   Folder        ===>
   File          ===>
   Document      ===>
   Password      ===>                        ( If password protected )
   Keyword       ===>
   Sort sequence ===>                        ( A/D )
   Max documents ===>









Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right :s
```

You can specify the Con-nect document you wish to maintain in the input fields and enter a function command in the command line. Meaning of the input fields:

Field	Meaning
`Cabinet`	Name of cabinet in which the document is or is to be filed.
`Folder`	Name of folder in which the document is or is to be filed. Note that Natural ISPF does not support listing of folders.
`File`	Name of file in which the document is or is to be filed. Note that Natural ISPF does not support listing of files.
`Document`	Name of document to be added or maintained. Leave blank or use a combination of strings and wildcards (* _ < >) to list documents in the specified folder/file. (See the subsection ***Selection Windows and Wildcards*** in section *Command Logic*.)
`Password`	Password, if the cabinet is password-protected.
`Keyword`	One of the keywords assigned to the document when using the `LIST` function. If you enter a keyword, all documents with this keyword are listed. The `Folder`, `File` and `Document` fields are ignored. Keywords can be assigned to existing documents using the `INFORMATION` function.
`Sort sequence`	Relevant with `LIST`. Specifies whether to list documents in ascending (`A`) (default) or descending (`D`) alphabetical order.
`Max documents`	Relevant with `LIST`. Specifies the maximum number of documents to be listed.

Function Commands

The available function commands for Con-nect documents are as follows:

Command	Object Parameter Syntax
BROWSE	*cabinet*(*document*) PASSWORD=*p*
DELETE	*cabinet*(*document*) PASSWORD=*p*
EDIT	*cabinet*(*document*) FOLDER=*fo* FILE=*fi* PASSWORD=*p*
INFORMATION	*cabinet*(*document*) PASSWORD=*p*
LIST	cabinet(*_*) FOLDER=fo FILE=fi PASSWORD=p KEYWORD=k
PRINT	*cabinet*(*document*) PASSWORD=*p*

The object parameters correspond to the input fields on the Con-nect Documents Entry Panel. For a full description of these commands, see section *Command Reference*.

Notes:

1. If you issue any of the above commands from outside the Con-nect document facility, you must specify the object type DOC before the object parameters.
2. The cabinet parameter is optional. Your personal cabinet is the default.
3. When issuing the PRINT command for a Con-nect document, a window prompts you for the printer name and you can specify whether the document is to be printed formatted or not.

Example: LIST

The following figure shows an example of a list of all documents in a Con-nect file using the command:

```
LIST DOC MBE(*) FOLDER=WORK FILE=ISPF
```

The command lists all documents from the ISPF file in the WORK folder in cabinet MBE:

```
LIST-DOC:MBE(*)/FOLDER=WORK ---------------------- Row 0 of 2 - Columns 045 076
 COMMAND===>                                                   SCROLL===> CSR
   Document                          Int.Number          Created    File
** ******************************** top of list ******************************
   Editor 134                        0001485546          1989-11-24 ISPF
   Textfile                          0006907082          1993-05-06 ISPF
** ****************************** bottom of list ******************************









Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right :s
```

Meaning of the column headings:

Field	Meaning
`Document`	Document name.
`Int.Number`	Internal number used by Natural ISPF to find the document.
`Created`	Document creation date in format *`YY-MM-DD`*.
File	Name of the Con-nect file in which the document is stored.

Example: INFORMATION

The following screen illustrates the information screen for a Con-nect document invoked using the command:

```
INFO DOC MBE(TEXTFILE)
```

The screen shows information on document `TEXTFILE` from cabinet `MBE`:

```
----------------------- Information DOC:MBE(TEXTFILE) -------------------------
 Command ===>


    Document  ===> TEXTFILE
    Folder    ===> Work___________   File     ISPF___________
    Subject   ===> To test ISPF14_____________________________________________
                   ___________________________________________________________
 ____________________________________________________________
                   ___________________________________________________________
    Keywords  ===> TEST___________  ISPF___________  ________________
                   ________________ ________________ ________________
    Private   ===> _
    Exp-date  ===> 1994/05/06        Archive  _

    Attached  ===>                   Enclos
    Created   ===> 1993/05/06  11:12 By       MBE
    Modified  ===> 1993/05/06  14:48          MBE
    Mailed    ===>
    Forward   ===>
    Reply     ===>

 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
       Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right :s
```

The information presented in this screen is the same as documented in the *Con-nect User's Guide*. You can modify the highlighted fields in this screen, for example, rename the document, edit the description in the Subject field or assign new keywords.

Line Commands

Select a document from a list by entering a line command in the input field preceding the document name and pressing ENTER.

Line Command	Corresponding Function Command
B	BROWSE
D	DELETE
E	EDIT
I	INFORMATION
PR	PRINT

When selecting a document with the PR line command, a window prompts you for the printer name and you can specify whether the document is to be printed formatted or not.

Line commands can also be used as valid abbreviations of function commands entered in the command line of any system screen.

Local Commands

In Edit Mode:

If you display a Con-nect document in Editor format in `EDIT` mode, you can issue the following local command from the Editor command line in addition to Editor commands:

Command	Meaning
`IMPORT`	Imports a PC file or another Con-nect document into the document (see section ***Useful Features***).

In List Mode:

If you display lists of Con-nect documents in Editor format, you can issue the local commands `ALL`, `LAYOUT`, `RELIST` and `SORT` in addition to Editor scroll commands. For detailed information, see the subsections in section ***Useful Features***.

User Workpool

The user workpool is an internal pool from which you can select output files of Natural programs, Natural utilities or job output for further maintenance.

The following objects are written to the user workpool;

- The output of macro objects as a result of the `RUN` command;
- The generated code of any object that includes inline macros or the `INCLUDE-MACRO` statement, as a result of the `STOW`, `CATALOG` or `RUN` command (Natural objects), or the `SUBMIT` command (other sources);
- The output of any Natural program that defines the workpool as a printer and which includes the `WRITE`, `PRINT` and/or `DISPLAY` statements referring to that printer;
- The output from Natural ISPF and Natural utilities by specifying the workpool as print destination;
- A command script executed by the `PLAY` command. Also, if a command script is interrupted by the `PAUSE` command or error, the command lines not yet executed are kept in the workpool and can be modified;
- Command sequences including (error) messages recorded by the `RECORD` session command;
- A Zap generated with the `GENERATE` or `SAVE` command after a `CSECT` has been edited in Natural ISPF.

An example of objects that use the macro facility is contained in the section *Macro Facility* in the *Natural ISPF Programmer's Guide*. An example of a Natural program that uses the workpool as output destination is contained in the subsection ***Write-To-Workpool Feature***. An example of

specifying the workpool as printer in a Natural utility is contained in the example. For details on the other instances, see the description of the PLAY and RECORD commands in the section *Useful Features*, and the description of CSECT handling in the subsection ***Load Modules and CSECTs*** in the section *z/OS Objects*.

The workpool holds only one entry for each generated Natural program and report. If a program with output in the workpool is run, stowed or cataloged again, the existing output is replaced according to object type. You can browse, edit, save and delete output in the workpool.

Note: Workpool files are intermediate files only. If you wish to keep source that was generated in the workpool, it is strongly recommended that you store it as another object elsewhere in Natural ISPF (see the subsection ***Saving Output***).

To enter the user workpool, select the WORKPOOL option from the Natural ISPF Main Menu. This displays the Workpool Entry Panel:

```
------------------------- WORKPOOL - ENTRY PANEL ----------------------------
 COMMAND ===>


    Program       ===>
    Type          ===>                 ( MACRO, REPORT, ZAP etc.  )









 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
       Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Meaning of the input fields:

Field	Meaning
Program	Name of program or member whose output is to be maintained. Enter the asterisk wildcard (*) to list all output files or enter a prefix followed by an asterisk (*) to list all file names with that prefix (see also the **example of the LIST command**).
Type	Type of output. Leave blank to list all types according to the name selection criteria. Specify: MACRO - To list only output of programs that use the macro facility. PLAY - To list a command script executed or interrupted. RECORD - To list recorded command sequences. ZAP - To list Zaps generated after CSECT editing. REPORT - To list all other output files (Natural programs that use the workpool as printer destination).

To select an output file for maintenance specify its program name in the Program input field and enter a function command in the command line. Alternatively, you can issue a function command with object type O and member name from any system screen.

Function Commands

The following function commands are available for the workpool facility:

Command	Parameter Syntax
BROWSE	*output-name*
COPY	*output-name*, *object-type object-parameters*, REP
DELETE	*output-name*
EDIT	*output-name*
EXPORT	*output-name*, *destination*
LIST	..* TYPE=*t*
PLAY	*output-name*
PRINT	*output-name*, *printer-id*
SUBMIT	*output-name*, TARGET=*node-id*

These commands are described in detail in section *Command Reference*.

If you issue any of these commands from outside the workpool facility, you must specify object type O (output) in the command syntax before the object parameters.

You can display your edit and the workpool sessions in split-screen mode and immediately see the effect of any modifications on the output using the RUN, CATALOG, STOW or SUBMIT command from the edit session as appropriate.

Below are some examples of function commands using full command syntax.

Example: LIST

The following figure illustrates the result of the command:

```
LIST O * TYPE=MACRO
```

The list includes all output files that use the macro facility (macro objects and output of members that have inline macros):

```
LIST-OUT:/TYPE=MACRO ---------------------------- Row 0 of 1 - Columns 028 049
 COMMAND===>                                                   SCROLL===> CSR
   PROGRAM           TYPE     DATE     TIME    LINES
** ****************** top of list ******************
   MYPROG            MACRO    19980116 14:03      22
   ##SUBMIT          MACRO    19980116 14:05      14
** ***************** bottom of list ****************

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Meaning of the column headings:

Field	Meaning
PROGRAM	Name of the member that generated this output: *name*><: Name of macro object, or of Natural program that uses the workpool as printer destination. ##INLINE Output of Natural program that uses inline macros. ##SUBMIT Output of macro objects and non-Natural members as a result of the SUBMIT command.
TYPE	Type of output (e.g., MACRO, REPORT, PLAY, ...)

Field	Meaning
DATE	Date output was generated in the workpool.
TIME	Time output was generated in the workpool.
LINES	Number of lines in the output.

Line Commands

The following line commands are available from a list of output objects. Each line command is an abbreviation of the corresponding function command:

Line Command	Corresponding Function Command
B	BROWSE
CP	COPY
D	DELETE
E	EDIT
EX	EXPORT
PL	PLAY
PR	PRINT
SB	SUBMIT

Line commands can also be used as valid abbreviations of function commands entered in the command line of any system screen.

Example 1: Printing output from a Natural utility to the workpool

This example shows you how to write selected messages from the Natural utility SYSERR to the workpool.

In the Natural SYSERR Utility Menu, specify function code PR, an appropriate message type and the required application (in our example, SYSISPS1). Then specify the required message range to be printed:

```
17:28:33              ***** NATURAL SYSERR Utility *****              94-12-27
                                    - Menu -

                  Code  Function
                  ----  ----------------------------------------
                   AD   Add new messages
                   DE   Delete messages
                   DI   Display messages
                   MO   Modify messages
                   PR   Print messages
                   SC   Scan in messages
                   SE   Select messages from a list
                   TR   Translate messages into another language
                   ?    Help
                   .    Exit
                  ----  ----------------------------------------
           Code .. PR   Message type .... UL
                        Library ......... SYSISPS1
                        Message number .. 6800 - 6810
                        Language codes .. 1________
 Please enter code.
 Command ===>
 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
       Help        Exit                                                  Canc
```

Press ENTER to display the printer specifications.

You must specify WORKPOOL as printer:

```
17:28:33              ***** NATURAL SYSERR Utility *****              94-12-27
                                   - Menu -
                    +------Print User Defined Error Texts------+
                    !                                          !
                    !   Library .......... SYSISPS1            !
                    !   Language code .... 1                   !
                    !                                          !
                    !   Long texts, too .. Y                   !
                    !   Error number ..... 6800 -  6810        !
                    !   Lines per page ... 60_                 !
                    !   Left margin ...... 10                  !
                    !   Top margin ....... 0_                  !
                    !   Bottom margin .... 0_                  !
                    !   Printer .......... WORKPOOL            !
                    !                                          !
                    +------------------------------------------+
           Code .. PR   Message type .... UL
                        Library ......... SYSISPS1
                        Message number .. 6800 - 6810
                        Language codes .. 1________

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

Press ENTER to perform the function. If you entered UL as Message type, the specified messages are written to the workpool, including long texts.

You can view the messages by selecting the WORKPOOL option from the Natural ISPF Main Menu:

```
BROWSE-OUT:PPRTUSR/TYPE=REPORT-1 ------------------------------ Columns 001 076
 COMMAND===>                                                    SCROLL===> CSR
** ********************************* top of list ********************************
            ---------------------------------------------------------------------
            SYSISPS16800 Invalid command
            ---------------------------------------------------------------------

            Text:  The command that was entered was not a valid N-ISPF command.

            Expl:  .

            Actn:  .

            ---------------------------------------------------------------------
            SYSISPS16801 Invalid parameter
            ---------------------------------------------------------------------

            Text:  Invalid parameter

            Expl:  The parameter that was entered was not a valid N-ISPF parame

            Actn:  .
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Note: You can proceed as in the above example with all Natural utilities which prompt for a printer name. Output of other Natural utilities and Natural system commands can also be routed to the workpool by means of the Natural terminal command `%H#WORKPOOL` (see Natural *Terminal Commands* documentation).

Example 2: Printing output from any Natural ISPF screen to the workpool

From any Natural ISPF screen, enter the command:

```
NAT LIST P MYPROG EXPAND C *
```

In the command line of the resulting screen, enter the command:

```
%H#WORKPOOL
```

Then press PF2 to print the expanded program list and return to Natural ISPF by pressing PF3. You can also handle the expanded list in the workpool as appropriate.

Example 3: Printing Predict object lists to the workpool

Within Software AG's Interactive Data Dictionary Predict, lists of objects can usually be shown in one of the display modes `SELECT`, `LIST` and `DISPLAY`. If you choose either the `LIST` or the `DISPLAY` mode and enter the terminal command `%H#WORKPOOL`, as reply to the `MORE` prompt, the complete list of Predict objects is written to the workpool where it can be further processed.

Saving Output

When browsing or editing output in the workpool, you can save the output currently displayed as another Natural ISPF object using the Editor command `CREATE` from the Editor command line.

You must mark the block of lines you wish to copy to the target source with two Editor line commands `CC`, and then issue the `CREATE` command in the format:

```
CREATE object-type object-parameters
```

If you omit the *`object-type`* and/or the *`object-parameters`*, prompt windows help you make a valid selection (note that the command format and prompt windows are the same as used for the `COPY` function command).

Example: CREATE

The command displayed in the Editor command line of the following screen creates PDS member `MYJOB` in the library `MY.ONLY.SOURCE` using all four lines of the displayed output (marked with two `CC` Editor line commands):

```
EDIT-OUT:EXJCL/TYPE=MACRO------------------------------------ columns 001 072
 COMMAND===> CREATE P MY.ONLY.SOURCE(MYJOB                      SCROLL===> CSR
***** ****************************** top of data ******************************
CC001 //* FR=//* ,
00002 //* SV #VOL = COM811
00003 //JWOTP12  JOB  JWO,CLASS=1,MSGCLASS=X,REGION=2500K
CC004 //SCAN     EXEC TAPESCAN,TAPE=COM811
***** ***************************** bottom of data ***************************









Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Alternatively, you can copy output from the workpool with the:

- `CP` line command from a list of workpool entries or
- `COPY` function command from any system screen in the format:

```
COPY O name,object-type object parameters,REP
```

Recovery Files

If you set `RECOVERY ON` in your Editor profile, recovery files are written while you are editing. Recovery files can be retrieved after an abnormal termination. The frequency at which recovery files are written can also be specified in your Editor profile. For example, a value of 10 means that a recovery file is written every 10th line you modify (see the subsection ***Editor Profile*** in section *Profile Maintenance*).

Recovery files are separate objects in Natural ISPF with object type `REC`, though they have no Entry Panel. If there is an abnormal termination while you are editing, the next time you log on to Natural ISPF, you are notified with the message: You have lost files. Enter `RECOVER` command to recover. If you issue the `RECOVER` command (an implicit `LIST REC`), you are presented with a list of recoverable files (see **the example of the `LIST` command**). Alternatively, you can issue another function command with the object-type parameter `REC`.

Authorized users such as the system administrator can maintain recovery files of all users. Recovery files of all users have object type `BPR`, and available functions are `LIST` and `DELETE` (see also the *Natural ISPF Administration Guide*).

Function Commands

The following function commands are available for recovery files:

Command	Object Parameter Syntax
`DELETE`	`REC` *`member-name`*
`EDIT`	`REC` *`member-name`*
`LIST`	`REC`

where *`member-name`* is the name of the edited object.

Example LIST

The following display is a result of the `LIST REC` command after an abnormal termination during an editing session with a Natural program and a PDS member:

```
LIST-REC: ---------------------------------------- Row 0 of 2 - Columns 030 076
 COMMAND===>                                                SCROLL===> CSR
   TYPE BLOCK DATE     TIME              OBJECT-IDENTIFICATION
** ******************************** top of list ******************************
   NAT  00512 19981228 10:12:54          BRY(ISPJ---U)
   PDS  00723 19981228 10:14:29          BRY.COMN.SOURCE(ISPRULES)
** ****************************** bottom of list ******************************

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Meaning of the column headings:

Column	Meaning
TYPE	Object type.
BLOCK	Internal block number.
DATE	Date recovery file was written.
TIME	Time recovery file was written.
OBJECT-IDENTIFICATION	Library and (object name).

Use a line command to select any recovery file from the list for further handling (see the subsection *Line Commands*).

Line Commands

The following line commands are available for a list of recovery files. Each line command is an abbreviation of a function command.

Line Command	Corresponding Function Command
D	DELETE
E	EDIT

Line commands can also be used as valid abbreviations of function commands entered in the command line of any system screen.

A recovery file can be edited and saved as any normal edited member. After you have edited and saved a recovery file, the END command (usually assigned to PF3) returns you to the list of recovery files.

Once a recovery file has been edited or deleted, it is erased from the list of recovery files, regardless of whether the recovered edit session is terminated by the END or CANCEL command.

Container Files

Natural ISPF allows you to access the Incore database (IDB) container file. Container files are usually created by your application programs using IDB. Select the CONTAINER option from the Natural ISPF Main Menu. The IDB Container Entry Panel appears:

```
------------------------- IDB CONTAINER ENTRY PANEL --------------------------
 COMMAND ===>


    Type          ===>
    Group         ===>
    Name          ===>
```

The meaning of the input fields is explained in the following table:

Field	Meaning

`Type`
`Group`
`Name`
	These 3 fields identify a container file.
	
	If any of the fields contain an asterisk (*), the `LIST` function is invoked, if not, the container file is browsed.

Function Commands

The available function commands are:

Command	Parameter Syntax
`BROWSE`	`TYPE=`*`type`* `GROUP=`*`group`* `NAME=`*`name`*
`DELETE`	`TYPE=`*`type`* `GROUP=`*`group`* `NAME=`*`name`*
`LIST`	`TYPE=`*`type`* `GROUP=`*`group`* `NAME=`*`name`*

Note: If you issue any of the above function commands from outside the container files facility, you must specify the *`object-type`* parameter `CTN` before the object parameters.

Line Commands

The following line commands are available:

Line Command	Corresponding Function Command
`B`	`BROWSE`
`D`	`DELETE`

Example:

For example, if you enter an asterisk (*) in the `Type` field and press ENTER, a list of container files appears in the following format:

```
LIST-CTN:/TYPE=*/GROUP=/NAME=* ----------------- Row 0 of 50 - Columns 040 076
 COMMAND===>                                                  SCROLL===> CSR
   Type     Group    Name                 Action   User     Date     Time
** ******************************** top of list ******************************
   PERSONNE CV       11100114                      MZC      19981201 11:02:21.6
   PERSONNE CV       11500327                      JWO      19970705 19:21:11.6
   PERSONNE CV       40008001                      MZC      19970714 12:26:48.7
   REPORT   PERSON   ADDRESSES                     JWO      19961120 15:21:01.6
   REPORT   PERSOX   ADDRESSES                     JWO      19981108 18:12:21.1
   SAETZE   TEST     WOHL                          MZC      19981201 11:13:10.0
   TEXT     CARS     JWO                           JWO      19970715 15:54:52.2
   TEXT     COMMENT  AL JARREAU                    MZC      19970624 12:30:06.7
   TEXT     COMMENT  BEACH BOYS                    JWO      19961124 08:23:10.0
   TEXT     COMMENT  BEATLES                       JWO      19970720 17:21:43.8
   TEXT     COMMENT  COMEDIAN HARMONISTS           MZC      19981212 22:11:31.4
   TEXT     COMMENT  DIE FANTASTISCHEN 4           MZC      19971209 11:44:11.1
B  TEXT     COMMENT  ERIC BURDON          *Browsed MZC      19981212 22:21:43.9
   TEXT     COMMENT  F.SINATRA & C.BASIE           MZC      19980124 15:27:58.9
   TEXT     COMMENT  JETHRO TULL                   MZC      19971209 12:32:04.2
   TEXT     COMMENT  JIMI HENDRIX                  JWO      19970406 19:05:28.0
   TEXT     COMMENT  MARBLES                       JWO      19970706 14:43:15.8
   TEXT     COMMENT  MONKEYS                       JWO      19970624 14:23:25.0
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

If, for example, you select the file `ERIC-BURDON` for browsing with the `B` line command, the contents of the file are displayed as follows:

```
BROWSE-CTN:/TYPE=TEXT/GROUP=COMMENT/NAME=ERIC-BURDON ---------- Columns 001 050
 COMMAND===>                                                   SCROLL===> PAGE
=cols> ----+----1----+----2----+----3----+----4----+----5
       LINE
****** ******************* top of data ******************
000001 An English singer who has been in  the  music  biz
000002 for more than thirty years, in several formations,
000003 the  most  successful  of  which   was   certainly
000004 "Eric Burdon and War".
000005 Some people say  he  is  the  blackest  among  all
000006 these white blues singers !
****** ***************** bottom of data *****************



Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

6 Profile Maintenance

- Profile Maintenance Overview 150
- PF-Key Definition 151
- Library Definition 153
- Magic-Character Definition 155
- Editor Profile 157
- User Defaults 163
- BS2000 Defaults 166
- Natural Defaults 171
- Editor Color Definition 173

Profile Maintenance Overview

With the profile maintenance facility, you can modify your user profile according to your personal requirements when working with Natural ISPF.

A special feature allows you to specify whether any modification is to be valid for the current session only, or whether it is to be saved for the next time you log on to Natural ISPF: when you press PF3 to leave a specific profile screen, a window is opened in which Natural ISPF asks you to specify `Y` to save the definition or `N` to retain it for the current session only (`Y` is the default). Press ENTER to close the window.

To enter the profile maintenance facility

- Select the `PROFILE` option from the Natural ISPF Main Menu. Alternatively, issue the `PROFILES` session command from any system screen.

  The Profiles Menu appears:

```
------------------------------ PROFILES MENU ---------------------------------
 OPTION  ===>
                                                                 Userid   BRY
                                                              Time     15:40:50
   1    KEYS       - Display and update pf keys definition    Terminal DAEFTC30
    2    LIBRARIES  - Libraries definition                        Library  BRY
    3    CHARS      - Magic chars definition                      Node     148
    4    EDITOR     - Editor profile
    5    DEFAULTS   - User parameters
    6    BS2 - 1    - BS2000 General defaults
    7    BS2 - 2    - BS2000 Submit  defaults
    8    NATURAL    - Natural defaults
    9    COLOURS    - Editor colour definition









Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

  This menu offers the following options:

Option	Meaning
KEYS	Assign commands to PF keys.
LIBRARIES	Define abbreviations for library names.
CHARS	Define magic characters for commands.
EDITOR	Modify your Editor profile.
DEFAULTS	Maintain user defaults.
BS2 - 1	Maintain BS2000-specific user defaults.
BS2 - 2	Define default parameters for BS2000 jobs to be submitted.
NATURAL	Define default parameters for processing Natural objects.
COLORS	Define colors for Editor.

These options are described in detail in the following subsections.

Note: The BS2000 options are offered only at BS2000 sites, or if the BS2000 subsystem has been enabled by your System Administrator.

PF-Key Definition

To display and modify your PF-key assignments

- Select option 1 (KEYS) from the Profiles Menu. Alternatively, you can issue the KEYS session command from any system screen.

  The Keys Update screen appears with a list of PF keys and their current assignments:

```
--------------------------------KEYS-UPDATE-----------------------------------
COMMAND ==>

    PF01  Help                          PF13  Help
    PF02  Split                         PF14  Split
    PF03  End                           PF15  End
    PF04  Suspend                       PF16  Suspend
    PF05  Rfind                         PF17  Rfind
    PF06  Rchange                       PF18  Rchange
    PF07  Up                            PF19  Up
    PF08  Down                          PF20  Down
    PF09  Swap                          PF21  Swap
    PF10  Left                          PF22  Left
    PF11  Right                         PF23  Right
    PF12  Cursor                        PF24  Cursor

    SHOW KEYS ON SCREEN OPTION  ON
    KEYS PORTION TO DISPLAY     FIRST




Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      HELP  SPLIT END   SUSPE RFIND RCHAN UP    DOWN  SWAP  LEFT  RIGHT CURSO
```

You can modify your PF-key assignments by overtyping any command with one or more valid Natural ISPF commands. If you assign more than one command to a PF key, they must be separated with the command delimiter specified in your user defaults. Pressing the associated PF key from any system screen has the same effect as entering the command(s) in the command line.

An additional feature enables you to assign Editor line commands to PF keys. When assigning Editor line commands to PF keys, you must precede them with a colon (:) . For example, specifying `:I` to PF13 means that whenever you are in an edit session and press PF13, a line is inserted at the cursor position as if you had used the `I` line command.

Some useful command sequences assigned to PF keys are:

```
SUSPEND;E :C
SUBMIT;FOLLOW
:A;COPY
```

The PF key assignment screen also allows you to specify the following:

- In the field labelled `SHOW KEYS ON SCREEN OPTION`, you can specify `ON` to display PF keys with their associated direct commands in the last two lines of any system screen. Specify `OFF` to suppress PF-key display;

- If you opt to display PF keys on system screens, you can specify which PF keys are to be displayed in the field labelled `KEYS PORTION TO DISPLAY`. Specify `FIRST` to display PF1 to PF12; specify `LAST` to display PF13 to PF24. You can use the `FLIP` direct command in any system screen to change from `FIRST` to `LAST` or vice versa.

Having entered your PF-key definition, press PF3 to open the confirmation window. Enter `Y` to save the PF-key definition or `N` to retain the definition for the current session only (`Y` is the default). Press ENTER to return to the Profiles Menu.

You can also use the `KEYS` session command with parameters to assign a command sequence to a PF key, without entering the profile facility. The command format is:

```
KEYS n <string>
```

where *n* is the number of the PF key and *string* the command sequence to be assigned.

Important: When assigning a command sequence consisting of two or more commands to a PF key using the KEYS command syntax, you must enter the command delimiter twice. For example, the command `KEYS 13 SUBMIT;;FOLLOW` assigns the command sequence `SUBMIT;FOLLOW` to PF13. See also the description of the `KEYS` **session command** in the section *Command Reference*.

Library Definition

This function does not apply to Natural libraries.

To define a two-character short name as an alias for any library

- Select option 2 (`LIBRARIES`) on the Profiles Menu. Alternatively, you can access this function directly from any system screen by issuing the `SHORTLIB` session command.

  The following screen appears:

```
------------------------------ SHORT LIBRARIES --------------------- Row 1 of 1
 COMMAND ===>

     Short Name    Data Set Name                                      Volser
     ----------    ---------------------------------------------      ------
     RW            RW.COMN.SOURCE
     __
     __
     __
     __
     __
     __
     __
     __
     __
     __
     __
     __
     __
     __
 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
       Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Meaning of the input fields according to column headings:

Column	Meaning
`Short Name`	Enter a two-character alias
`Data Set Name`	Enter the full library name
`Volser`	Specify the volume serial number if the dataset is not cataloged (at BS2000 sites, this field is not relevant for LMS-type libraries)

You can define up to 126 short names. You can scroll the list with the `UP` and `DOWN` commands.

Having defined short names, press PF3 to open the confirmation window. Enter `Y` to save the short names or `N` to retain the definitions for the current session only (`Y` is the default). Press ENTER to return to the Profiles Menu. You can use the short library name instead of the full name anywhere in Natural ISPF (in dataset selection fields and as object parameter in function command syntax).

Note: The library abbreviations specified here override the abbreviations for the same libraries defined by the system administrator.

To list available short IDs, enter the asterisk wildcard (*) in the Data Set Name field of the appropriate object Entry Panel and press ENTER. Alternatively, you can issue one of the following commands:

```
LIST DS *         (for z/OS environments)
```

```
BROWSE FIL *      (for z/VSE environments)
```

```
LIST LMS *(*)     (for BS2000 environments)
```

from any system screen. This opens a window with the user short IDs. To list system-wide short IDs, enter `G` in the window and press ENTER.

Magic-Character Definition

You can abbreviate any string used in Natural ISPF commands by assigning it to a magic character, usually a special character. If you enter this special character in the command line of any Natural ISPF screen, the effect is the same as if you had entered the associated string.

This feature makes entering long command sequences and multiple command input much more comfortable, for example when switching sessions, accessing other system functions or issuing operator commands. For example, if you assign the string `SUSPEND`; to the exclamation mark (!), the command input

```
!E NAT name
```

automatically starts a new Natural ISPF editing session with the specified Natural object. Or if you assign the stroke (/) to the string `OPERATOR_`, the command input:

```
/C object=name
```

cancels the specified object from the operating system. Blanks required at the end of a string assigned to a magic character are represented by the underscore wildcard (_).

To display the magic-character definition screen

- Select option 3 (`CHARS`) from the Profiles Menu. Alternatively, you can issue the `CHARPROF` session command from any system screen.

  The Magic-Character definition screen appears:

```
--------------------------------MAGIC-CHARS----------------------------------
COMMAND ===>

    Magic Char Position Substitution
    ---------- -------- -------------------------
    !                   SUSPEND;
    =          F        RETURN;
    /                   OPERATOR_
    #                   NAT_
    %                   MY.COMN
    +                   MYFILE
    >                   ;NEWNAME_









Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Meaning of the input fields according to column heading:

Column	Meaning
Magic Char	Enter the character you wish to use as an abbreviation for the string. It is recommended that you use special (non-alphanumeric) characters.
Position	Specify F if the string is to be substituted only if the magic character is in the first position of the command sequence. Leave blank if the character is to be substituted regardless of position.
Substitution	Enter the string to be substituted when the magic character is typed in the command line (use the underscore wildcard (_) for blanks in the string).

You can assign any number of strings to special characters. Press PF3 to open the confirmation window. Enter Y to save the magic characters or N to retain them for the current session only (Y is the default). Press ENTER to return to the Profiles Menu.

Below are some example command sequences using magic characters as defined in the above figure:

Command	Meaning
/S PROC01	OPERATOR S PROC01
!E PDS %.SOURCE	SUSPEND;EDIT PDS MY.COMN.SOURCE
E +	EDIT MYFILE

Note: Like any command string, magic characters can be assigned to PF keys.

Editor Profile

The Natural ISPF Editor provides a default editing profile which you can display by issuing the Editor PROFILE command from the edit screen. You can modify Editor profile settings for the current edit session using appropriate Editor commands or the Editor SET command.

To change your Editor profile default settings

- Use the PROFILE facility from the Natural ISPF Main Menu. Select option 4 (EDITOR) from the Profiles Menu, or issue the EDITPROF session command from any system screen.

  Your Editor Profile appears with its current settings:

```
------------------------------- EDIT PROFILE ----------------------------------
 COMMAND ===>


    PROFILE NAME  *default  ( Enter command SELECT to select another profile)
                            (                DELETE to delete this profile)
    SCROLL    CSR           ( Default scroll )
    CAPS      OFF           ( Upper case translation (ON/OFF/PGM))
    HEX       OFF           ( Hexadecimal mode )
    NULLS     ON            ( Fill end of line with nulls )
    RECOVERY  ON   EACH 7   ( Create recovery file each # updates )
    LOG       ON            ( Retain activity log (allow UNDO )
    AUTOSAVE  OFF           ( Automatic SAVE with END command )
    AUTOREN   ON            ( Automatic renumbering )
    PROTECT   OFF           ( Protect prefix area OFF/ON/INS )
    PREFIX    ON            ( Display prefix area ON/OFF )
    ADVANCE   ON            ( Automatic advance OFF/ON/PAGE )
    TABS      OFF  CHAR     ( Use tabs, optionally with specified character )
    TABS POS   9 15 39 71
   ESCAPE    OFF  CHAR     ( Escape to line command with specified character )
    LIMIT                   ( Limit for Find commands )

 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
       Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Meaning of the input fields:

Field	Meaning
PROFILE NAME	Name of the profile being edited, if only 1 unnamed profile exists, its name is *default. For further information, see the subsection ***Assigning Multiple Editor Profiles***.
SCROLL	Default scroll amount displayed in the SCROLL field of the Editor screen. Determines scroll value when using unqualified UP and DOWN commands (usually assigned to PF7 and PF8), and LEFT and RIGHT commands (usually assigned to PF10 and PF11). Possible options: CSR Scroll to cursor position.
DATA	Scroll by PAGE minus one line or column.
HALF	Scroll by half a page length or width. MAX Scroll by maximum amount (top, bottom, first or last column) .*n* Scroll by n number of lines or columns. PAGE Scroll by page length or width.
CAPS	Specifies upper case or mixed case translation. Possible options: OFF Leave as entered (mixed case). ON Translate to upper case. PGM Translates program data to upper case, but leaves comments in mixed case.
HEX	Specifies hexadecimal mode of data display. Possible options: OFF Alphanumeric mode only. ON Display data in hexadecimal mode.
NULLS	Fills end of line with nulls. Possible options: OFF Fill with blank characters. ON Fill with nulls.
RECOVERY	Activates recovery facility when editing files. Possible options: OFF No Recovery possible. ON Activate recovery.
EACH	Specifies number of updates to recovery checkpoint. For example, 5 means a recovery file is written every 5th modified line.
LOG	Activates log file. When ON, you can use the Editor command UNDO to back out all modifications since the last time you pressed ENTER (see the UNDO **command** in the section *Editor*). Possible options: OFF Deactivate log file. ON Activate log file.

Field	Meaning
AUTOSAVE	Specifies whether automatic SAVE command is performed when you issue an END command (usually assigned to PF3). Possible options: OFF No SAVE. ON Automatic SAVE is performed if data was changed.
AUTOREN	(Not relevant for editing BS2000 files.) Automatic renumbering when inserting/moving/copying lines. Possible options: OFF Automatic renumbering disabled. ON Automatic renumbering enabled.
PROTECT	Protects prefix area. Possible options: INS Prefix area of inserted lines protected. OFF Prefix area not protected. ON Prefix area protected.
PREFIX	Specifies whether prefix area is displayed or not. Possible options: OFF Prefix area not displayed. When OFF, line commands can be entered in the data area, preceded by the escape character. ON Prefix area displayed.
ADVANCE	Controls movement of cursor or data when only ENTER is pressed. Possible options: OFF No advance. ON Cursor is placed in next line. PAGE Data is advanced by amount in SCROLL field.
TABS	Sets tabulation. Possible options: OFF Set tabs off. ON Set tabs on.
CHAR	Defines the logical tabulation character (usually a special character, for example: %). When left blank, physical tabulation is in effect, if activated.
TABS POS	Defines columns for tab positions. Up to ten tab columns can be specified.
ESCAPE	Escape character to precede line command if line command is issued from the data area. Possible options: OFF Line command escape character function off. ON Character is taken as line command escape.
CHAR	Special character to be used as escape character for line commands if issued from the data area

Field	Meaning
LIMIT	Defines a limit for the FIND command. FIND command searches only the number of records entered here.

Having modified your Editor profile, press ENTER to open the confirmation window. Enter Y to save the profile or N to retain it for the current session only (Y is the default). Press ENTER to return to the Profiles Menu.

Assigning Multiple Editor Profiles

Natural ISPF allows you to define multiple Editor profiles for each user or a group. This profile is called *default profile and is also used in BROWSE and LIST sessions. In EDIT sessions the default profile is used if no other profile is found.

Profiles are identified with 8-byte names and when you open an EDIT session, a list of profile names (called a request list) is created. The contents of this list depend on the Natural ISPF object type to be edited and are described in more detail below. If none of the named profiles are found, (remember that a search is usually performed for a user and all groups) the default profile is used.

The following table describes how the request list is built for each object type which can be edited in Natural ISPF. (Member: EDASSIGN)

Object Type	1	2	3	4	...	last
Natural	Object type	Library name				N-ISPF object
Workpool	Type					N-ISPF object
Error texts	*constant* TEXT	*constant* ERROR				
PDS members	DSN *qua.n*	DSN *qua.n*-1	DSN *qua.n*-2	DSN *qua*. 1		N-ISPF object
z/OS datasets	DSN *qua.n*	DSN *qua.n*-1	DSN *qua.n*-2	DSN *qua*. 1		N-ISPF object
z/VSE Librarian members	Member type	Sublib	DSN *qua.n*	DSN *qua*. 1		N-ISPF object
z/VSE files	DSN *qua.n*	DSN *qua.n*-1	DSN *qua.n*-2	DSN *qua*. 1		N-ISPF object
CA Panvalet members	Language	DSN *qua.n*	DSN *qua.n*-1	DSN *qua*. 1		N-ISPF object
CA Librarian members	Language	DSN *qua.n*	DSN *qua.n*-1	DSN *qua*. 1		N-ISPF object
BS2000 LMS members	Element type	DSN *qua.n*	DSN *qua.n*-1	DSN *qua*. 1		N-ISPF object
BS2000 files	DSN *qua.n*	DSN *qua.n*-1	DSN *qua.n*-2	DSN *qua*. 1	DS Org	N-ISPF object
BS2000 job variables	JV *qua.n*	JV *qua.n*-1	JV *qua.n*-2	JV *qua*. 1		N-ISPF object
Menu						N-ISPF object

where *qua* stands for qualifier, which is part of a z/OS dataset name.

Example:

The following commands create the lists of profile names below:

Command	Profile name list			
ED NAT MYAPPL(PROG1) TYPE=C	COPYCODE	MYAPPL	NATURAL	
ED OUT PROG1 TYPE=REPORT	REPORT	WORKPOOL		
ED ERR MYAPPL(1011)	TEXT	ERROR		
ED PDS TSIS.COMN.ASM(PROG1)	ASM	COMN	TSIS	PDS
ED DS TSIS.COMN.OUTPUT	OUTPUT	COMN	TSIS	DS
ED MEM MYFILE.TEST(PROG1.JOB)	JOB	TEST	MYFILE	MEMBER
ED FIL TSIS.COMN.OUTPUT	OUTPUT	COMN	TSIS	FILE
ED PAN PANT.LIB1(PROG1) LANG=COBOL	COBOL	LIB1	PANT	PAN
ED LIB LIBR.MASTER(PROG1) LANG=ASM	ASM	LIBR	MASTER	LIB
ED LMS NATISPF.TESTLIB(PROG1)S	S	TESTLIB	NATISPF	LMS
ED BF NATISPF.TESTFILE	TESTFILE	NATISPF	SAM	BF
ED JV ASF.TEST	TEST	ASF	JV	
ED MENU MAIN	MENU			

Whenever a non-default profile is selected for an edit session, a message is displayed indicating the name of the selected profile. If it was taken from another user (group), the user ID is also displayed.

With this mechanism it is very easy to define special Editor profiles, for example, for Assembler programs (including the correct tab setting). If all Assembler programs are stored in datasets with last qualifier `ASM`, the administrator can define an Editor profile named `ASM` for user `*`. This profile is automatically used by all users editing a member in a dataset for `ASM` programs, unless they have defined a personal Editor profile named `ASM`.

Local Commands

To maintain the different profiles, 2 additional local commands are available:

Local Command	Meaning
DELETE	Delete the profile being edited.
SELECT	Select another existing profile or define a new profile.

Example: SELECT

If you issue the `SELECT` local command from the Edit Profile screen, the following window opens:

```
------------------------------- EDIT PROFILE ----------------------------------
COMMAND ===> sel
                        +------Select Profile-------+
                        !                           !
   PROFILE NAME  *defau ! Enter new name  ________  ! elect another profile)
                        ! or select:       1 of 4   ! elete this profile)
   SCROLL    CSR        !   Profile  User           !
   CAPS      OFF        ! _ *default                ! N/OFF/PGM))
   HEX       OFF        ! _ COMN                    !
   NULLS     ON         ! _ COPYCODE                ! ls )
   RECOVERY  ON   EACH  ! _ COBOL    *              !  # updates )
   LOG       ON         ! _                         ! w UNDO )
   AUTOSAVE  OFF        ! _                         ! ommand )
   AUTOREN   ON         ! _                         !
   PROTECT   OFF        ! _                         ! N/INS )
   PREFIX    ON         ! _                         ! F )
   ADVANCE   ON         ! _                         ! PAGE )
   TABS      OFF  CHAR  ! Entr-PF3--PF7--PF8--      !  specified character )
   TABS POS   9 15 39 7 ! Down End  Up   Down       !
   ESCAPE    OFF  CHAR  +---------------------------+ th specified character )
   LIMIT                   ( Limit for Find commands )

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

This window lists all Editor profiles accessible to this user. If a profile is stored for a group, the name of the group is also listed (as for the profile `COBOL` in this example).

You can scroll forward or backward through the profile list and can select one for editing.

To create a new editor profile

- Just enter the name of the profile in the appropriate field and press ENTER.

User Defaults

To set several user defaults that affect your working environment in Natural ISPF

- Select option 5 (`DEFAULTS`) from the Profiles Menu. You can also access this function directly with the `DEFPROF` session command from any system screen.

  The User Defaults definition screen appears:

```
------------------------------ USER DEFAULTS ---------------------------------
COMMAND ===>


   NODE              148      ( Default system node number )
   PRINTER         *          ( Default printer. '*' for prompt on print )
   FILE TYPE       NAT        ( Default file type in direct commands )
   DSNAME
                              ( Default DSname in direct commands )
   MACRO EXPAND    N          ( Expand MACRO in NATURAL programs )
   MACRO SMODE     S          ( SMODE for non NATURAL macros: S/R )
   TRACE           1          ( Interval (seconds) for tracing functions )
   BREAK                      ( Allow break after number of traces )
   VERSIONS        N          ( Save PDS/NAT/MEM last versions )
   CONFIRM         SHORT      ( Confirmation type: SHORT/LONG )
   COMMAND DEL.    ;          ( Delimiter between commands in command line )
   PARM DEL        ,          ( Delimiter between command parameters )
   COMMAND
                              ( Default command when entering NSPF )
   NO-RECOVER                 ( No automatic check for recovery files )
   PRINT VIA NOM              ( Use ENTIRE OUTPUT MANAGEMENT for printing )

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

All fields which are stored in the user profile are highlighted. Fields inherited from a user group (like `*`) or taken from predefined defaults are displayed in normal intensity and are not stored in the user profile. If a field is modified on the screen, it will be highlighted and stored in the user profile.

Note: With this function, the administrator can easily maintain several important default values like `NODE` or `PRINTER` in the user profile of user `*`. If these fields are not modified in the individual user profiles, they are not stored in individual profile records but are kept in one place only. If the node number must be changed for some reason, the administrator must change only one user ID and all others inherit the new value without having to change every single user profile.

Meaning of the input fields:

Field	Meaning
NODE	Your default Entire System Server node number. You can select another node at any time for the session by using the NODE *id* session command, where id is the new node number.
PRINTER	Default printer for Natural ISPF printouts. If you enter the asterisk wildcard (*) here, a window opens every time you use the PRINT function command, prompting you for the printer name. If you leave this field blank, the printer is selected according to the following hierarchy: ■ Printer specified in the PRINT command; ■ Printer specified in your user profile; ■ Printer specified in your user group profile; ■ Printer defined in your TP environment; ■ Printer assigned to Natural profile parameter PRINTER2.
FILE TYPE	Default object type to be addressed when issuing commands without the object type parameter. Any valid Natural ISPF object type is possible, for example: LIB - CA Librarian member LMS - LMS library element MEM - z/VSE member NAT - Natural object PAN - CA Panvalet member PDS - PDS member
DSNAME	Default dataset name to be addressed when issuing direct commands without the dataset name parameter.
MACRO EXPAND	Specifies whether Natural programs and other objects that may contain job control (for example, z/VSE or PDS members, BS2000 files or LMS elements) are checked for inline macros, which are expanded when using STOW, CAT, RUN and SUBMIT commands. Possible options: N - Check is not performed. Y - Check is performed. The Macro expansion facility can be activated or deactivated at any time using the MACRO ON/OFF session command.
MACRO SMODE	Mode to be assumed when non-Natural objects that use the macro facility are submitted. Possible options: R - Report mode

Field	Meaning
	S - Structured mode The specification made here overrides the system default defined by the system administrator. If you do wish to set another mode, you must do so before starting your edit session.
TRACE	Time interval in seconds for the tracing facility. A value of zero (0) means TRACE OFF. Modifiable with the TRACE session command.
BREAK	Allow interruption of Natural ISPF activity after *n* trace intervals. Modifiable with the BREAK *nn* session command.
VERSIONS	Activates versioning function for PDS, z/VSE Librarian and Natural members. Possible options: N - Versioning not active. Y - Versioning active. Modifiable with the VERSIONS ON/OFF session command.
CONFIRM	Specifies mode of confirmation in confirmation windows when using certain function commands such as DELETE. Possible options: LONG Confirm by typing object name in confirmation window. SHORT Confirm by typing Y in confirmation window. You can deactivate and reactivate the confirmation feature for the current system screen using the CONFIRM OFF/ON session command (see also the description of the session command CONFIRM in the section *Command Reference*).
COMMAND DEL.	Delimiter to separate commands when issuing multiple commands in a single input operation.
PARM DEL.	Delimiter to separate object parameters from function parameters in function command syntax (for examples, see the PRINT and COPY function commands).
COMMAND	Default command which is always executed when the user starts Natural ISPF. If several commands are to be executed, a command script can be written and executed using the PLAY function command.
NO RECOVER	When Natural ISPF is started, it automatically checks whether checkpoint files exist for the user and, if they do, informs the user to enter the command RECOVER. This automatic check, which slows response time for a few seconds during each start of Natural ISPF, can be suppressed by entering Y in this field. Checkpoints are still written by Natural ISPF and with the command RECOVER you can always check manually whether checkpoints exist.
PRINT VIA NOM	To use the extended interface between Natural ISPF and Entire Output Management, enter Y. See also the description for the NOM PRINTER field (on the NSPF Parameter screen) in the section *System Configuration* of the *Natural ISPF Administration Guide*.

Having defined your user defaults, press PF3 to open the SAVE PROFILE confirmation window. Enter Y to save the definitions permanently or N to retain them for the current session only (Y is the default). Press ENTER to return to the Profiles Menu.

BS2000 Defaults

User defaults for BS2000 sites can be specified using two options on the Profiles Menu:

- General Defaults
- Defaults for job submission

These options are only accessible if the BS2000 subsystem is enabled.

General Defaults

To set several user defaults that affect your environment in Natural ISPF when working with BS2000 objects

- Select option 6 (BS2 - 1) on the Profiles Menu. You can also access this function directly by using the DEFBS2PROF direct command (shortest abbreviation is DEFB) from any system screen.

  The BS2000 User Defaults screen appears:

```
--------------------------- BS2000 USER DEFAULTS -------------------------
COMMAND ===>


 Default parameters for LMS element maintenance:
   LMS Element Type        S                       ( used for new elements )
   LMS Element Version     §                       ( used for new elements )
   LMS Storage Mode        V                       ( for future use        )
   Allow Override on SAVE  *                       ( * / YES / NO          )

 Default parameters for function DELETE-BS2FILE:
   ERASE Option                             ( * or blank/DATA/DESTROY/SPACE )
   ERASE Range (for FGGs)  *                ( * or A/B;  * will cause prompt)









Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
       Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right :s
```

The profile settings have the following meaning:

Field	Meaning
LMS Element Type	Specifies the default type of LMS element used when a new element is created without explicit specification of the element type. Possible options: D Text data, for example, program input H Compiler result J BS2000 job control M Assembler Macro P Protocol data S Source program X General data of any format **Note:** For normal development work, types P and H are not recommended as defaults, but they can be useful when copying large amounts of files into library elements of that type.

Field	Meaning
LMS Element Version	Specifies the default version name to be used when a new element is created without explicit specification of a version name.
LMS Storage Mode	For future use. Currently, Full Storage Mode (V) is in effect and cannot be modified.
Allow Override on SAVE	Controls the action taken by Natural ISPF when an existing LMS element or element version has been edited and either a SAVE command is issued, or an END command is issued and AUTOSAVE=ON is set in your Editor profile. Valid options are: * A window opens where you can specify a new version name under which to store the updated element (the previous version is kept), or to confirm that the previous version can be overridden. YES The element version is updated in place, which means that the previous version is overridden without warning. NO A window opens where you can specify a new version name under which to store the updated element. The previous version cannot be overridden. In Natural Batch mode, an attempt to update an existing element version will be rejected. A SAVE operation will then be accepted only if a NEWNAME or RENAME command was issued first. **Note:** Values ONLY, EXTEND, V and D, known from the corresponding LMS parameter when used in TIAM dialog operation, are currently not supported in Natural ISPF.
ERASE Option	This operand allows you to specify ERASE macro execution mode when executing function DELETE-BS2FILE. Possible options: * Prompt user every time deletion of a BS2000 file is requested. blank Default: erase both space and catalog entry. DATA Erase data only: keep allocated space and catalog entry. DESTROY Overwrite with binary zeroes. SPACE Like DATA, but release the allocated space (not for files on private disks).
ERASE Range	Generation files only: this operand corresponds to the POS operand of the BS2000 ERASE macro. Possible options: * Prompt user every time a deletion of a generation file is requested. A All files after the generation number specified in FILENAME are deleted. B All files before the generation number specified in FILENAME are deleted. **Note:** The specified generation number itself is not deleted.

Submit Defaults

To set several user defaults that take affect when you submit batch jobs from your Natural ISPF session

- Select option 7 (`BS2 - 2`) on the Profiles Menu. You can also access this function directly by using the direct command `DEFSUBPROF` (shortest abbreviation is `DEFS`) from any system screen.

  The following screen appears:

```
----------------- USER DEFAULTS FOR SUBMITTING BS2000 JOBS --------------------
COMMAND ===>


   USERID                                  ( if USERID or ACCOUNT is preset,
   ACCOUNT                                   the PASSWORD will be prompted  )
   JOBCLASS                                ( blank / full name of jobclass  )
   TIMELIMIT          STD                  ( * /STD/NTL / nnnnn (in seconds))
   JOB-PRIORITY                            ( blank / value from 1 to 9      )
   RUN-PRIORITY                            ( blank / value from 1 to 255    )
   MONITORING-JOBVAR

  Additional options:
   FLUSH              NO                   ( * / YES / NO                   )
   RERUN              NO                   ( * / YES / NO                   )
   ERASE              NO                   ( * / YES / NO                   )
   START              STD                  ( * / STD/SOON/IMM/WITHIN/AT etc.)

  Note: Any value set to * will cause all values to be prompted.



Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

The profile settings have the following meaning:

Field	Meaning
`USERID`	Enter a user ID valid in the target environment, to be assigned to the job being submitted. If this field is left blank, the job is submitted under the current (Natural) user ID. Otherwise the required password is prompted at submission time.
`ACCOUNT`	Enter the account number associated with the specified user ID, to be assigned to the job being submitted. Leave this field blank if job is to be submitted under the current (Natural) user ID.
`JOBCLASS`	Specify the name of the job class where the job being submitted is to be queued. If you omit this parameter or specify `*STD` (which would be equivalent), a default job class is assigned as defined by your BS2000 System Administrator.

Field	Meaning
TIMELIMIT	Specify the expected amount of CPU time for the job being submitted. Possible options: * Prompt for value at submission time. *nnnnn* The amount of CPU time in seconds. NTL No time limit is assigned. STD Default: time limit is assigned according to the job class specified.
JOB-PRIORITY	The priority to be assigned to the submitted job within its job class: specify a numeric value less than or equal to 9; the smaller the value, the higher the priority of the job in the job queue, compared to jobs of the same job class. This value affects only the time when the job will start; it does not affect the amount of time the task (once started) will stay active. If you omit this parameter, it assumes a default value preassigned to the job class, as defined by your BS2000 System Administrator.
RUN-PRIORITY	The priority to be assigned to the submitted job once it gets started. Specify a numeric value less than or equal to 255; the lower the value, the higher the priority of the job compared to other tasks when it gets started. If you omit this parameter, it assumes a default value preassigned to the job class, as defined by your BS2000 System Administrator.
MONITORING-JOBVAR	Specify the name of a monitor job variable if you want this job variable to be used for all jobs submitted out of Natural ISPF. For further information, refer to BS2000 system literature.
FLUSH	States whether job is to remain in the job queue, should a system crash or system shutdown occur before the job has started. Possible options: * Prompt for actual value at submission time. NO Default: retain job. YES Remove job from queue in that case.
RERUN	States whether the job is to be restarted if the job is active when the system goes down. Possible options: * Prompt for actual value at submission time. NO Default: no restart. YES Restart job.
ERASE	States if the ENTER file is to be erased after job completion. Possible options: * Prompt for actual value at submission time. NO Default: do not erase job file.

Field	Meaning
	YES Erase job file.
START	The Start Option, together with the parameters Start-Date and/or Start-Time, specifies a time (or a time range) for the submitted job to be started. Possible options: * Prompt for actual value at submission time. AT A Time and/or date AT-STREAM-STARTUP Job starts at start of scheduler. EARLIEST Do not start before specified date and/or time. IMMEDIATELY Start now. LATEST Do not start after specified date and/or time. SOON As soon as possible. STD Default option is assigned according to job class definitions. WITHIN Specifies a time range, starting from now (for example: within the next 2 hours). If the specified START option takes one of the values *, AT, WITHIN, EARLIEST or LATEST, a window opens at submission time, prompting you for values for parameters Start-Date and/or Start-Time. **Note:** Some of the above options may be restricted to certain job classes, according to site-specific job class definitions.

Natural Defaults

To set several defaults that affect your working environment in Natural ISPF

- Select option 8 (NATURAL) from the Profiles Menu. You can also access this function directly by using the NATDEF direct command from any system screen.
  - The Natural Defaults screen appears:

```
---------------------------- Natural Defaults -------------------------------
COMMAND ===>


   INIT LOGON              ( Initial logon when entering N-ISPF )
   MODE        REPORT      ( Default mode of natural programs STRUCT/REPORT )
   TYPE         P           ( Default type of natural program )
   EDITOR       N           ( Use ISPF Editor,not NATURAL program editor)









Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Meaning of the input fields:

Field	Meaning
INIT LOGON	Default Natural library name when logging on to Natural ISPF. Leave blank to select current logon library.
MODE	Default mode when editing Natural programs. Possible options: STRUCT or REPORT.
TYPE	Default type of Natural program. For possible types, see description for TYPE field (Natural objects) in the section ***Common Objects***.
EDITOR	You can disallow the Natural Program Editor. Possible options: N Natural Program Editor is still available. Y When you invoke the Natural Program Editor from the NEXT prompt, you are automatically transferred to Natural ISPF.

Editor Color Definition

This option enables you to define the colors to be used by the Natural ISPF browser, which is responsible for all `LIST`, `BROWSE` and `EDIT` sessions. The colors used in other screens cannot be modified with this profile option.

To define the colors to be used by the Natural ISPF browser

- Select option 9 (`COLOR`) from the Profiles Menu. You can also access this function directly by using the `COLPROF` direct command from any system screen.

  The Color Definition screen appears:

```
---------------------------- Colour Definition ------------------------------
 COMMAND ===>


  Command Line              TU

  Edit/List sessions                      ( Valid Colours )
    Intense Input           NE              BL   Blue
    Intense Output          TU              GR   Green
    Normal  Input           GR              NE   Neutral
    Normal  Output          TU              PI   Pink
                                            RE   Red
  Browse sessions                           TU   Turquoise
    Intense Input           NE              YE   Yellow
    Intense Output          TU
    Normal  Input           GR
    Normal  Output          TU





Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Meaning of the input fields:

Field	Meaning
`Command Line`	Here you can assign a color to all input fields in the command line, which are the command field and the scroll field.
`Edit/List sessions`	Here you can define how the following screen attributes are mapped into colors for Edit and List sessions: ■ Intense Input ■ Intense Output ■ Normal Input ■ Normal Output
`Browse sessions`	Here you can define how the following screen attributes are mapped into colors for Browse sessions: ■ Intense Input ■ Intense Output ■ Normal Input ■ Normal Output

7 Editor

- Editor Overview 176
- Starting the Editor 176
- Special Features 178
- Editor Commands 179
- Invoke the Online Help Facility 226
- Main Commands 229
- Line Commands 192

Editor Overview

The Natural ISPF Editor, Software AG's editing facility for Natural ISPF, is a variant of the Software AG Editor, specially adapted to the Natural ISPF environment.

This section provides an overview of Editor commands available in the Natural ISPF environment. For a detailed description of the Editor, see the Software AG Editor documentation.

The Editor provides ISPF-like functionality to display and/or edit Natural ISPF objects such as:

- Natural programs and error messages;
- PDS members and sequential files;
- z/VSE members and files;
- CA Panvalet members;
- CA Librarian members;
- BS2000 files;
- LMS library elements;
- Job SYSOUTs (browse mode only);
- Output in the user workpool;
- Lists of system objects (browse mode only).

Starting the Editor

The Natural ISPF screen from which you enter the Editor will depend on the type of object you wish to edit or display. For example, if you wish to edit a Natural member, you might enter the Editor from the Natural Object Entry Panel. The possibilities for entering the Editor are described under the appropriate section headings in this documentation.

If you specify a new object name for editing, you will see an edit screen similar to the following:

```
EDIT-NAT:NATLIB1(JOB1JCL)-Program->Struct-Free-30K------------ columns 001 072
 COMMAND===>                                                  SCROLL===> CSR
****** ****************************** top of data ******************************
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
****** **************************** bottom of data ****************************
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

The first line on the screen identifies the object to be edited according to the environment from which the Editor is called.

In the above example, `JOB1JCL` is the new member name and `NATLIB1` identifies the object library which is a Natural library.

The edit screen contains four input fields:

- You can enter Editor main commands in the field labelled `COMMAND` in the second screen line;
- You can specify default scrolling amounts in the field labelled `SCROLL` in the second screen line;
- You enter Editor line commands by overtyping the line numbers on the left of the screen;
- You can enter data in the data area to the right of the apostrophes.

  You may now start your edit session by entering data into the edit screen and using the edit commands described in the subsection ***Editor Commands***. When you press ENTER, the apostrophes on the left of the screen will be replaced by line numbers on used lines. Unused lines are automatically deleted.

Special Features

The Natural ISPF Editor provides some special features to make use of commands more comfortable. These features are described in the following subsections.

Cursor-Sensitive Command Selection

The Editor provides a feature that makes Editor command parameter input more comfortable. In any Editor session (`LIST`, `EDIT` or `BROWSE` a Natural ISPF object), Editor command input can consist of documentation command keyword input plus any word from the data displayed.

Any word can be selected by placing the cursor on it after typing `:C` in the command line. When you press ENTER, the `:C` directive is substituted by the selected string and the complete command is executed.

Example:

```
EDIT0-NAT:NSPF101(JOB1JCL)-Program->Struct-Free-30K------------ columns 001 072
 COMMAND===> EDIT :C                                            SCROLL===> CSR
****** ****************************** top of data ****************************
...
000010 MOVE A TO B
000020 INCLUDE MYPROG
...
```

In the above edit screen, type the command:

```
EDIT :C
```

in the command line, move the cursor to the first character of the string `MYPROG` and press ENTER. The command executed is

```
EDIT MYPROG
```

Local Command Input

The Editor also allows you to enter object-specific commands from the Editor command line, for example the `CHECK` command when editing a Natural program. These commands are known as local commands and are described for each object type in the appropriate section of this documentation, as well as in the section ***Command Reference***. If you enter an asterisk (*) in the command line and press ENTER, a window opens with a list of all local commands.

Command Redisplay

The Editor provides a command redisplay feature that retains the display of the Editor command last issued. Precede the command with an ampersand (&). After command execution, the command remains in the command line and remains in display every time you press ENTER until you clear the command line or overtype the displayed command.

Note: This command redisplay feature is available only for Editor commands (e.g. `CHANGE`, `FIND`, `UNDO`), not for local commands , session commands or function commands entered in the command line of an edit screen; for these commands, use the session command `LAST`.

The session command `LAST` is also available for Editor commands entered in the command line: the last ten commands consisting of more than one word and entered via the keyboard are displayed in a window and are selectable for editing and reexecution (see the subsection ***Session Commands - Description*** in the section *Command Reference*).

Editor Commands

The Editor provides two types of edit commands with which you can control your editing session:

- **Main Commands**
  entered in the command field on the second line of your screen. You can enter several commands in the same input operation, separated by a semi-colon ;n. For a full description of command syntax, see the Software AG Editor documentation.
- **Line Commands**
  entered in the command field of any line by overtyping the line number on the left of your screen. Line commands refer to the line on which they are entered or to a block of data delineated by line commands. You can also enter line commands in the main command line if you precede them with a colon (:). The cursor then marks the line to be addressed. Alternatively, you can enter line commands from the first column in the data area of any line if you precede it with the escape character (See your edit profile. For more information see the subsection ***Editor Profile*** in the section *Profile Maintenance*).

Your Edit Profile

Each user has edit profiles containing default settings of his edit environment. To see the current settings of your edit profile, use the Editor command `PROFILE`.

You can modify single settings in your edit profile using appropriate Editor commands. The new settings are valid for the remainder of the edit session or until you change them again using the appropriate Editor commands.

To modify multiple profile settings for the current session in one input operation

- Use the `SET` main command.

  You can maintain the default settings of your edit profile using the `PROFILE` option on the Natural ISPF Main Menu or by using the `EDITPROF` session command from any system screen.

Below is an example of an edit profile, followed by an explanation of profile parameters.

```
EDITNAT:NSPF101(JOB1JCL)-Program->Struct-Free-30K-------------- columns 001 072
 COMMAND===>                                                   SCROLL===> CSR
****** ****************************** top of data ******************************
=prof> date: 05/06/89 10:24:02 user: MBE      init size:00000 size:00000
=prof> var   - 088,..recovery on  (0004 0000)...autosave off... empty line off
=prof> mask off.caps off.hex off      nulls off.autoren off 10..auto order off
=prof> log on  0001.mso on .fix off   .escape off + . ..tabs off    ...
=prof> advance off.protect off.limit on
=tabs>        *     *                       *                               *
=cols> ----+----1----+----2----+----3----+----4----+----5----+----6----+----7--
****** **************************** bottom of data ****************************
```

Meaning of profile parameters:

Item	Explanation
`init size`	Initial size of member.
`size`	Current size of member.
`var/fixed`	Denotes variable or fixed line length.
`recovery`	Specifies whether recovery function is on or off, together with number of line modifications for each checkpoint. Modifiable with the `RECOVERY ON/OFF` command.
`autosave`	Specifies whether the Editor will execute an automatic `SAVE` command when you issue the `END` main command. You can modify this setting using the `AUTOSAVE ON/OFF` main command.
`empty line`	Specifies suppression of blank line when inserting a blank line with the `I` line command. Modifiable with `EMPTY ON/OFF`.
`mask`	Specifies whether mask function is on or off. When on, the mask line is added on each insert line operation (see the `MASK` line command). Modifiable with `MASK ON/OFF` main command.

Item	Explanation
caps	Specifies whether data is to be translated into upper case. CAPS ON means upper case translation. Modifiable with the CAPS ON/OFF/PGM main command, where PGM means that all characters are translated except these enclosed in quotation marks or those that are part of a comment string. **Note:** Comment strings are identified by Natural syntax rules, even if the edited object is a non-Natural source (for example, PDS members containing JCL statements).
hex	Specifies whether data is to be displayed in hexadecimal mode. Modifiable with HEX ON/OFF main command.
nulls	Specifies whether null characters are to be set at the end of each line. Modifiable with the NULLS ON/OFF main command.
autoren	Automatically renumbers lines after modification. Modifiable with AUTOREN OFF/ON.
auto order	Automatically orders text within set boundaries. Can be activated using the AORDER command.
log	Specifies whether the log file is enabled or disabled for the session. When the log file is active (LOG ON), the UNDO command allows you to back out changes made since the previous Enter. Modifiable with the LOG ON/OFF command. The value VER is automatically set after you have issued the VERSIONS ON command. VER cannot be deactivated with LOG OFF.
mso	Specifies multi-session operations such as copy.
fix	Fixes specified number of columns to display when scrolling right. Modifiable with FIX *n* command, where *n* is the number of columns to be fixed.
escape	Specifies activation of escape character to precede line commands entered in the first column of the data area. Modifiable with ESCAPE ON/OFF char.
tabs	Activates tabulation. Modifiable with the TABS ON/OFF command.
advance	Specifies whether cursor moves to next line automatically when no main command is entered. Modifiable with the ADVANCE ON/OFF/PAGE main command.
protect	Specifies protection of prefix area (line numbers). Use the escape character to enter line commands from the data area when PROTECT ON is set. Modifiable with PROTECT ON/OFF/INS, where INS protects prefix area of lines inserted with the I line command.
limit	Maximum number of lines to be searched when the FIND or RFIND command is issued. Modifiable with the LIMIT *n* command, where *n* denotes the number of lines to be searched.

Invoke the Online Help Facility

The Editor provides a comprehensive online help facility.

To invoke the online help

- Press PF1 from the edit screen, or

  Or:

Issue the `HELP` main command.

The online help Main Menu appears and you can select the topic on which you wish to display help text (options 1 to 6 allow you to enter the main command help texts at different places in the help structure).

```
HELP---------------------- The NATURAL ISPF Editor ----------------------------
 COMMAND ==>                                                            EDITOR

  The Editor allows you to create or change source data.   The  following
  topics are presented only if selected by number.  Editor  main  commands
  are presented in alphabetical order, select a start value:

     1  -  ADVANCE
     2  -  COLS
     3  -  FILE
     4  -  INCLUDE
     5  -  ORDER
     6  -  SET

     7  - Editor line commands
     8  - Scrolling data

     9  - NATURAL commands
     A  - Sysout browsing commands



Enter-PF13--PF14--PF15--PF16--PF17--PF18--PF19--PF20--PF21--PF22--PF23--PF24---
     Help  e :c  Save; Suspe Rfind Rchan Up   Down  Swap  Left  Right Curso
```

Some help texts are longer than one help screen. In this case, the help text will notify you with the message Continued and you can scroll the help using scroll commands.

Editor PF Key Assignments

Some Editor main commands are assigned to PF keys by default. You can modify these settings:

Command	Explanation
`END`	PF3 and PF15
`RFIND`	PF5 and PF17
`RCHANGE`	PF6 and PF18
`UP`	PF7 and PF19
`DOWN`	PF8 and PF20
`RIGHT`	PF10 and PF22
`LEFT`	PF11 and PF23

The PF12 and PF24 keys move the cursor to the command field in the second line of the edit screen. See also the subsection ***PF Key Assignments*** in the section *Command Logic*.

Main Commands

ADVANCE

Specifies whether the cursor will move to the next line automatically after a line update.

AORDER

Specifies whether ordering of data within the set boundaries is to take place automatically after a line update or when terminating insert mode. If an unqualified `AORDER` command is issued, it is interpreted as an `AORDER ON` command. Default setting is `AORDER OFF`.

AUTOREN

Specifies whether the lines are renumbered automatically after modification. If an unqualified `AUTOREN` command is issued, it is interpreted as `AUTOREN ON`. Default is `AUTOREN OFF`.

AUTOSAVE

Specifies whether the Editor will execute an automatic `SAVE` command when you issue the `END` command.

If an unqualified `AUTOSAVE`/`ASAVE` command is issued, it is interpreted as an `AUTOSAVE ON` command. Default setting is `AUTOSAVE ON`. Note that the setting of this command also affects the `LOGOFF` session command.

BNDS

Sets boundaries at specified columns between which text can be formatted. An unqualified `BNDS` command sets boundaries at the first and last column of the edit screen (default).

BOTTOM

Scrolls data until last screen of data is displayed.

CANCEL

Backs out all changes to data made during the current editing session and leaves the Editor.

CAPS

Specifies upper case translator of entered data. If you issue the `CAPS` command without a parameter, `ON` is the default.

CENTER

Centers specified data within set boundaries.

CHANGE

Changes the first specified character string into the second. If the strings contain blanks, they must be delimited by quotation marks.

Note: If single quotation marks are part of the string to be changed, you must use different separators in the `CHANGE` command, for example double quotation marks.

COLS

Displays a line at the top of the editing subsection showing column positions.

COPY object-type library(member)

Copies data as specified in the command parameters into the current member.

If the current member already has data in it, you must mark the place where you wish the data to be inserted with an `A` or `B` line command.

If you wish to copy a member from the current library, use the command

```
COPY member
```

CREATE object-type library(member)

Creates the block marked by two `CC` line commands as new member in the specified object library.

If you wish to create a member in the current library, use the command

```
CREATE member
```

CURSOR

Typed in the command line, returns the cursor to the command line when you next press the ENTER key. This command is usually assigned to PF12. If you press PF12, the cursor will be placed in the command line.

CWINDOW

Copies a data window according to the command parameters.

DELETE

Deletes specified line(s) or line(s) containing a given character string. An unqualified `DELETE` command deletes the current line.

DOWN

Scrolls data down (forward). If the cursor is in the edit area, the scrolling amount is given by the current setting of the `SCROLL` field.

If an operand is specified and the cursor is in the command line, the scrolling amount (number of lines) is giving by the operand.

DWINDOW

Deletes the last defined data window.

EMPTY

Specifies whether blank line is to be suppressed. If an unqualified `EMPTY` command is used, it is interpreted as `EMPTY ON`. The default setting is `EMPTY OFF` (no suppression).

END

Stores the data including all changes and leaves the Editor.

Note: If `AUTOSAVE` is set to `OFF` and you have changed data, the Editor asks you to issue the `SAVE` or `CANCEL` command.

ESCAPE

Activates the specified escape character to precede line commands entered in the first column of the data area. If an unqualified `ESCAPE` command is used, it is interpreted as `ESCAPE ON`. The default setting is `ESCAPE OFF`.

EXCLUDE

Excludes specified line(s) or line(s) with the given character string from display. Use the `INCLUDE` command to redisplay excluded lines.

EXPORT

Transfers a Natural ISPF object (PDS member, Natural program, view, sequential dataset, SYSOUT file, workpool entry, library list) to Con-nect or a PC. For further information, see the subsection ***Natural Interface to External Environments*** in the section *Useful Features*.

FIND

Locates a given string. If the string contains blanks, it must be delimited by single quotation marks. The cursor will be placed on the beginning of the string. If the line containing the string was excluded from display, it will now be included.

Note: If single quotation marks are part of the string to be found, you must use different separators in the `FIND` command, for example double quotation marks.

FIX

Specifies number of columns starting with column 1 to remain in display when scrolling right. The default setting is `FIX OFF 000`.

HEX

Sets hexadecimal display mode on/off.

IMPORT

Transfers a Con-nect document or PC file into Natural ISPF. For further information, see the subsection ***Natural Interface to External Environments*** in the section *Useful Features*.

INCLUDE

Recalls specified line(s) or line(s) with the given character string removed from display using the `EXCLUDE` command.

An unqualified `INCLUDE` command recalls the first line of an excluded block.

JLEFT

Justifies the specified data within the set boundaries to the left.

JRIGHT

Justifies the specified data within the set boundaries to the right.

JUSTIFY

Specifies order mode for data marked by a `TO` line command or two `TOO` line commands.

LABEL

Marks the current line with a character or string.

LC

Changes specified line(s) or line(s) with the given character string to lower case. An unqualified `LC` command will change the current line to lower case.

LEFT

Scrolls data to the left. If no operand is given, the scrolling amount is given by the current setting of the `SCROLL` field.

LOCATE

Specifies line to become the current line. The line can be specified by its number, label, or search string.

Please note the following differences between the `LOCATE` and `FIND` commands:

- If you issue the `LOCATE` command with a character string (`L 'ABC'`), the string is only found if it starts in column 1; the `FIND` command searches the whole data area;
- With the `LOCATE` command, it is assumed that the data to be searched is sorted in ascending alphabetical order;
- When a line is located with the `LOCATE` command, the cursor is placed in the prefix area; with the `FIND` command, the cursor is placed on the found string and the line is not necessarily made the current line.
- There is no wildcard (*) notation for the `LOCATE` command. Executing `'LOCATE X*'` means finding the string `'X*'`.
- The command searches only in uppercase.
- The cursor is placed on the line which precedes the first line whose string is greater than the search string.

LIMIT

Specifies maximum number of lines to be searched when the `FIND` or `RFIND` command is issued.

LOG

Specifies whether the log file is enabled or disabled for the session. When the log is active (`LOG ON`), the `UNDO` command allows you to back out changes made since the previous ENTER.

MASK

Adds mask line in each insert line operation. You can define a mask line using the line command `MASK`.

MWINDOW

Moves a data window according to the command parameters. The original data window defined is deleted as a result of the `MWINDOW` command.

NULLS

Sets null characters at end of each line.

Default is `NULLS ON`.

ORDER

Orders the specified data within the set boundaries using the `BNDS` command.

POWER

Switches Editor to enter text mode (blank screen, entered data is automatically ordered between the set boundaries on ENTER).

PREFIX

Specifies whether the Editor prefix area (6-digit line numbers) is to be displayed (`PREFIX ON`) or not (`PREFIX OFF`). If `OFF` is specified, the data area on the screen is increased by 7 columns. If an unqualified `PREFIX` command is issued, it is interpreted as `PREFIX ON`.

PRINT

Prints the contents of the current member on the printer defined in your host environment.

PROFILE

Displays your edit profile at the top of the edit screen.

PROTECT

Protects the prefix area (line numbers). To enter line commands with the prefix area protected, type the line command in column 1 of the edit area preceded by the escape character.

RCHANGE

Repeats the last `CHANGE` command. Usually assigned to PF6.

RECOVER

Activates or deactivates the recovery feature for the current edit session. You can also specify the number of updates to be performed before a checkpoint is performed.

If you issue the `RECOVER` command without parameters, the default is `ON`.

RENUMBER

For PDS members and sequential datasets only. Specifies renumbering of the lines in the edit area according to the parameters. To deactivate line renumbering, use the `UNREN` command.

REPLACE object-type library(member)

Overwrites an existing member with the block marked by two `CC` line commands in the specified object library.

If the specified member does not exist, `REPLACE` works like `CREATE`, that is, a new member with the specified identifiers is created.

RESET

Resets all pending line commands and deletes all line labels. Also resets all definitions made during the edit session (for example, mask line, boundaries).

RFIND

Repeats the last `FIND` command. Usually assigned to PF5.

RIGHT

Scrolls data to the right. If no operand is given, the scrolling amount is given by the current setting of the `SCROLL` field.

SAVE 'text'

Stores the member in the current library, including any changes to data. The editing session continues. If you are editing an existing member and versioning is active, you can specify a reason for changing the member in the text parameter. The reason must be enclosed in quotation marks.

SET

The `SET` command opens a window with the current edit profile settings. You can modify any setting by overtyping the current value.

Note: Modified settings are valid for the current session only.

SORT

Sorts specified data in an edit session alphabetically in ascending or descending order. Entered without parameters, the data is sorted in alphabetical order starting in line 1, column 1.

TABS

Sets physical and logical tabulation.

If the `TABS` command is used without the character notation, physical tabulation is set. If the `TABS` command is used with the character notation, logical tabulation is referenced. The default setting is `TABS OFF` with a blank. For example, the command

```
TABS %
```

activates the tabulator and sets the logical tabulation character to the percent sign (%).

TOP

Scrolls data up (backwards) until first screen of data is displayed.

UC

Changes specified line(s) or line(s) with the given character string to upper case. An unqualified `UC` command will change the current line to upper case.

UNDO

If the log file is active (see the `LOG` command), the `UNDO` command backs out all changes made since the last time you pressed ENTER. If you then issue the `UNDO` command again, all changes made since the previous time you pressed ENTER are backed out. You can thus back out all changes one by one until you restore the member to the status at opening time.

UNREN

Deactivates the renumbering of lines.

UP

Scrolls data up (backward). If the cursor is in the edit area, the scrolling amount is given by the current setting of the `SCROLL` field.

If an operand is specified and the cursor is in the command line, the scrolling amount (number of lines) is giving by the operand.

WINDOW

Defines a data window to be copied or moved. The starting line and column and the end line and column of the window are specified in the command parameters.

XSWAP

Exchanges displayed with excluded lines.

Line Commands

This subsection lists the Editor line commands with a brief description in tabular form. You can enter a line command in the command field of any line by overtyping the line number on the left of your edit screen.

Alternatively, you can issue line commands from the first column of the data area if you precede them with the Editor escape character (see your edit profile). The line command affects the line on which it is entered.

You can also issue line commands from the main command line if you precede them with a colon (:) . The command addresses the line marked by the cursor.

Each of the line commands listed in the table below is available in EDIT sessions. For LIST and BROWSE sessions, only a subset of line commands is available. The List and Browse columns indicate which line commands are valid in these sessions: an X means "available", an asterisk (*) means the command is available for BROWSE sessions held in the Editor buffer pool, especially for the objects NATURAL, VIEW, Workpool OUTPUT, Predict descriptions, Con-nect documents and Container Files, but not for most other Natural ISPF objects that can be browsed. The numbers refer to footnotes at the end of the table.

Command	List	Browse	Explanation
A	- [1]	-	Move/copy data after this line.
B	- [2]	-	Move/copy data before this line.
O	-	-	Merges data with this line.
O*n*	-	-	Merges data with the following *n* number of lines.
OO	-	-	Marks the first line of a block with which block of data is to be merged. A second OO is required to mark the second line of the target block.
I	- [2]	-	Inserts one line (see also the main command EMPTY).
I*n*	-	-	Inserts *n* number of lines.
W	-	-	Opens window with one line.
W*n*	-	-	Opens window with *n* number of lines.
D	- [2]	-	Deletes this line.
D*n*	-	-	Deletes the next *n* number of lines.
DD	-	-	Marks the first line of a block to be deleted. A second DD is required to mark the second line of the block. Deletion is performed after second DD is entered.
DX	-	-	Deletes the line labelled .X.
DY	-	-	Deletes the line labelled .Y.
DX-Y	-	-	Deletes the block of lines from the line labelled .X to the line labelled .Y.
C	X	X	Copies this line.
C*n*	X	X	Copies the next *n* number of lines.
CC	X	X	Marks the first line of a block to be copied. A second CC is required to mark the second line of the block. Copying is performed after target line has been marked.
CX	-	-	Copies the line labelled .X. Inserts data after this line.
CY	-	-	Copies the line labelled .Y. Inserts data after this line.
CX-Y	-	-	Copies the block of lines from the line labelled .X to the line labelled .Y. Inserts data after this line.
M	-	-	Moves this line.
M*n*	-	-	Moves the next *n* number of lines.

Command	List	Browse	Explanation
MM	-	-	Marks the first line of the block to be moved. A second MM is required to mark the second line of the block. The move is performed after target line has been marked.
MX	-	-	Moves the line labelled .X. Inserts data after this line.
MY	-	-	Moves the line labelled .Y. Inserts data after this line.
MX-Y	-	-	Moves the block of lines from the line labelled .X to the line labelled .Y. Inserts data after this line.
N	-	-	Modifications made in this line do not take effect when ENTER is pressed.
R	- [3]	-	Repeats this line once.
R*n*	-	-	Repeats this line n number of times.
RR	-	-	Marks the first line of a block to be repeated. A second RR is required to mark the second line of the block.
RR*n*	-	-	Repeats block *n* number of times.
WS	-	-	Marks start of data window. The cursor position marks the column from which data will be read. If the cursor is not in the line for which the command is entered, column 1 is taken.
WS*n*	-	-	Data window starts in column n of this line.
WE	-	-	Marks end of data window. Works in the same way as WS. If the window is to start and end in the same line, overtype the WS command with the WE command. The Editor acknowledges the set window with message WW in the line command field.
WC	-	-	Copies the data window. The cursor position marks the column at which this line is to be split to insert the copied data.
WC*n*	-	-	Splits this line in column n, and copies the data between the two parts of the line.
WM	-	-	Moves the data window. Works in the same way as WC, but the original data is deleted after the copy operation.
X	X	*	Excludes this line.
X*n*	X	*	Excludes the following *n* number of lines.
XX	X	*	Marks the first line of the block to be excluded. A second XX is required to mark the second line of the block.
F	X	*	Includes the first of excluded lines.
F*n*	X	*	Includes the first *n* number of excluded lines.
L*n*	-	*	Includes the last *n* number of excluded lines.
LC	-	-	Changes this line to lower case.
LC*n*	-	-	Changes the following *n* number of lines to lower case.
LCC	-	-	Marks the first line of a block to be changed to lower case. A second LC is required to mark the second line in the block.
UC	-	-	Changes this line to upper case.
UC*n*	-	-	Changes the following *n* number of lines to upper case.
UC	-	-	Marks the first line of a block to be changed to upper case. A second UC is required to mark the second line in the block.

Command	List	Browse	Explanation
)	-	-	Moves this line right by one column.
)*n*	-	-	Moves this line right by *n* number of columns, irrespective of any other data in the line: you may lose data in moved line.
))*n*	-	-	Marks first line of a block to be moved right by n number of columns. A second))*n* is required to mark the last line of the block. The block is moved regardless of any other data in the block: you may lose data in the moved block.
(	-	-	Moves this line left by one column.
(*n*	-	-	Moves this line left by *n* number of columns regardless of any other data (you may lose data of moved lines).
((*n*	-	-	Marks first line of a block to be moved left by n number of columns.
			A second ((*n* is required to mark the last line of the block.
<	-	-	Moves data in this line left by one column.
>	-	-	Moves data in this line right by one column.
>*n*	-	-	Moves data in this line right by *n* number of columns (or up to first non-blank character: no data is lost)
>>*n*	-	-	Marks first line in a block to be moved to the right by *n* number of columns (or until first non-blank character). A second >>*n* is required to mark the second line of the block.
<*n*	-	-	Moves data in this line left by *n* number of columns (or until first non-blank character).
<<*n*	-	-	Marks first line in a block to be moved to the left by *n* number of columns (or until first non-blank character). A second <<*n* is required to mark the second line of the block.
S or TS	- [4]	-	Splits this line into two lines at cursor position; an empty line is also automatically inserted, but deleted if unused.
J or TJ	-	-	Joins next line with this one.
T	X	X	Scrolls the data to make the marked line the top line.
TE	-	-	Switches Editor to text enter mode (blank screen to end of screen).
TO	-	-	Orders this line within the set boundaries (see also the JUSTIFY main command).
TOO	-	-	Marks the first line of a block of text to be ordered within set boundaries. Requires a second TOO to mark the last line of the block (see also the JUSTIFY main command).
TF	-	-	Orders text from this line to next blank line or paragraph with right boundary.
TF*n*	-	-	Orders text from this line to next blank line or paragraph with column n as right boundary.
TC	-	-	Centers this line within set boundaries.
TCC	-	-	Marks first line of a block to be centered within the set boundaries. Requires a second TC to mark the second line of the block.
LJ	-	-	Justifies the data in this line with the left boundary.
LJJ	-	-	Marks the first line of a block to be justified with the left boundary. Requires a second LJJ to mark the last line of the block.

Command	List	Browse	Explanation
RJ	-	-	Justifies the data in this line with the right boundary.
RJJ	-	-	Marks the first line of a block to be justified with the right boundary. Requires a second RJJ to mark the last line of the block.
P	X	X	Prints this line at selected printer
PP	X	X	Marks first line of a block to be printed. Requires a second PP to mark the last line of the block.
BNDS	-	*	Displays boundary positions.
TABS	-	-	Displays tab positions.
COLS	- [5]	* 5	Displays column positions.
MASK	-	-	Defines a mask line to be displayed on insert line line command.
.X	X	X	Labels this line .X.

[1] In a list of datasets, the A line command is interpreted as the function command ALLOCATE.

[2] In a list of objects that allow the function commands BROWSE, INFORMATION or DELETE, the line commands B, I and D are interpreted as function commands.

[3] In a list of objects that allow the function command RENAME, the line command R is interpreted as a RENAME command.

[4] In a list of objects that can be edited or browsed, the line command S selects an object for editing or browsing, depending on the default function defined for that object type.

[5] Wherever the COLS line command is not available, you can use the Editor main command COLS instead.

8 z/OS Objects

- PDS Members 198
- Load Modules and CSECTs 207
- z/OS Dataset Maintenance 216
- z/OS Job Information 233
- z/OS System Operations 245

This chapter explains all the functions you can perform on objects only available in a z/OS environment.

PDS Members

In the PDS members maintenance facility, you can list, edit, delete, copy, rename and browse PDS members.

- If the member consists of job control, you can use the Natural ISPF macro facility and all types of macro statements. Macro expansion is performed at submission time (see the `SUBMIT` command below). When creating a new member, you can also use the Edit macro feature to automatically create text lines which can then be modified. For details on the macro facility, see the section *Macro Facility* in the *Natural ISPF Programmer's Guide*).
- If the member is a load module, its attributes, applied Zaps and the external references to it can be displayed. You can also generate a list of all the load module's `CSECT`s. Note that `CSECT`s are a separate object type in Natural ISPF, see the subsection ***Load Modules and CSECTs***.

To enter the PDS object maintenance facility

1 Select the `PDS` option from the Natural ISPF Main Menu.

  The PDS Objects Entry Panel appears:

```
-------------------------PDS-OBJECTS---ENTRY-PANEL----------------------------
COMMAND ===>



   Data Set Name ===> MBE.COMN.SOURCE
   Member        ===>
   Volume        ===>                        (If not catalogued)
   Password      ===>                        (If password protected)
   Scan for      ===>
   Edit macro    ===>
   Node          ===>









Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

2 Specify the PDS object you wish to maintain in the input fields and enter a function command in the command line.

Meaning of the input fields:

Field	Meaning
`Data Set Name`	Displays the PDS name last used. You can select any other PDS by overtyping this name, or generate a list of datasets for a specified volume in the same way as described for the Member field below. See also the subsection ***Support of HSM***.
`Member`	Name of required member. Leave blank or use a combination of strings and wildcards (* and _) to generate a list of member names in the specified library. See the subsection ***Selection Windows and Wildcards*** in section *Command Logic*. See also the subsection ***Support of HSM***.
`Volume`	Volume serial number. Required only for uncataloged datasets. Enter the asterisk wildcard (*) to list volume numbers or generate a list of volumes as described for the Member field above.
`Password`	System password if file is protected. Must be specified irrespective of read or write protection.
`Scan for`	Lists members which contain the string specified here. When you select a member from this list for `EDIT` or `BROWSE`, the cursor is placed on the first occurrence of this string in the member. Issue the `RFIND` command to find the next occurrence.

Field	Meaning
`Edit macro`	Name of macro object to be used as a model for the member. The specified macro is executed and loaded into the Editor. See section *Macro Facility* in the *Natural ISPF Programmer's Guide* for details. When used with `LIST`, the list contains all members according to the name criteria that use the specified macro as a model.
`Node`	Select Entire System Server node. Enter a question mark (?) and press ENTER to open a window in which all node numbers are scrolled with an `ACTIVE` or `INACTIVE` status report. If you do not specify a node, the default node is assumed.

Support of HSM - Hierarchical Storage Manager

Your Natural ISPF system may be configured to prompt every time you are trying to access a dataset or a member of a dataset which has been migrated by HSM to secondary storage. The prompt window allows you to continue working or to abort the current function and avoid the overhead of recalling a dataset which you do not want to use.

Usually your system is set up to wait until a dataset recall has terminated, but it can also perform a asynchronous recall in batch using a Natural ISPF exit. If you have questions about your configuration, contact your administrator.

Function Commands

The available function commands for PDS members are as follows (but see also the subsection ***Automatic Transfer of Commands*** in section *Useful Features*):

Command	Parameter Syntax
`BROWSE`	*`dataset(member)`* `VOL=`*`n`* `PASSWORD=`*`p`* `NODE=`*`id`*
`COPY`	*`dataset(member)`* `VOL=`*`n`* `PASSWORD=`*`p`* `NODE=`*`id`*`,` *`object-type object-parms`*`, REP`
`DELETE`	*`dataset(member)`* `VOL=`*`n`* `PASSWORD=`*`p`* `NODE=`*`id`*
`DOWNLOAD`	*`dataset(member)`* `VOL=`*`n`* `PASSWORD=`*`p`* `NODE=`*`id`*
`EDIT`	*`dataset(member)`* `VOL=`*`n`* `PASSWORD=`*`p`* `MACRO=`*`name`* `NODE=`*`id`*
`EXPORT`	*`dataset(member)`* `VOL=`*`n`* `PASSWORD=`*`p`* `NODE=`*`id`*`,` *`target-environment`*
`EXTERNS`	*`dataset(member)`* `VOL=`*`n`* `PASSWORD=`*`p`* `NODE=`*`id`*
`INFO`	*`dataset(member)`* `VOL=`*`n`* `PASSWORD=`*`p`* `NODE=`*`id`*
`HOLD`	*`dataset(member)`* `VOL=`*`n`* `PASSWORD=`*`p`* `NODE=`*`id`*
`LIST`	*`dataset(*_*)`* `VOL=`*`n`* `PASSWORD=`*`p`* `SC=`*`string`* `MACRO=`*`name`* `NODE=`*`id`*
`PLAY`	*`dataset(member)`* `VOL=`*`n`* `PASSWORD=`*`p`* `NODE=`*`id`*
`PRINT`	*`dataset(member)`* `VOL=`*`n`* `PASSWORD=`*`p`* `NODE=`*`id`*`,` *`printer-name`* `CC`
`RENAME`	*`dataset(member)`* `VOL=`*`n`* `PASSWORD=`*`p`* `NODE=`*`id`*`,` *`new-name`*
`SUBMIT`	*`dataset(member)`* `VOL=`*`n`* `PASSWORD=`*`p`* `NODE=`*`id1`*`, TARGET=`*`id2`*
`UPLOAD`	*`dataset(member)`* `VOL=`*`n`* `PASSWORD=`*`p`* `NODE=`*`id`*

Command	Parameter Syntax
ZAPS	*dataset(member)* VOL=*n* PASSWORD=*p* NODE=*id*

The *dataset* parameter is optional, Natural ISPF then takes the current dataset name, or if you issue a command from outside the PDS facility, from your profile.

A full description of these commands is contained in the section ***Command Reference***. The object parameters correspond to the input fields on the PDS Objects Entry Panel.

Note: If you issue any of the above function commands from outside the PDS facility, you must specify the object-type parameter P before the object parameters.

For more information on the EXTERNS and ZAPS commands, see the subsection ***Load Modules and CSECTs***.

Example: INFORMATION

The following display is the result of the command:

```
INFO P FHI.SOURCE(E2AIBM)
```

```
 -------------------------- PDS MEMBER INFORMATION -----------------------------
 COMMAND ===>


     Data set name :  FHI.SOURCE

       Volume Serial  :  ADA004
       Device Type    :  3380
       Organization   :  PO
       Record format  :  FB
       Record length  :  80
       Block size     :  6000

     Member        :  E2AIBM3            Alias of: E2AIBM

       Version        :  01.21          Maximum no. of versions  : 20_
       Created        :  2001-01-20     Actual  no. of versions  :
       Modified       :  2001-01-26 12:10
       User           :  FHI
       Initial        :  36
       Size           :  36

 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
       Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

The information fields in this display generally speak for themselves.

Notes:

1. If the member is an alias for another PDS member, `Alias of` contains the name of the original member. For example `E2AIBM3` is an alias of `E2AIBM`.
2. You can modify the `Maximum number of versions` field here to override the default value set by the system administrator.

Copying PDS members with aliases

With Entire System Server 3.2 and above, aliases are automatically copied when the member for which they are an alias is copied. Consequently, Natural ISPF ignores aliases when copying a complete PDS or PDSE dataset (that is, copying with no member selection criteria).

Caution: When copying a complete PDS or PDSE dataset with an earlier version of Entire System Server, aliases are not copied.

Listing Volumes, Datasets, Members and CSECTs

The following items can be listed using the `LIST` command and selection criteria in the parameter input fields of the PDS Objects Entry Panel:

- Volumes according to prefix;
- Dataset names according to prefix;
- All datasets on a specified volume (`VTOC`); you can optionally narrow the list down by specifying a prefix, suffix or a string for dataset names using the asterisk wildcard (*) (see the subsection *Selection Windows and Wildcards*);
- Members in a partitioned dataset (PDS); you can optionally specify a prefix for member names using wildcards (* and _);
- Members in a partitioned dataset (PDS) according to prefix or wildcard specification, but whose data contain a certain character string (Scan for option);
- Previous versions of the specified member;
- CSECTs, if the specified member is a load module.

If you wish to generate lists using function command syntax in the command line of any screen, you must address the following object types:

Type of List	Object-Type
List of volumes	VOL
List of datasets	D
List of members, versions or CSECTs	P

This section deals with maintenance of PDS members only. For details on dataset handling and volumes, see the subsection ***z/OS Dataset Maintenance***. For details on previous versions, see the subsection ***Versioning*** in the section *Useful Features*. For details on CSECTs, see the subsection ***Load Modules and CSECTs***.

Below are some examples of the `LIST` function command using full command syntax.

Example LIST (1)

The following figure shows an example of a list of all members in a PDS library generated using the command:

```
LIST P FHI.SOURCE(*)
```

```
LIST-PDS:FHI.SOURCE(*) -------------------------- Row 0 of 51 - Columns 010 076
 COMMAND===>                                                   SCROLL===> CSR
   MEMBER            VV.MM  CREATED  MODIFIED TIME  SIZE   INIT   TID   ID
** ******************************** top of list ******************************
   ACB
   ADACDEP           01.01  19940921 19940921 11:39    148    148       FHI
   A2EIBM            01.38  19940120 19940126 11:57     36     36       FHI
   A2ESIE            01.57  19940120 19940126 12:46     36     36       FHI
   A2EWIN            01.01  19950327 19950327 14:49     36     36       FHI
   CCALL
   CPCHAIN
   CPROLOG
   CQ
   CREGS
   E2AIBM            01.21  19940120 19940126 12:10     36     36       FHI
   E2ASIE            01.22  19940120 19940126 12:45     36     36       FHI
   E2AWIN            01.02  19950327 19950327 15:07     36     36       FHI
   FHITST            01.01  19980918 19980918 15:57      2      2       FHI
   FHITST5           01.21  19940120 19940126 12:10     36     36       FHI
   NATOSDF           01.01  19980115 19980115 16:54    376    376       FHI
   NAT1ZAP           01.01  19940329 19940329 15:02    180    180       FHI
   NAT2ZAP           01.01  19940329 19940329 15:02     54     54       FHI
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Meaning of the column headings:

Column	Meaning
MEMBER	Name of member.
VV	Version number of member. When a member is created, this field shows 01. It can be increased using the HOLD function command. [1]
MM	Modification level: shows the number of times the current version of the member has been modified. A plus sign (+) in this field means that previous versions exist for the member. [1]
CREATED	Date this member was created.
MODIFIED	Date of last modification.
TIME	Time of last modification.
SIZE	Number of lines in member.
INIT	Initial size of this version.
TID	Terminal ID at which the member was last modified.
ID	User who modified member last.

[1] The string ALIAS in the columns below VV.MM indicates that the member name shown in this line is an alias name.

Example LIST (2)

The following figure illustrates a list of members generated using the SCAN= option in the command:

```
LIST P MBE.COMN.SOURCE(NSPF*) SC=NATURAL
```

```
LIST-PDS:MBE.COMN.SOURCE(NSPF*)/SCAN=NATURAL --- Row 0 of 12 - columns 010 076
 COMMAND===>                                                  SCROLL===> CSR
   MEMBER              NUM FIRST FOUND
** ******************************** top of list ******************************
   NSPFINST             69 .chap1 'Installing NATURAL ISPF'
   NSPFREF0            111 .init 'NATURAL ISPF Reference Manual'
   NSPFREF1            122 .chap1 'Working with NATURAL ISPF'
   NSPFREF2             91 .chap1 'NATURAL Objects'
   NSPFREF3             10 NATURAL ISPF provides a facility with which you can
   NSPFREF4              9 (DATA SETS) from the NATURAL ISPF main menu. This di
   NSPFREF5              4 (SYSTEM) from the NATURAL ISPF main menu. This displ
   NSPFREF6             16 output files of NATURAL programs for further mainten
   NSPFREF7             24 with NATURAL ISPF.
   NSPFREF8             39 .chap1 'NATURAL ISPF Commands'
** ******************************* bottom of list ****************************
```

The list displays all members starting with NSPF which contain the string NATURAL in the PDS library MBE.COMN.SOURCE.

Meaning of the column headings:

Column	Meaning
MEMBER	Name of member.
NUM	Number of occurrences of specified string in member.
FIRST FOUND	First occurrence of specified string in member.

The lists appear in Natural ISPF Editor format in browse mode. This means you can use all available Editor browse commands (UP, DOWN, BOTTOM, TOP, FIND, LOCATE).

If you select a member from a list generated with the Scan for option for EDIT or BROWSE, the cursor is placed on the first occurrence of the string. If you then issue the RFIND command, the cursor is placed on the next occurrence.

Example LIST (3)

the command:

```
LIST P MBE.*(A*E*)
```

opens an active help window containing a list of all datasets with prefix MBE on the default node. After selecting a dataset name from the list, all members in the dataset are listed that begin with A and also have an E in their name.

Line Commands

You select a member from a list by typing in a line command in the input field preceding the member name and pressing ENTER. Each line command is an abbreviation of a function command (exception - the special L line command for a member):

Line Command	Corresponding Function Command
B	BROWSE
CP	COPY
D	DELETE
DW	DOWNLOAD
E	EDIT
EX	EXPORT
HL	HOLD
I	INFORMATION
L	LIST previous versions of the member, or, if the member is a load module, list the CSECTs.

Line Command	Corresponding Function Command
PL	PLAY
PR	PRINT
R	RENAME
SB	SUBMIT
UP	UPLOAD
XT	EXTERNS
ZP	ZAPS

Line commands can also be used as valid abbreviations of function commands entered in the command line of any system screen.

Local Commands

In Edit Mode:

If you display a PDS member in Editor format in EDIT mode, you can issue local commands from the Editor command line in addition to Editor commands.

The following local commands are available:

Command	Meaning
IMPORT	Imports a PC file or Con-nect document into the PDS member (see section ***Useful Features***).
PASSWORD *password*	If the dataset is password-protected, use this command to enter the valid password in order to update the member. If you enter the PASSWORD command without parameter, a window prompts you for the password. Password input in the window is invisible.
REGENERATE	Available for members written using the Edit macro option. Reexecutes the specified macro object and writes the result in protected lines in the current edit session. Any defined user code remains in place. For details, see section *Macro Facility* in the *Natural ISPF Programmer's Guide*.

In List Mode:

If you display lists of PDS libraries or members in Editor format, you can issue the following local commands in addition to Editor scroll commands: ALL, LAYOUT, RELIST and SORT. For detailed information, see the subsections in section ***Useful Features***.

Previous Versions

Previous versions of PDS members can be listed and retrieved (see the line command `L` for `LIST`). They are separate objects in Natural ISPF, accessible via the PDS Objects Entry Panel or using function commands that address object type `PV` from any screen. To activate the versioning feature, you must issue the command `VERSIONS ON` before starting your edit session, unless your administrator has activated obligatory versioning for the member's dataset. For details, see the subsection *Versioning* in the section *Useful Features*.

Load Modules and CSECTs

The `LIST` and `INFORMATION` displays of PDS load modules are different from those of other PDS members in that they show load module attributes. Additionally, all Zaps applied to a load module, and all external references of the load module can be displayed. CSECTs of a load module can also be listed, browsed or edited.

Natural ISPF provides commands to generate a Zap in the user workpool and apply the Zap to a load module. With Natural ISPF's `DOWNLOAD` and `UPLOAD` commands, you can transfer z/OS load modules to and from a PC via Entire Connection.

Example: LIST load modules

The following list is the result of the command:

```
LIST P JWO.COMN.LOAD(*)
```

where the specified library is a load library:

```
LIST-PDS:JWO.COMN.LOAD(*) ---------------------- Row 0 of 15 - Columns 010 076
 COMMAND===>                                                  SCROLL===> CSR
   MEMBER           ATTRIBUTES
** ******************************** top of list ******************************
   ADASCR
   EDITBPA
   EDRBPMM
   EDRPHSE          RENT REUS
   EDRWFMT          AUTH RENT REUS
   NATCOM21
   NATIDBOK
   NATIDB07
   NATPARM
   NATTEXT
   NATTS022
   NATTXT2          RENT REUS
   NRMCMP           RENT REUS
   TEST1            RENT REUS
   TTEST
** ****************************** bottom of list ******************************


Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right :s
```

Meaning of the possible attributes:

Attribute	Meaning
ALIAS	Member name is an alias name
AUTH	APF-authorized
LOADONLY	Only loadable
NOTEXEC	Not executable
OVLY	Overlay
REFR	Refreshable
RENT	Reentrant
REUS	Reusable
SCTR	Scatter
TEST	Test option (TSO)

Load modules can be selected from the list using the same line commands as described for PDS members, except that the EDIT and BROWSE commands apply to CSECTs and there are some more commands available. The functions specific to load modules and CSECTs are:

Line Command	Corresponding Function Command and Meaning
B	BROWSE a CSECT (select from prompt window).
E	EDIT a CSECT (select from prompt window).
I	INFORMATION on the load module.
L	LIST CSECTs for the module.
XT	EXTERNS. Display external references to the load module.
ZP	ZAPS. Display all Zaps applied to the load module.

For complete function command syntax, see the subsection *Function Commands*.

Example: INFORMATION

The following display is the result of the command:

```
INFO P PPEX.NATURAL.LOAD(IUPD)
```

where the specified member is a load module:

```
-------------------------- PDS MEMBER INFORMATION -----------------------------
COMMAND ===>


    Data set name :  PPEX.NATURAL.LOAD

      Volume Serial  :  SYSF10
      Device Type    :  3390
      Organization   :  PO
      Record format  :  U
      Record length  :
      Block size     :  32760

   Module         :  IUPD

      Length         :  112        00000070
      Attributes     :
      Entry point    :  OSCOPY
      Linkage date   :  93/04/21
      Linkage editor :  566528408 V 2   R 4
      ZAPS           :  NONE
      Unresolved     :  NONE
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

The information fields speak for themselves. Note, however, that the module length is given both in numeric and hexadecimal format and that the number of unresolved references is also given.

Example: EXTERNS

The following display is the result of the command:

```
EXTERNS P JWO.COMN.LOAD(NATPARM)
```

where the specified member is a load module. The display shows a list of all external references made to load module NATPARM:

```
EXTERNS-PDS:JWO.COMN.LOAD(NATPARM) -------------- Row 0 of 62 - Columns 001 031
 COMMAND===>                                                   SCROLL===> CSR
   ADDRESS REFERENCE TYPE IN-CSECT
** ********* top of list *********
   000024  SYSPRM    ER   ACMPARM
   000064  ADATBLE   SD   ACMPARM
   0002AC  ACMUDEB   WX   ACMPARM
   0002B8  ACMLIST   WX   ACMPARM
   0002C4  CMRSP     WX   ACMPARM
   0002D0  ACMUB     WX   ACMPARM
   0002DC  ACMKEY    WX   ACMPARM
   0002E8  ACMZUL    WX   ACMPARM
   0002F4  CMDDP     WX   ACMPARM
   000300  CMMPP     WX   ACMPARM
   00030C  CMMA2     WX   ACMPARM
   000318  CMMA3     WX   ACMPARM
   000324  CMGLB     WX   ACMPARM
   000330  CMADA     WX   ACMPARM
   00033C  CMZAP     WX   ACMPARM
   000348  CMDIX     WX   ACMPARM
   000354  CMMASH    WX   ACMPARM
   000360  CMRADAR   WX   ACMPARM
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right :s
```

The display shows the external references to load module NATPARM.

Meaning of the column headings:

Column	Meaning
ADDRESS	Address of the external symbol.
REFERENCE	Name of the external symbol.
TYPE	External symbol type. Possible options: CM Common ER External symbol is unresolved. LR Label reference

Column	Meaning
	PC Private code PS Pseudo register SD External symbol is resolved. WX Weak external symbol is unresolved.
IN-CSECT	Name of the CSECT in which the reference to the external symbol appears.

Example: ZAPS

The following display is the result of the command:

```
ZAPS P JWO.COMN.LOAD(NATPARM)
```

where the specified member is a load module.

```
ZAPS-PDS:JWO.COMN.LOAD(NATPARM) ----------------- Row 0 of 1 - Columns 001 026
 COMMAND===>                                                  SCROLL===> CSR
   CSECT    DATE     IDR-DATA
** ******* top of list ******
   ACMPARM  93/02/08 NO IDENT
   ACMPARM  93/08/12 NO IDENT
** ***** bottom of list *****









Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right :s
```

The display shows the Zaps applied to the load module NATPARM. Meaning of the column headings:

Column	Meaning
CSECT	Name of Zap CSECT.
DATE	Date the Zap was generated.
IDR-DATA	Information on the Zap.

Example: LIST

The following display is the result of the command:

```
LIST P JWO.COMN.LOAD(NATPARM)
```

where the specified member is a load module. The display shows all CSECTs for the specified load module:

```
LIST-CST:JWO.COMN.LOAD(NATPARM)-/* --------------- Row 0 of 6 - Columns 001 055
 COMMAND===>                                                  SCROLL===> CSR
   CSECT            OFFSET LENGTH AMODE RMODE ID    LR-ID
** ********************* top of list *********************
   ACMPARM          000000 000709 ANY   24    00001 00000
   ADATBLE          000730 0000F4 ANY   24    00004 00000
   NSTATIC          000710 00001C ANY   24    00064 00000
   CMDBID           000828 0000FF ANY   24    00067 00000
   CMTRF            000928 000002 ANY   24    00068 00000
   ADBSYS           000930 000058 ANY   24    00069 00000
** ******************* bottom of list ********************





Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right :s
```

Meaning of the column headings:

Column	Meaning
CSECT	Name of CSECT.
OFFSET	Offset of CSECT in physical block.
LENGTH	Length of CSECT in hex.
AMODE	AMODE assigned to CSECT.
RMODE	RMODE assigned to CSECT.
ID	ID of external symbol.
LR-ID	ID of label reference (LR) symbol.

You can select a CSECT for further maintenance from the list using one of the following line commands:

Line Command	Corresponding Function Command and Meaning
B	BROWSE the CSECT.
E	EDIT the CSECT.
XT	EXTERNS. Display external references of the CSECT
ZP	ZAPS. Display all Zaps applied to the CSECT.

CSECTs are a separate object type in Natural ISPF. When requesting a function for a CSECT using function command syntax from any system screen, you must therefore use the object code CST.

Example: BROWSE

the command:

```
BROWSE CST JWO.COMN.LOAD(NATPARM)
```

opens a prompt window in which you can specify the name of the required CSECT. The following display shows the result for CSECT ACMPARM:

```
BROWSE-CST:JWO.COMN.LOAD(NATPARM)-/ACMPARM -------------------- Columns 044 076
 COMMAND===>                                                    SCROLL===> CSR
   OFFSET DATA                                TEXT             DISASSEMBLY
** ******************************** top of list ******************************
   000000 00007FF8 00007800 00010000 00050000 ??"8?? ????????     SU      LE
   000010 000002A4 00006C00 40404040 40404040 ???u??%?                    MD
   000020 80000008 00000000 F0F0F0F7 000002A4  ???????0007???u SSM
   000030 00000000 003A000A 07D0040A C80A0606 ?????????§??H???
   000040 32010000 00000018 E2D6D9E3 40404040 ????????SORT     LTER
   000050 00000000 0000001E 000A00F9 00000000 ??????????9????
   000060 0000055C 00000730 40404040 40404040 ???*????             BALR    BCR
   000070 00000000 00000000 00000000 00000000 ???????????????
   000080 00000000 00000000 00000000 00000563 ??????????????
   000090 000A0000 00000000 00380028 00000000 ???????????????
   0000A0 00000000 00000684 00000006 00000703 ???????d????????             BCTR
   0000B0 40404040 40404040 0000FF00 00000928         ?? ????? STH STH STH STH
   0000C0 00000000 00000000 00010048 F140044B ?????????? 1 ?.
   0000D0 6B84427E 010301F1 B06F0000 00000000 ,d =???1 ??????? SD  STC
   0000E0 00001770 00000000 05F5E0FF 05F5E0FF ??? ?????5§ ?5§      XR
   0000F0 D900006C 643203F8 00851214 05F5E0FF R??% ??8?e???5§
   000100 00000011 8E60C1C0 10800009 00000000 ???? -A§? ??????         SRDA
   000110 00000000 00000000 00000000 00000000 ???????????????
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right :s
```

If you enter the local command LINE 2, the Assembler commands in the column DISASSEMBLY are ordered in a separate line under the hexadecimal representation:

```
BROWSE-CST:JWO.COMN.LOAD(NATPARM)-/ACMPARM -------------------- Columns 044 059
 COMMAND===>                                                    SCROLL===> CSR
   OFFSET DATA                                 TEXT
** ********************** top of list **********************
   000000 00007FF8 00007800 00010000 00050000 ??"8?? ?????????
              SU       LE
   000010 000002A4 00006C00 40404040 40404040 ???u??%?
                       MD   STH STH  STH STH
   000020 80000008 00000000 F0F0F0F7 000002A4  ???????0007???u
          SSM               SRP SRP
   000030 00000000 003A000A 07D0040A C80A0606 ?????????§??H???
                            BCR SPM      BCTR
   000040 32010000 00000018 E2D6D9E3 40404040 ????????SORT
          LTER                       STH STH
   000050 00000000 0000001E 000A00F9 00000000 ???????????9????

   000060 0000055C 00000730 40404040 40404040 ???*????
              BALR     BCR  STH STH  STH STH
   000070 00000000 00000000 00000000 00000000 ??????????????

   000080 00000000 00000000 00000000 00000563 ?????????????
                                         BALR
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right :s
```

To revert to the default display, issue the `LINE 1` local command.

Example: EDIT

The command:

```
EDIT CST JWO.COMN.LOAD(NATPARM)
```

opens a prompt window in which you can specify the name of the required CSECT. The resulting display is the same as for `BROWSE` (see above), but the hexadecimal characters are modifiable. Modifications can subsequently be evaluated in order to generate or apply Zaps to the edited CSECT of the module (see the description of local commands `GENERATE` and `SAVE` below).

Local Commands

If you display a CSECT in Editor format in `EDIT` or `BROWSE` mode, you can issue local commands from the Editor command line in addition to Editor commands.

The following local commands are available:

In BROWSE mode:

Command	Meaning
`LINE 2`	Displays CSECT in 2 line format: the Assembler commands appear directly under the corresponding hexadecimal representation (see example above).
`LINE 1`	Issued after `LINE 2`, redisplays CSECT in default format.

In EDIT mode:

Command	Meaning
`GENERATE`	Generates a Zap in the user workpool based on the modifications made to the CSECT.
`SAVE`	Generates a Zap in the user workpool based on the modifications made to the CSECT, and applies the Zap using `AMASPZAP` to the load module.

You can display or edit the Zap generated in the user workpool by addressing object type `OUT`. For example, when you have edited CSECT `ACMPARM` and issued the command `GENERATE`, the following command opens an edit session with the Zap in the user workpool:

```
SPLIT;EDIT OUT ACMPARM TYPE=ZAP
```

```
EDIT-CST:JWO.COMN.LOAD(NATPARM)-/ACMPARM ---------------------- Columns 044 076
 COMMAND===>                                                    SCROLL===> CSR
   OFFSET DATA                                 TEXT             DISASSEMBLY
** ******************************** top of list ******************************
   000000 00007FF8 00007800 00010000 00050000 ??"8?? ?????????     SU      LE
EDIT-OUT:ACMPARM/TYPE=ZAP ------------------------------------- Columns 001 072
 COMMAND===>                                                    SCROLL===> CSR
****** ****************************** top of data ******************************
000001 NAME NATPARM ACMPARM
000002 VER 000070 00000000
000003 REP 000070 40404040
****** ***************************** bottom of data ******************************





Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right :s
```

The output in the workpool shows the name of the load module and CSECT, and offset and the change made.

z/OS Dataset Maintenance

The dataset maintenance facility allows you to perform functions on the following objects:

- Sequential (`PS`) and PDS-type (`PO`) datasets. Available functions include `LIST`, `BROWSE`, `RENAME`, `DELETE`, `COMPRESS`, `CATALOG`/`UNCATALOG`, and `ALLOCATE` datasets. You can also display dataset information and access PDS members from a list of datasets.
- GDG-type datasets (Generation Data Groups). These can be handled as any other dataset. When a GDG is allocated, Natural ISPF automatically allocates a model dataset, which appears on the catalog with type `GDG-BASE`. When the `DELETE` function is requested for a GDG, both the GDG and its associated files are deleted on confirmation.
- Volumes. You can display information on volumes.

To enter the dataset maintenance facility

- Select the `DATASETS` option from the Natural ISPF Main Menu.

  The Data Sets Entry Panel appears:

```
---------------------------DATA-SETS---ENTRY-PANEL-----------------------------
COMMAND ===>



   Data Set Name ===>
   Volume        ===>                          ( If not catalogued     )
   Password      ===>                          ( If password protected )
   Node          ===>



Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

You can specify the dataset you wish to maintain in the input fields and enter a function command in the command line.

Meaning of the input fields:

Field	Meaning
Data Set Name	Displays the dataset last used. You can select any other dataset by overtyping this name, or generate a list of datasets on the specified volume by using a combination of strings and asterisk wildcard (*) as described in the subsection ***Selection Windows and Wildcards*** in section ***Command Logic***. See also the subsection ***Support of HSM***.
Volume	Required only if the dataset is not cataloged. To list volume serial numbers, leave the Data Set Name field blank and use the asterisk wildcard (*) as described above.
Password	System password if dataset is protected.
Node	Select Entire System Server node. Enter a question mark (?) and press ENTER to open a window in which all node numbers are scrolled with an ACTIVE or INACTIVE status report. If you do not specify a node, the default node is assumed.

Function Commands

The available function commands are:

Command	Object Parameter Syntax
ALLOCATE	*dataset* VOL=*n*
BROWSE*	dataset VOL=*n* NODE=*id*
CATALOG	*dataset* VOL=*n*
COMPRESS**	*dataset* VOL=*n* NODE=*id*
COPY	dataset VOL=*n* NODE=*id*, *object-type object-parameters*, REP
DELETE	*dataset* VOL=*n* NODE=*id*
EDIT*	*dataset* VOL=*n* NODE=*id*
EXPORT*	*dataset* VOL=*n* NODE=*id*
EXTENTS	*dataset* VOL=*n* NODE=*id*
INFORMATION	*dataset* VOL=*n* NODE=*id*
LIST	*_* VOL=*n* NODE=*id*
PRINT*	*dataset* VOL=*n* NODE=*id*, *printer-name* CC NO
RENAME	*dataset* VOL=*n* NODE=*id*, *new-name*
UNCATALOG	*dataset* VOL=*n* NODE=*id*

Notes:

1. * Apply to sequential datasets only.
2. ** Applies to partitioned datasets only.

A full description of these commands, including the function parameters, is contained in section *Command Reference*. The object parameters correspond to the input fields on the Data Sets Entry Panel.

Notes:

1. If you issue any of the above function commands from outside the dataset facility, you must specify the object-type parameter D before the object parameters.
2. In the case of multi-volume datasets, only the first volume need be specified in the VOL parameter.

Example: CATALOG

Use the CATALOG command to catalog a dataset. If you issue the CATALOG command and specify a volume serial number using the VOL option, the catalog function is performed without further prompting. If you issue the CATALOG command with only the dataset name, Natural ISPF presents you with the following screen:

```
-------------------------------CATALOG-DATASET---------------------------------
COMMAND ===>

  Data set Name ===> RW.COMN.SOURCE
  Volsers       ===>        ===>        ===>        ===>        ===>
                            ===>        ===>        ===>        ===>
                            ===>        ===>        ===>        ===>
                            ===>        ===>        ===>        ===>
  Device        ===>









 Press ENTER to catalog, END to cancel
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      HELP  SPLIT END   SUSPE RFIND RCHAN UP    DOWN  SWAP  LEFT  RIGHT CURSO
```

Type in the volume serial number of the dataset and device type. Press ENTER to perform the catalog operation.

Example: ALLOCATE

With the ALLOCATE command, you can allocate a new dataset specified in the dataset name parameter. Natural ISPF provides a special feature here. If you issue the ALLOCATE command for an existing (allocated) dataset, the information for the dataset is displayed in the following format:

```
----------------------------- ALLOCATE DATASET --------------------------------
COMMAND ==>


   Data set name ===> MBE.SYSF.ISPF.141.DOC
      VOLume SERial        ===> COM811 /        /        /        /
   or Generic UNIT         ===> 3380
      Dataset Organization ===> PO
      Space Units          ===> CYL                            (BLK,TRK,CYL)
      Quantity:    Primary ===>            Secondary ===>
      Directory Blocks     ===> 70
      Record Format        ===> FB
      Record Length        ===> 80
      Block Size           ===> 3120
      Rlse                 ===> NO                             (YES,NO)
      Contiguous           ===> NO                             (YES,NO)
      Round                ===> NO                             (YES,NO)
      Expiration Date      ===>                                ('YYDDD')
      Catalog Data Set     ===> YES                            (YES,NO)
      GDG            limit ===>      More attributes ===>      (YES,NO)
      Node                 ===> 148
Press ENTER to allocate
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right :s
```

You can type in the name of the dataset to be allocated by overtyping the value in the `Data set name` field. Modify any other value as required and press ENTER to perform the allocation.

The `ALLOCATE DATASET` screen will also appear if you copy a dataset to a target which does not exist.

Meaning of the fields:

Field	Meaning
`Volume Serial`	Serial number of the volume on which the dataset is to be allocated. You can specify up to 5 volumes for multi-volume datasets.
`Generic Unit`	If you do not specify a volume, specify the generic identifier from which a volume is to be selected (e.g. 3380).
`Dataset Organ.`	For example: `PO` (PDS), `PS` (sequential dataset), `DA` (direct access).
`Space Units`	Space type for dataset. Possible values: `BLK` Blocks `TRK` Tracks `CYL` Cylinders

Field	Meaning
`Primary Qty.`	Initial quantity to be allocated.
`Secondary Qty.`	Additional quantity to be allocated if dataset fills.
`Directory Blocks`	Must be specified for `PO`-type datasets.
`Record Format`	For example: `FB` (fixed block), `VB` (variable block), `FBA` (fixed block ANSI control characters), etc.
`Record Length`	Given in bytes.
`Block Size`	Given in bytes.
`Rlse`	`YES` specifies that allocated space is released if not used by dataset.
`Contiguous`	`YES` specifies that tracks or cylinders must be adjacent.
`Round`	`YES` specifies that space is automatically rounded up to the nearest cylinder if tracks or blocks are specified as space units.
`Expiration Date`	Date the dataset expires. Until this date is reached, each attempt to update or delete the dataset causes a console message, requiring an operator reply.
`Catalog Data Set`	`YES` specifies the dataset is to be automatically cataloged when allocated.
`GDG limit`	A value in this field identifies the file to be allocated as GDG. The value specifies the maximum number of Generation Datasets that can be associated with the GDG being defined.
`More attributes`	Specify `YES` to define SMS attributes or more GDG attributes. Another window opens for entering additional data (see example below).
`Node`	Entire System Server node number on which the dataset is to be allocated.

Type in the required values in the input fields and press ENTER to allocate the dataset.

When the allocation of a dataset fails and an error message showing a hexadecimal reason code is displayed, you can use the command `HELP` to display a more meaningful explanation of the error.

Example: ALLOCATE (More attributes)

Specifying `YES` in the `More attributes` field of the allocate screen opens the following window:

```
---------------------------- ALLOCATE DATASET -------------------------------
COMMAND ==>


    +-------------------Additional dataset attributes-------------------+
    !                                                                    !
    !                        SMS ATTRIBUTES                              !
    ! MANAGEMENT CLASS           :  ________                             !
    ! STORAGE CLASS              :  ________                             ! L)
    ! DATA CLASS                 :  ________                             !
    !                                                                    !
    !                        GDG ATTRIBUTES                              !
    ! EMPTY ALL CATALOG ENTRIES, :  ___ (YES- all, NO - last only)       !
    !  WHEN LIMIT REACHED                                                !
    ! DSCB CLEAR FROM VTOC,      :  ___ (YES - deleted, NO - left)       !
    !   WHEN DSN IS UNCATALOGED                                          !
    +--------------------------------------------------------------------+
      Expiration Date      ===>                                ('YYDDD')
      Catalog Data Set     ===> YES                            (YES,NO)
      GDG            limit ===>      More attributes ===> yes  (YES,NO)
      Node                ===> 148
Press ENTER to allocate
```

Meaning of the fields:

Field	Meaning
SMS Attributes:	
`MANAGEMENT CLASS`	The management class to be used for obtaining the data management-related information for SMS (migration, backup and retention criteria) to allocate the dataset.
`STORAGE CLASS`	The storage class to be used for obtaining the storage-related information for dataset allocation.
`DATA CLASS`	The data class to be used for obtaining the data-related information (`SPACE`, `LRECL`, etc.) for dataset allocation.
GDG Attributes:	
`EMPTY ALL CATALOG ENTRIES`	When the `LIMIT` value is reached: `YES` specifies all Generation Datasets are uncataloged. `NO` specifies only the oldest Generation Dataset is uncataloged.
`DSCB CLEAR FROM VTOC`	When the Generation Dataset is uncataloged (due to `DELETE` command or `EMPTY ALL` value): `YES` specifies the dataset's DSCB is deleted from the VTOC. The GDS no longer exists.

Field	Meaning
	NO specifies the dataset's DSCB is not deleted from the VTOC. The DSCB is left in the VTOC and the dataset can be processed as any non-VSAM dataset.

For more information on Generation Data Groups, see the relevant section in the documentation *MVS/DFP: Access Method Services for the Integrated Catalog Facility*.

Examples: ALLOCATE

- Assuming you have a cataloged dataset named MYFILE, the command:

  ```
  AL D MYFILE
  ```

  displays the information for the MYFILE. Modify the display for the dataset to be allocated.
- If you wish to allocate a new, uncataloged dataset without a model, the command:

  ```
  AL D NEWFILE VOL=com811
  ```

  displays the blank allocation screen for file NEWFILE on volume COM811.

Example: INFORMATION (1)

The following figure is the result of the command:

```
INFORMATION D FHI.SOURCE
```

The fields in the information screen reflect the specification of the allocation parameters described for the ALLOCATE command above, with additional information such as date of last reference, current number of cylinders or tracks and allocated extents, and in the case of PDS libraries, number of members, directory blocks and unused blocks:

```
--------------------------- DATA SET INFORMATION ------------------------------
COMMAND ===>


    DATA SET NAME :  FHI.SOURCE

    GENERAL DATA
      VOLUME SERIAL  :  ADA004
      DEVICE TYPE    :  3380                CURRENT-ALLOCATION
      ORGANIZATION   :  PO                    ALLOCATED CYLINDERS:  10
      RECORD FORMAT  :  FB                    ALLOCATED EXTENTS :   1
      RECORD LENGTH  :  80
      BLOCK SIZE     :  6000                CURRENT UTILIZATION
      ALLOCATION TYPE:  CYL                   PERCENT USED:  37
      1ST EXTENT     :  10      CYL 0   TRK
      SECONDARY      :  1
      SECURITY       :  NONE

                                             PARTITIONED DATA SET
    CREATION DATE  :  1993-07-13              NUMBER-OF-MEMBERS :  51
    LAST-REFERENCE :  1998-09-29              DIRECTORY-BLOCKS  :  20
    EXPIRATION-DATE:  ********                UNUSED-BLOCKS     :
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Example: INFORMATION (2)

The following figure is the result of the command:

```
INFORMATION D JWO.SYSF.GDG
```

The fields in the information screen reflect the specification of the allocation parameters for a GDG described for the `ALLOCATE` command above. Additional information provided by this display is the number of generations with creation date, volume serial number and device series of each Generation Dataset, as well as the assigned name suffix indicating the version number with which as the Generation Datasets appears in a list:

```
--------------------------- DATA SET INFORMATION -----------------------------
COMMAND ===>


    GDG NAME      :  JWO.SYSF.GDG

                                        GENERATIONS:
    GENERAL DATA                        NUM  CREATED  VOLSER SERIES NAME-SUFFIX
      GDG LIMIT NUMBER     :  5         0   18/02/93 USR8A6 3380   '.G0002V00
      EMPTY ALL FROM CATLG :  NO       -1   30/11/92 USR8A6 3380   '.G0001V00
        WHEN LIMIT REACHED
      DSCB CLEAR FROM VTOC :  NO
        WHEN UNCATALOGED
      EXPIRATION DATE      :  93356


    MODEL DSN ATTRIBUTES
      ORGANIZATION         :  PS
      RECORD FORMAT        :  VBA
      RECORD LENGTH        :  137
      BLOCK SIZE           :  9240

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right :s
```

The above example shows a GDG with a limit of 5 datasets. If this limit is reached, the oldest dataset is deleted from the catalog, but not from the VTOC. Currently, there are two generations. The attributes of the model dataset are shown at the bottom of the display.

Example: EXTENTS

The following figure is the result of the command:

```
EXTENTS D MBE.COMN.SOURCE
```

The display shows the dataset name, volume serial number and number of extents, and lists the extents, giving the disk addresses (hexadecimal) and the size (decimal) of each one:

```
 EXTENTS-DS:MBE.COMN.SOURCE ------------------------------------ Columns 001 056
  COMMAND===>                                                   SCROLL===> CSR
    VOLSER NR. BEGIN(CYL/TRK) END(CYL/TRK)   SIZE(CYL/TRK)
 ** ********************** top of list *********************
    COM810   0 0171  00       0173  0E             3  0
             1 018F  00       0191  0E             3  0
             2 02C1  00       02C1  0E             1  0
 ** ******************** bottom of list ********************









 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
       Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Examples: DELETE

- When deleting a dataset with `CONFIRM ON`, the confirmation window asks you for a catalog update. The following window opens as the result of the command:

```
DELETE D ISP.INST.ALL
```

```
+-------------------------------------------------------------------------+
! Confirm scratching  ISP.INST.ALL                                         !
! From catalog and all volsers                                             !
! with Y              _                                                    !
+-------------------------------------------------------------------------+
```

- If you mark the input field with `Y`, the dataset catalog is updated after the `DELETE` operation.

  If you issue the `DELETE` command for an empty GDG, the following confirmation window appears. You can specify `NO` in the appropriate field to retain the model dataset on the disk:

```
+-------------------------------------------------------------------------+
! Confirm scratching GDG JWO.SYSF.GDG                                      !
! with Y              _                                                    !
! Delete also model DSN from catalog volume            YES (YES / NO)      !
! (GDG would be deleted, though retention period -------- not expired)     !
+-------------------------------------------------------------------------+
```

- If you issue the DELETE command for a non-empty GDG, the following confirmation window opens. You can specify the delete parameters as appropriate before the deletion is performed:

```
+-------------------------------------------------------------------------+
! Confirm scratching GDG JWO.SYSF.GDG                                      !
! with Y              _                                                    !
! though GDG DSN's exist (DSN's would also be removed from catalog)        !
! Delete also all GDG DSN's from VTOC's                YES (YES / NO)      !
! Delete also model DSN from catalog volume            YES (YES / NO)      !
+-------------------------------------------------------------------------+
```

Example: Special BROWSE Command

The command:

```
BROWSE DS *
```

lists dataset names for which short IDs are defined (see the subsection *Library Definition* in section *Profile Maintenance*). Select a dataset for display by marking it with any character in the window and pressing ENTER.

Example: Special LIST Command

The command:

```
LIST DS *
```

lists available library short names (see also the subsection *Library Definition* in section *Profile Maintenance*).

Example: LIST Datasets

The following list of datasets is the result of the command:

```
LIST D JWO.COMN.*
```

```
LIST-DS:JWO.* ---------------------------------- Row 0 of 65 - Columns 046 076
 COMMAND===>                                                  SCROLL===> CSR
   DATA SET NAME                                  VOLSER  SERIES CLASS TYPE
** ******************************** top of list ********************************
   JWO.COMN.ALLOC                                 DCN010  3390   DASD  NONVSAM
   JWO.COMN.BIGNEU                                DCN010  3390   DASD  NONVSAM
   JWO.COMN.BIGNEU22                              DCN010  3390   DASD  NONVSAM
   JWO.COMN.BIGNEU23                              DCN010  3390   DASD  NONVSAM
   JWO.COMN.CATTESN                               DCN010+ 3390   DASD  NONVSAM
   JWO.COMN.CPTEST                                COM811  3380   DASD  NONVSAM
   JWO.COMN.DEMOGGG                               DCN010  3390   DASD  NONVSAM
   JWO.COMN.DEMOGGG1                                              -NA-  GDG
   JWO.COMN.EMPTY1                                COM811  3380   DASD  NONVSAM
   JWO.COMN.JCL1                                  COM811  3380   DASD  NONVSAM
   JWO.COMN.LTEST                                 DCN010  3390   DASD  NONVSAM
   JWO.COMN.L133                                  DCN010  3390   DASD  NONVSAM
   JWO.COMN.L133A                                 DCN010  3390   DASD  NONVSAM
   JWO.COMN.L133B                                 DCN010  3390   DASD  NONVSAM
   JWO.COMN.L2500                                 DCN010  3390   DASD  NONVSAM
   JWO.COMN.L300                                  DCN010  3390   DASD  NONVSAM
   JWO.COMN.L300A                                 DCN010  3390   DASD  NONVSAM
   JWO.COMN.L80                                   DCN010  3390   DASD  NONVSAM
  Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
        Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right :s
```

The list shows all cataloged datasets on the default node starting with `JWO.COMN`.

Use the `RIGHT` command (usually assigned to PF11) to scroll right and display the full `TYPE` field as in the example above.

Meaning of the data provided for each dataset:

Column	Meaning
`VOLSER`	Volume serial number. If a dataset resides on multiple volumes, only one line is displayed in the list and a plus sign (+) is added to the volser to indicate that it is a multi-volume dataset.
`SERIES`	Device series .
`CLASS`	Device class. Possible values: `COMM` Communications `CTCA` Channel-to-channel adapter

Column	Meaning
	DASD Direct access DISP Display station TAPE Tape UREC Unit record
TYPE	Dataset type (for example, NONVSAM, CLUSTER, DATA, GDG BASE).

Example: LIST a VTOC

The following shows an example VTOC generated using the command:

```
LIST D * VOLSER=COM811
```

```
 LIST-DS:*/VOL=COM811--------------------------------Row-1-of-215-columns-046-076
  COMMAND===>                                                  SCROLL===> PAGE
    DATA SET NAME                                        DSORG LRECL BLKSZ RCFM
** ****************************** top of list ******************************
    SYS1.VTOCIX.COM811                                   PS    02048 02048 F
    ADABAS.COMN.V5.DEP.SOURCE                            PO    00080 06000 FB
    ADABAS.COMN.V5.DEP.MACLIB                            PO    00080 03600 FB
    PRD.COMN.NOC111.LOAD                                 PO    00000 06447 U
    ADABAS.COMN.V5.DEP.LOAD                              PO    00000 04096 U
    UP.COMN.PERS                                         PS    00068 10000 VB
    ADABAS.COMN.V5.LOAD                                  PO    00000 06447 U
    ADABAS.COMN.TELEX                                    PO    00080 03600 FB
    ADL100.COMN.LOAD                                     PO    06233 19069 U
    ALO.COMN.INPL                                        PS    04624 04628 VB
    ADL100.COMN.OUTPUT                                   PO    00132 05280 FB
    ADL100.COMN.UNLOAD                                   PS    09996 10000 VB
    ALO.COMN.SOURCE                                      PO    00080 03120 FB
    ADL100.COMN.LIB                                      PO    00080 03120 FB
    BF.COMN.SOURCE                                       PO    00080 03120 FB
    ALO.COMN.LOAD                                        PO    00000 19040 U
    BMRK.COMN.ADABAS.V513.LOAD                           PO    00000 19069 U
    AOS.COMN.V112.INPL                                   PS    04624 04628 VB
 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
       Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

The list contains all datasets on volume COM811. You must scroll right to display all the information provided (RIGHT command, usually assigned to PF11).

Meaning of the data provided for each dataset:

Column	Meaning
DSORG	Dataset organization. For example: DA Direct access PO Partitioned dataset (PDS) PS Sequential dataset
LRECL	Logical record length in bytes
BLKSZ	Block size in bytes
RCFM	Record format. For example: F Fixed length record F Fixed blocked record F Fixed blocked record (ANSI control characters) U Unformatted record V Variable blocked record
SIZE CYL/TRK	Size of dataset in cylinders and tracks
% USED	Percentage of dataset used
CREATED	Dataset creation date
LAST-ACC	Date of last access
SMS	SMS-controlled device/unit (YES, NO).
UPD	Has the file been updated since the last backup (YES, NO)?

Example: LIST Volumes

Volumes are separate objects in Natural ISPF, but have no Entry Panel. You can access volume information using the Data Sets Entry Panel, or using a function command that addresses object type VOL.

The following figure illustrates a list of all volumes generated using the command:

```
LIST VOL *
```

```
LIST-VOL:* ------------------------------------- Row 0 of 470 - Columns 013 076
 COMMAND===>                                                   SCROLL===> CSR
   UNIT VOLSER          SERIES STATUS  MOUNT      FREE(CYL/TRK) CONTIG(CYL/TRK)
** ******************************** top of list ******************************
   100  BMC003          3380   ONLINE  RESIDENT     64 , 0029         52 , 0000
   101  BMC004          3380   ONLINE  RESIDENT     14 , 0031          5 , 0014
   102                  3380   OFFLINE               0 , 0000          0 , 0000
   103                  3380   OFFLINE               0 , 0000          0 , 0000
   104                  3380   OFFLINE               0 , 0000          0 , 0000
   105                  3380   OFFLINE               0 , 0000          0 , 0000
   106                  3380   OFFLINE               0 , 0000          0 , 0000
   107                  3380   OFFLINE               0 , 0000          0 , 0000
   108                  3380   OFFLINE               0 , 0000          0 , 0000
   109                  3380   OFFLINE               0 , 0000          0 , 0000
   10A                  3380   OFFLINE               0 , 0000          0 , 0000
   10B                  3380   OFFLINE               0 , 0000          0 , 0000
   10C                  3380   OFFLINE               0 , 0000          0 , 0000
   10D                  3380   OFFLINE               0 , 0000          0 , 0000
   10E                  3380   OFFLINE               0 , 0000          0 , 0000
   10F                  3380   OFFLINE               0 , 0000          0 , 0000
   110  BMCRES          3380   ONLINE  RESIDENT     27 , 0036         11 , 0000
   111  BMC001          3380   ONLINE  RESIDENT     81 , 0167         69 , 0014
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right :s
```

You must scroll right to display all the information provided (`RIGHT` command, usually assigned to PF11).

Meaning of the data according to column heading:

Column	Meaning
`UNIT`	Unit address
`VOLSER`	Volume serial number
`SERIES`	Device series
`STATUS`	Device status. Possible values: `CHANGE` Device status is changing. `OFFLINE` Device is offline. `ONLINE` Device is online.
`MOUNT`	Device mount status. Possible values: `MOUNT PEND` Mount is pending. `NOT READY` Device not ready. `REMOVABLE` Device is removable (e.g. a tape).

Column	Meaning
	RESERVED Device is reserved. RESIDENT Device is resident (e.g. a hard disc).
FREE(CYL/TRK)	Number of free cylinders / unused tracks
CONTIG (CYL/TRK)	Contiguous cylinders / tracks
CLASS	Device class. Possible values: COMM Communications CTCA Channel-to-channel adapter DASD Direct access DISP Display station TAPE Tape UREC Unit record
SMS	SMS-controlled device/unit (YES, NO).

You can select a volume and list its contents (VTOC) by entering the line command L in the input field preceding the UNIT notation (see the following subsection).

The line command I (INFORMATION) displays volume information. For details, see the **example** in the subsection *z/OS System Operations*.

Line Commands

When listing datasets, the line commands in the following table are available. When listing volumes, only the I and L line commands are available.

Line Command	Corresponding Function	Remarks
A	ALLOCATE	
B	BROWSE	Sequential datasets only (but see the example of the special BROWSE command).
CM	COMPRESS	Partitioned datasets only.
CP	COPY	
CT	CATALOG	
D	DELETE	
E	EDIT	Sequential datasets only.
ET	EXTENTS	
EX	EXPORT	Sequential datasets only.
I	INFORMATION	

Line Command	Corresponding Function	Remarks
L	LIST	From a list of volumes, lists datasets on a volume; from a list of datasets, lists members of a PDS.
PR	PRINT	Sequential datasets only.
R	RENAME	
U	UNCATALOG	

Line commands can also be used as valid abbreviations for function commands entered in the command line of any screen.

Local Commands

In Edit Mode:

If you display a sequential dataset in Editor format in EDIT mode, you can issue local commands from the Editor command line in addition to Editor commands.

The following local commands are available:

Command	Meaning
IMPORT	Imports a PC file or Con-nect document into the sequential dataset (see the section *Useful Features*).
PASSWORD *password*	If the dataset is password-protected, use this command to enter the valid password in order to update the dataset.

In List Mode:

If you display lists of datasets or volumes, the following local commands are available in addition to Editor scroll commands: ALL, LAYOUT, RELIST and SORT. For detailed information, see the sub-sections in the section *Useful Features*.

z/OS Job Information

The job information facility enables you to display job information. You can also use this facility to handle SYSOUT files, and to hold, release or purge jobs that reside in the JES input or output queues.

To enter the job information facility, select the JOBS option from the Natural ISPF Main Menu. The Jobs Entry Panel appears, for example:

```
-----------------------------JOBS---ENTRY-PANEL--------------------------------
COMMAND ===>

   Job Name      ===>
   Job Number    ===>
   Type          ===>                        ( J, S, T                )
   Queue         ===>                        ( X,I,O,H,W              )
   Class         ===>                        ( Output Class           )
   JOB Class     ===>                        ( Execution Class        )
   Sysout        ===>                        ( SI,JL,SM,SO,CC         )
   Sys Num       ===>                        ( Sysout file number     )
   User          ===>
   Node          ===>









Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

You can specify the job or SYSOUT file you wish to handle in the input fields and enter a function command in the command line.

The meaning of the input fields is explained in the following table:

Field	Meaning
`Job Name`	Job name on the job card. Leave blank or use a combination of strings and wildcards (* and _) to generate a list of jobs matching the job name pattern. See the subsection ***Selection Windows and Wildcards*** in the section *Command Logic*.
`Job Number`	Job number assigned
`Type`	Type of job. Possible options: `J` Standard operating system job `S` Started task `T` TSO user Combinations of types are possible, for example `ST` selects types `S` and `T`.
`Queue`	Job entry system queue. Possible options: `H` Output queue jobs on `HOLD`

Field	Meaning
	I Input queue (held and non-held jobs) O Output queue X Executing queue W Input queue jobs on HOLD
Class	Installation-specific output class. Combinations are possible to list jobs of several classes.
Jobclass	Installation-specific execution job class. Only one class can be selected.
Sysout	Type of file when using the BROWSE command. Possible options: AL All SYSOUT files of the job CC Summary of job steps and condition codes JL JCL of selected job SI SYSIN data SM System messages SO SYSOUT data
Sys Num	File number of SYSOUT type.
User	Select job by user. For JES2, this the programmer. For JES3, this is the JOB owner.
Node	Node ID of machine on which the job ran. Enter a question mark (?) and press ENTER to list available nodes. Leave blank to select the default node.

Only the combination of job name and number uniquely identifies a submitted job (if more than one copy of the same job was started).

Function Commands

The available function commands are:

Command	Object Parameter Syntax
BROWSE	*job-number job-name* SI=*f* FILE=*n*
CC	*job-number job-name* NODE=*id*
CHANGE	*job-number job-name* CLASS=*c* NODE=*id*, NEWCLASS=*x* DEST=*d*
COPY	*job-number job-name* SI=*f* FILE=*n* NODE=*id*, *object-type object-parms* REP
EXPORT	*job-number job-name* SI=*f* FILE=*n* NODE=*id*, *target-environment*
FOLLOW	*job-number job-name* NODE=*id*
HOLD	*job-number job-name* NODE=*id*
LIST	*_* TYPE=*t* QUEUE=*q* CLASS=*c* NODE=*id*
PRINT	*job-number job-name* SI=*f* FILE=*n* NODE=*id*, *printer-name* CC NO

Command	Object Parameter Syntax
PURGE	*job-number job-name* CLASS=*c* NODE=*id*
RELEASE	*job-number job-name* CLASS=*c* NODE=*id*, NEWCLASS=*x*
STATUS	*job-number job-name* NODE=*id*

A full description of these commands (including the function parameters) is contained in section *Command Reference*. The object parameters correspond to the input fields on the Jobs Entry Panel.

Notes:

1. You can specify a job by name, number or both. If you use both, the job number must come first.
2. The BROWSE, PRINT, COPY and EXPORT commands apply to Object Type S (SYSOUT). If any of these commands are issued for a job, the command is automatically transferred to Object Type S.
3. The commands RELEASE and CHANGE accept parameter NEWCLASS to specify the new class in a direct command. The CHANGE command also accepts parameter DEST to change the destination of a job.
4. The FOLLOW, STATUS, CC and HOLD commands are available for Object Type J (JOB) only.
5. If you issue any of the above function commands from outside the JOBS facility, you must specify the appropriate object-type parameter (J or S) before the object parameters.
6. For a detailed list of specific execution job classes, enter selection criteria in the Jobclass field and XI in the Queue field.

Listing Jobs and Queue Entries of the Job Entry Subsystem - JES

The following items can be listed using the LIST command and selection criteria in the parameter input fields of the JOBS Entry Panel:

Item Listed	Use Input Field	Function Command Syntax
Jobs according to a job name prefix	Jobname	LIST JOB *prefix*
TSO sessions, started tasks or standard batch jobs	Type	LIST JOB *prefix* TYPE=*ttt*
Jobs on the INPUT, EXECUTING, OUTPUT or HELD-OUTPUT queue	Queue	LIST JOB *prefix* QUEUE=*qqqq*
Jobs associated with one to four output classes	Class	LIST JOB *prefix* CLASS=*cccc*
Jobs associated with a specific execution job class	Jobclass / Queue	LIST JOB *prefix* JOBCLASS=*j* QUEUE=XI

As with other LIST functions, the selection criteria can be combined to reduce the number of listed entries.

Below are some examples of function commands using command syntax.

Example: LIST Jobs

The following figure shows the result of the command:

```
L J OP*
```

The list contains all jobs starting with the string `OP`:

Note: The display of job information may vary from this example: some columns may be blank. This depends on the JES version in use, JES installation options and Natural ISPF configuration options. Ask your system administrator.

```
LIST-JOB:OP* ---------------------------------- Row 0 of 129 - Columns 024 076
 COMMAND===>                                                   SCROLL===> CSR
   JOBNAME  TYP JOBNO Q C PR  DEST    STAT  OWNER    STEP     PROC     CPU
** ******************************** top of list ******************************
   OPDMSLST JOB 32031 O X 128 LOCAL   AVAIL TSR
   OPN09LNU JOB 31089 O X 144 LOCAL   AVAIL SAL
   OPN09ADU JOB 31088 O X 128 LOCAL   AVAIL SAL
   OPFWDMS4 JOB 30505 O 9 032 LOCAL   AVAIL TSR
   OPRST009 JOB 31026 O X 144 LOCAL   AVAIL SAL
   OPSMFREP JOB 31005 O X 144 LOCAL   AVAIL HEC
   OPSMFREP JOB 31004 O X 144 LOCAL   AVAIL HEC
   OPSMFREP JOB 31003 O X 144 LOCAL   AVAIL HEC
   OPSMFREP JOB 31001 O X 144 LOCAL   AVAIL HEC
   OPSMFREP JOB 31000 O X 144 LOCAL   AVAIL HEC
   OPFWDMS5 JOB 30507 O 9 032 LOCAL   AVAIL TSR
   OPFWDMS2 JOB 30502 O 9 032 LOCAL   AVAIL TSR
   OPFWDMS1 JOB 30501 O 9 048 LOCAL   AVAIL TSR
   OPFWDMS3 JOB 30504 O 9 048 LOCAL   AVAIL TSR
   OPFCOBTN JOB 30875 O X 128 LOCAL   AVAIL TSR
   OPFCOBKP JOB 30866 O X 128 LOCAL   AVAIL TSR
   OPFCOBDE JOB 30867 O X 128 LOCAL   AVAIL TSR
   OPFCOBTS JOB 30868 O X 128 LOCAL   AVAIL TSR
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

This list of submitted jobs appears in Editor format in `BROWSE` mode and is refreshed every time you press ENTER.

Use the `RIGHT` command (usually assigned to PF11) to scroll right in order to display more information (next two screens):

```
LIST-JOB:OP* ---------------------------------- Row 0 of 129 - Columns 077 129
 COMMAND===>                                                  SCROLL===> CSR
   JOBNAME  TYP JOBNO Q C REG   IO     RECORDS EXECUTION START    END
** ******************************** top of list ******************************
   OPDMSLST JOB 32031 O X                 1133 20011212  12:49:11 2001121212:49
   OPNO9LNU JOB 31089 O X                  380 20011212  08:54:47 2001121208:54
   OPNO9ADU JOB 31088 O X                 1052 20011212  08:54:30 2001121208:54
   OPFWDMS4 JOB 30505 O 9                67029 20011212  03:00:09 2001121208:25
   OPRST009 JOB 31026 O X                  282 20011212  08:16:32 2001121208:18
   OPSMFREP JOB 31005 O X                   49 20011212  08:01:23 2001121208:01
   OPSMFREP JOB 31004 O X                   49 20011212  08:01:20 2001121208:01
   OPSMFREP JOB 31003 O X                   49 20011212  08:01:18 2001121208:01
   OPSMFREP JOB 31001 O X                   49 20011212  08:01:16 2001121208:01
   OPSMFREP JOB 31000 O X                   49 20011212  08:01:14 2001121208:01
   OPFWDMS5 JOB 30507 O 9                76020 20011212  03:00:10 2001121207:40
   OPFWDMS2 JOB 30502 O 9                40573 20011212  03:00:08 2001121206:53
   OPFWDMS1 JOB 30501 O 9                35613 20011212  03:00:08 2001121206:40
   OPFWDMS3 JOB 30504 O 9                31511 20011212  03:00:09 2001121205:53
   OPFCOBTN JOB 30875 O X                 1168 20011212  05:36:45 2001121205:42
   OPFCOBKP JOB 30866 O X                 1152 20011212  05:36:26 2001121205:42
   OPFCOBDE JOB 30867 O X                 1052 20011212  05:36:35 2001121205:42
   OPFCOBTS JOB 30868 O X                 1052 20011212  05:36:35 2001121205:41
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

```
LIST-JOB:OP* ---------------------------------- Row 0 of 129 - Columns 086 138
 COMMAND===>                                                  SCROLL===> CSR
   JOBNAME  TYP JOBNO Q C     RECORDS EXECUTION START    END              SYSID
** ******************************** top of list ******************************
   OPDMSLST JOB 32031 O X        1133 20011212  12:49:11 2001121212:49:13 DAEF
   OPNO9LNU JOB 31089 O X         380 20011212  08:54:47 2001121208:54:50 DAEF
   OPNO9ADU JOB 31088 O X        1052 20011212  08:54:30 2001121208:54:34 DAEF
   OPFWDMS4 JOB 30505 O 9       67029 20011212  03:00:09 2001121208:25:21 DAEF
   OPRST009 JOB 31026 O X         282 20011212  08:16:32 2001121208:18:09 DAEF
   OPSMFREP JOB 31005 O X          49 20011212  08:01:23 2001121208:01:24 DAEF
   OPSMFREP JOB 31004 O X          49 20011212  08:01:20 2001121208:01:22 DAEF
   OPSMFREP JOB 31003 O X          49 20011212  08:01:18 2001121208:01:19 DAEF
   OPSMFREP JOB 31001 O X          49 20011212  08:01:16 2001121208:01:17 DAEF
   OPSMFREP JOB 31000 O X          49 20011212  08:01:14 2001121208:01:15 DAEF
   OPFWDMS5 JOB 30507 O 9       76020 20011212  03:00:10 2001121207:40:14 DAEF
   OPFWDMS2 JOB 30502 O 9       40573 20011212  03:00:08 2001121206:53:41 DAEF
   OPFWDMS1 JOB 30501 O 9       35613 20011212  03:00:08 2001121206:40:47 DAEF
   OPFWDMS3 JOB 30504 O 9       31511 20011212  03:00:09 2001121205:53:41 DAEF
   OPFCOBTN JOB 30875 O X        1168 20011212  05:36:45 2001121205:42:06 DAEF
   OPFCOBKP JOB 30866 O X        1152 20011212  05:36:26 2001121205:42:02 DAEF
   OPFCOBDE JOB 30867 O X        1052 20011212  05:36:35 2001121205:42:00 DAEF
   OPFCOBTS JOB 30868 O X        1052 20011212  05:36:35 2001121205:41:45 DAEF
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Use the LEFT command (usually assigned to PF10) to scroll left again.

Meaning of the column headings:

Field	Meaning
JOBNAME	Name of job on job card.
TYP	Type of job: Standard Job JOB Started Task STC TSO user session TSO
JOBNO	JES job number.
Q	JES queue type: Input queue I Executing queue X Output queue O Held output queue H
C	JES job class (when Q=I or X) or output class (when Q=O or H).
PR	JES priority within current queue.
DEST	JES printout destination name.
STAT	Job status (can be blank or HOLD).
OWNER	ID of user who submitted the job.
STEP	Name of step currently executed (only when job is active).
PROC	Name of procedure currently executed (only when job is active).
CPU	CPU time consumed in minutes consumed by the address space (only when job is active).
REG	Amount of real storage used by address space in Kbytes (only when job is active).
IO	Number of I/Os performed so far (only when job is active).
RECORDS	Total number of records in job SYSOUT files.
EXECUTION START	Date and time when job execution started.
END	Date and time when job execution ended.
SYSID	System in which job executes in a Sysplex environment.

Note: In certain cases, a job may be represented by two or more lines of the LIST session. Typically this occurs if a job has produced output in several output classes.

Example: LIST Batch Jobs and Started Tasks

The following figure shows the result of the command:

```
LIST JOB DA* TYPE=JS
```

The list contains all batch jobs (`J`) and started tasks (`T`) having the prefix `DA`. No TSO sessions are selected:

```
LIST-JOB:DA*/TYPE=JS ---------------------------- Row 0 of 15 - Columns 026 076
 COMMAND===>                                                   SCROLL===> CSR
   JOBNAME  TYP JOBNO Q C PR DEST    STAT  OWNER    STEP     PROC     CPU     R
** ******************************** top of list ******************************
   DAEFTCP  STC 02018 X   15               DAEFTCP
   DAEFTCP2 STC 02019 X   15               DAEFTCP2
   DAEFCO   STC 02027 X   15               DAEFCO
   DAEFNWP  STC 02058 X   15               DAEFNWP
   DAEFETB  STC 02061 X   15               DAEFETB
   DAEFETB2 STC 02062 X   15               DAEFETB2
   DAEFWAP  STC 02064 X   15               ACF2STC
   DAEFPQA  STC 04218 X   15               DAEFPQA
   DAEFTMDB STC 04322 X   15               DAEFTMDB
   DAEFTCSM STC 04323 X   15               DAEFTCSM
   DAEFCOT  STC 04751 X   15               DAEFCOT
   DAEFETB  STC 02088 O C 08               DAEFETB
   DAEFETB2 STC 02091 O C 08               DAEFETB2
   DAEFPQA  STC 03731 O X 08               DAEFPQA
   DAEFCOT  STC 02102 O 9 08               DAEFCOT
** ****************************** bottom of list ******************************


Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Example: LIST Jobs in HOLD

The following figure shows the result of the command:

```
LIST JOB O* QUEUE=H CLASS=XW
```

The list contains only jobs in the held output queue with output in one of the specified output classes `X` and `W`:

```
LIST-JOB:O*/QUEUE=H/CLASS=XW -------------------- Row 0 of 35 - Columns 026 076
 COMMAND===>                                                   SCROLL===> CSR
   JOBNAME  TYP JOBNO Q C PR DEST    STAT  OWNER    STEP     PROC     CPU     R
** ******************************** top of list ******************************
   OPGWDMS  JOB 01343 H W 00         JHOLD SOL
   OPQDDMS  JOB 01333 H W 00         JHOLD SOL
   OPGDDMS  JOB 01327 H W 00         JHOLD SOL
   OPHBV016 JOB 01312 H W 00         JHOLD ACF2BAT
   OPDMPHBV JOB 01295 H W 00         JHOLD ACF2BAT
   OPSDAYU  JOB 01304 H W 00         JHOLD +++++++
   OPHDDMS  JOB 01293 H W 00         JHOLD MEL
   OPPLGHBV JOB 01287 H W 00         JHOLD ACF2BAT
   OPDMPHBV JOB 01269 H W 00         JHOLD ACF2BAT
   OPDMPHBV JOB 01264 H W 00         JHOLD ACF2BAT
   OPULDOPS JOB 07330 H W 00         JHOLD ACF2BAT
   OPDMP008 JOB 05544 H W 00         JHOLD ACF2BAT
   OPFWDMS1 JOB 05203 H W 00         JHOLD SOL
   OPDMPHBV JOB 05278 H W 00         JHOLD ACF2BAT
   OPPLGHBV JOB 05251 H W 00         JHOLD ACF2BAT
   OPDMPHBV JOB 05185 H W 00         JHOLD ACF2BAT
   OPPLGHBV JOB 05156 H W 00         JHOLD ACF2BAT
   OPHBV016 JOB 05099 H W 00         JHOLD ACF2BAT
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Example: LIST SYSOUT Files

SYSOUT files are separate objects in Natural ISPF. You can access SYSOUT files via the Jobs Entry Panel, or using a function command from any system screen with object-type parameter `S`. You can also list the SYSOUT files of a specific job by issuing the `L` line command for the appropriate job in a list of jobs.

The following list of SYSOUT files is the result of the command:

```
LIST S OPDMSEXD
```

The display shows all SYSOUT files for the job `OPDMSEXD`:

```
LIST-JOB-FILES:OPDMSEXD(5025)-------------------- Row 0 of 8 - columns 010
 COMMAND===>                                                SCROLL===> CSR
   FILE NUM PROC     STEP     DD       C RECORDS COP FORM     FCB  WRITER   FLS
** ******************************** top of data ******************************
   JL     1                    $JCL     W       7   1 STD      ****          ***
   SM     1          JES2      $JES2LOG W      13   1 STD      ****          ***
   SM     2          JES2      $JCLIMG  W       0   1 STD      ****          ***
   SM     3          JES2      $SYSMSGS W       5   1 STD      ****          ***
   SI     1 IXMAINT  A         SYSIN    W       1   1 STD      ****          ***
   SO     1 IXMAINT  A         CMDPRINT W       4   1 STD      ****          ***
   SO     2 IXMAINT  A         MSGPRINT W    2595   1 STD      ****          ***
   SO     3 IXMAINT  A         SYSPRINT W       9   1 STD      ****          ***
** ****************************** bottom of data ******************************









Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Meaning of the column headings:

Field	Meaning
FILE	File name.
NUM	File number.
PROC	Procedure being executed (if applicable).
STEP	Job step referencing the file.
DD	DD name.
C	Output class.
RECORDS	Number of records in the file.
COP	Number of file copies.
FORM	Print form.
FCB	Form control buffer.
WRITER	External writer.
FLSH	Form overlay (flash).

Example: BROWSE

The command:

```
BROWSE J COM*
```

displays an active help window with a list of all jobs that start with `COM`. Select a job with any character and press ENTER to display the first SYSOUT file.

Example: RELEASE

The command:

```
RL J 3687 CL=O,NEWCL=Y
```

releases held output of `Job 3687` from Class `O` and assigns new `Output Class Y` (without prompting).

Note: From a list of SYSOUT files, only `SM` and `SO` files can be subject to `RELEASE` operations.

Line Commands

The following line commands are available from lists of jobs and SYSOUT files:

Line Command	Corresponding Function Command	Remarks
B	BROWSE	For SYSOUT files only.
CC	CC	For jobs only.
CH	CHANGE	From a list of jobs or SYSOUT files, change job class (when Q=I) or output class (Q=O) and/or output destination of a JES queue entry. Opens a window in which you can specify a different class and/or destination.* **
CP	COPY	From a list of jobs, prompts for details of a specific spool file to be copied.
EX	EXPORT	From a list of jobs, prompts for details of a specific spool file to be exported.
FL	FOLLOW	For jobs only.
HL	HOLD	For jobs only.
L	LIST	From a list of jobs, lists SYSOUT files.
PG	PURGE	From a list of jobs, purges all SYSOUT. *
PR	PRINT	From a list of jobs, prompts for details of a specific spool file to be printed.

Line Command	Corresponding Function Command	Remarks
RL	RELEASE	From a list of jobs, releases job from HOLD queue (with optional specification of a new output class) or from HOLD status (when Q=I). The RELEASE command has no effect when Q=X or O. Opens a window in which you can specify a different class. *
ST	STATUS	For jobs only.

You can also use line commands as valid abbreviations for function commands entered in the command line of any screen.

* From a list of SYSOUT files, these commands are valid only if the selected file resides in the held output queue.

** From a list of jobs, CH is valid for jobs which are not in HOLD (that is for jobs with Q=O or Q=I). For jobs in the held output queue (Q=H), the command is valid only at sites where JES2.4.2 or a higher version of JES2 is installed.

Local Commands

In Browse Mode:

When displaying a SYSOUT file in Editor format, you can use the following local commands in addition to Editor scroll commands:

Command	Meaning
FILE *X*	Display SYSOUT file number *X* (e.g. FILE 4).
FILE *name*	Display SYSOUT file with name *name* (e.g. FILE SO).
FILE *name X*	Select SYSOUT file by number and name (e.g. FILE SO 4).
FILE AL	Display all SYSOUT files as one file.
NEXT	Display next SYSOUT file.
PREV	Display previous SYSOUT file.

In List Mode:

If you display lists of SYSOUT files, the following local commands are available in addition to Editor scroll commands: ALL, LAYOUT, RELIST and SORT. For detailed information, see the subsections in the section ***Useful Features***.

z/OS System Operations

With the system operations facility, you can perform certain monitoring and administrative functions: you can display the system log, active jobs, the console, lists of units, enqueues, and you can use IDCAMS services and issue operator commands.

To enter the system operations facility, select the `SYSTEM` option from the Natural ISPF Main Menu. The System Facilities Menu appears:

```
----------------------------- SYSTEM FACILITIES -------------------------------
OPTION  ===>
                                                              Userid   MBE
                                                              Time     08:39:41
   1    LOG        - Display system log                       Terminal DAELC521
   2    ACTIVE JOBS- Display address space information        Library  MBE
   3    CONSOLE    - Console display and commands             Node     148

   4    UNITS      - Display unit information
   5    ENQUE      - Display ENQ  information
   6    IDCAMS     - IDCAMS services









Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right :s
```

The System Facilities Menu provides the following options:

- `LOG`:

  display system log;

- `ACTIVE JOBS`:

  display active job information according to selection criteria;

- `CONSOLE`:

  display console and issue operator commands;

- `UNITS`:

list units and display unit information;

- `ENQUE`:

  list queues and dequeues resources, that is, remove locks from them;

- `IDCAMS`:

  use IDCAMS services.

These options are described in more detail in the following subsections.

System Log

If you select the `LOG` option from the System Facilities Menu, the system log is displayed in Editor format in browse mode. The last screen page of the system log appears. You can use all Editor browse commands (`UP`, `DOWN`, `TOP`, `BOTTOM`, `FIND`).

The system log is a separate object in Natural ISPF with object type `LOG`. To invoke the system log from any screen, use the function command

```
BROWSE LOG NODE=id
```

where *id* is the Entire System Server node number of the CPU to be addressed (in multi-CPU environments).

Active Jobs Display

You can access active job information via the System Facilities Menu. Active jobs are also separate Natural ISPF objects with Object Type `ACT` (see the subsection ***Function Commands***).

If you select the `ACTIVE JOBS` option from the System Facilities Menu, the Active Jobs Entry Panel appears:

```
-------------------------ACTIVE-JOBS--ENTRY-PANEL----------------------------
COMMAND ===>

   Job           ===>
   Type          ===>                    ( Combination of: S,T,J,I )
   Only IN       ===>                    ( Y, or blank)
   Node          ===>

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      HELP  SPLIT END   SUSPE RFIND RCHAN UP    DOWN  SWAP  LEFT  RIGHT CURSO
```

Meaning of the input fields:

Field	Meaning
Job	Job name. Use a combination of strings and wildcards (* and _) to generate a selective list of jobs as described in the subsection ***Selection Windows and Wildcards*** in the section *Command Logic*.
Type	Job type. Possible values: I Job entry system initiator (INI) J Standard operating system job (JOB) S Started task (STC) T TSO user (TSU) or any combination (e.g., JT selects all JOB- and TSU-type jobs).
Only IN	Specify Y to display in-memory active jobs only. Leave blank to show all jobs as specified in Job and Type fields.
Node	Optional. Specify Entire System Server node. Enter a question mark (?) and press ENTER to list available nodes. Leave blank to select the default node.

You can type selection criteria in the input fields and enter a function command in the command line.

Function Commands

The following function commands are available for active jobs:

Command	Object Parameter Syntax
BROWSE	*job-name* NODE=*id* [1]
INFORMATION	*job-name* NODE=*id* [2]
LIST	*_ _* TYPE=*t* IN=Y/N NODE=*id*

[1] The BROWSE command selects file SM1 of the job for display. You can use the **local commands** to select another SYSOUT file of the job.

[2] The INFORMATION command lists the loaded modules of the job.

Note: If you issue any of these commands from outside the active jobs facility, you must specify object type ACT before the object parameters in the command syntax.

Below is an example of the LIST command.

Example: LIST command

The following figure shows a list of active jobs generated using the command:

```
LIST ACT * TYPE=JT
```

```
LIST-ACT:*/TYPE=JT ------------------------------ Row 0 of 12 - Columns 001 076
 COMMAND===>                                                   SCROLL===> CSR
    JOBNAME  CLASS  STEP     PROC     TYP JOBNO POS DSP REG   CPU     IO
** ******************************** top of list ******************************
    WKY                               TSU 04952 OUT 255     0 0:15.04      6520
    TKIL                              TSU 04965 OUT 255     0 0:14.67      1317
    SAGTMH                            TSU 04954 OUT 255     0 0:03.96       661
    BGW                               TSU 04975 OUT 255     0 0:01.40       362
    EB                                TSU 04788 OUT 255     0 0:02.99       519
    RAK                               TSU 05006 OUT 255     0 0:02.43       796
    EB1                               TSU 04960 OUT 255     0 0:01.73       652
    SAGT     K      NUC               JOB 04686 N-S  87   240 0:01.30       434
    IMSMAF1H M      STEP01   REGION   JOB 02100 N-S  88   108 0:00.20       289
    NQAN22X  M      NDBM222  REGION   JOB 04887 N-S  82   120 0:00.76       586
** ****************************** bottom of list ******************************









Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

The list shows all `JOB`- and `TSU`-type active jobs. The in-memory jobs appear highlighted in this list.

Meaning of the fields according to column heading:

Column	Meaning	
`JOBNAME`	Job name.	
`CLASS`	JES2 batch initiator class(es) (JOB- or INI-type jobs only).	
`STEP`	Name of step currently executed.	
`PROC`	Name of procedure currently executed.	
`TYP`	Job type (JOB, TSU, STC, INI).	
`JOBNO`	Job number.	
`POS`	`IN`	Job is in-memory.
	`N-S`	Not swappable.
	`OUT`	Job is not in-memory.
`DSP`	Dispatching priority.	
`REG`	Amount of real storage used by address space in Kbytes.	
`CPU`	CPU time consumed by address space in minutes.	
`IO`	Number of I/O operations performed so far.	

The list is refreshed every time you press ENTER.

Line Commands

Line Command	Corresponding Function	Remarks
B	BROWSE	From the list of SYSOUT files (obtained with the L line command), you can select a file for display using the B line command (BROWSE).
I	INFORMATION	The load modules of an active job can be listed using the INFO command from the list of active jobs.
L	LIST	This line command is available on a list of active jobs for listing the SYSOUT files for the job. SYSOUT files of active jobs can be handled as SYS-type objects described in the subsection *z/OS Job Information*.

Example: INFORMATION line command

The list obtained with the I line command appears in Natural ISPF Editor format in BROWSE mode. This means you can use all available BROWSE commands (UP, DOWN, BOTTOM, TOP, FIND, LOCATE). For example, the following list was generated by issuing the I line command for the job XCOM140:

```
LIST-LMOD:XCOM140 ---------------------------------------------- Columns 026 076
 COMMAND===>                                                    SCROLL===> CSR
   MODULE   ADDRESS  LENGTH          TYPE ENTRY    USE STATE   PSW      TCB
** ******************************** top of list ******************************
   XCOMWOPN 000352D8 000510          PRB  000352D8 1   WAIT    00035340 008ED9E
   XCOMSMFW 0008C7F8 000808          PRB  0008C7F8 1   WAIT    0008C9CE 008EDCF
   XCOTABE  00108860 0017A0          LOAD 00108860 1                    008F621
   NATPCMDL 000CC290 0006B0          LOAD 000CC290 1                    008F621
   XCOMV206 00105CF0 001310          LOAD 00105CF0 1                    008F621
   XCOMV203 00103F60 0010A0          LOAD 00103F60 1                    008F621
   XCOMV204 00100E70 002190          LOAD 00100E70 1                    008F621
   XCOMVUSO 000EA028 000578          LOAD 000EA028 1                    008F621
   XCOMV200 000FE990 001670          LOAD 000FE990 1                    008F621
   XCOMV191 000FD6C8 000938          LOAD 000FD6C8 1                    008F621
   XCOMV190 000FB4E8 0008A8          LOAD 000FB4E8 1                    008F621
   XCOMV063 000FBD90 001270          LOAD 000FBD90 1                    008F621
   XCOMVUSI 000EF0A0 0005C0          LOAD 000EF0A0 1                    008F621
   XCOMV046 000F7308 000C80          LOAD 000F7308 1                    008F621
   XCOMV045 000E9090 000578          LOAD 000E9090 1                    008F621
   XCOMV044 000F9508 001AF8          LOAD 000F9508 1                    008F621
   XCOMV043 000F30E0 000D18          LOAD 000F30E0 1                    008F621
   XCOMV042 000F12F0 000908          LOAD 000F12F0 1                    008F621
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Use the RIGHT Editor command (usually assigned to PF11) to scroll right and display the last column for the list.

The meaning of the column headings:

Heading	Meaning
MODULE	Name of load module.
ADDRESS	Load address of module.
LENGTH	Length of module.
TYPE	Module type: LOAD - Module was loaded. PRB - Module is an active program. IRB - Interrupt request. TIRB - Interrupt request. SIRB - Interrupt request. SVRB - Supervisor SVC.
ENTRY	Entry point of module.
USE	Use count of module.
STATE	Run state of program (TYPE is anything except LOAD): WAIT - Program is in wait state. RUN - Program is running. SUSPEND - Program is suspended.
PSW	Current Program Status Word for a program (when TYPE is anything except LOAD).
TCB	Address of Task Control Block.
MOTHER	TCB address of task which attached this task.

Example: BROWSE line command for load modules

You can select load modules from the list using the line command BROWSE which displays the module loaded in memory. The following screen shows the result of the BR line command for module NATPCMDL:

```
BROWSE-LMOD:XCOM140(NATPCMDL) --------------------------------- Columns 001 059
 COMMAND===>                                                    SCROLL===> CSR
   OFFSET DATA                                 TEXT
** ********************** top of list **********************
   000000 47FF0024 D5D7D9C3 E5F1F3F4 D5C1E3D7 NPRCV134 NATP
   000010 C3D4C4D3 F9F460F0 F360F1F8 F1F64BF5 CMDL94-03-1816.5
   000020 F8404040 90ECD00C 18BF50D0 B5E841F0 805Y1022
   000030 B5E450FD 000841D0 B5E418F0 47FFB040 16987312
   000040 47F0B04C 47F0B0FA 47F0B53C 18215010 15778152
   000050 B6604110 B57C4100 B6580A18 86FFB572 11E35874
   000060 9200B672 4110B66C 41000006 D502B654 45901180
   000070 90004770 B07CD600 B6721000 41110001 05197543
   000080 41990003 4600B06C 9180B672 4780B572 35787924
   000090 95409000 4780B0C0 D403B668 B6684770 36897512
   0000A0 B0C00700 41009000 5810B0B0 47F0B0B4 58974587
   0000B0 80000000 0A0812FF 4770B0C0 5000B668 36587502
   0000C0 9110B5B4 4710B572 4510B0D0 8F0CC814 96852110
   0000D0 0A139110 B5B44780 B0F6D403 B664B664 36985743
   0000E0 4770B0F2 4800B5C2 4510B0EC 0A0A5010 A1578492
   0000F0 B6649201 B67347F0 B5725820 B6609180 0F342611
   000100 B6724780 B5389501 B6734770 B5385870 56F67543
   000110 B6645840 6008D201 B674401C D201B676 32587483
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Console Display

If you select the `CONSOLE` option from the System Facilities Menu, the console appears. The following figure shows an example console:

```
BROWSE-CON:/NODE=148/TYPE=ALL ---------------------------------- Columns 001 076
 COMMAND===>                                                     SCROLL===> CSR
    SERV  PG  PAGE  SWAP   VIO SWAPS
   JOB05957  -QTRSRXBP N21SRXBP C           00    371    907    .01    .00
    7485   2     0     0     0     0
   JOB05957  IEF404I QTRSRXBP - ENDED - TIME=14.25.23
   JOB05957  -QTRSRXBP ENDED.  NAME-                     TOTAL TCB CPU TIME=
    TOTAL ELAPSED TIME=    .1
   JOB05957  .HASP395 QTRSRXBP ENDED
    .HASP309    INIT  4 INACTIVE ******** C=K
   JOB05956  -HPFCOB   COB                  12    434   1222    .00    .00
    4640   2     0     0     0     0
   STC05724  .P      J05929
   JOB05929  .HASP608 HPFCOB   AWAITING PURGE            PRIO  1 PURGE ANY
   JOB05956  -HPFCOB   LNK               FLUSH      0      0    .00    .00
      0   2     0     0     0     0
   JOB05929  .HASP250 HPFCOB   IS PURGED
   JOB05956  IEF404I HPFCOB - ENDED - TIME=14.25.25
   JOB05956  -HPFCOB   ENDED.  NAME-                     TOTAL TCB CPU TIME=
    TOTAL ELAPSED TIME=    .2
   JOB05956  .HASP395 HPFCOB   ENDED
    .HASP309    INIT  3 INACTIVE ******** C=K
    IKT100I USERID           CANCELED DUE TO UNCONDITIONAL LOGOFF
** ****************************** bottom of list ******************************
```

You can select information to be displayed by entering one of the following local commands:

Local Command	Meaning
`ALL`	Displays all information.
`LINES` *nnn*	Control the number of lines held in the session (*nnn* <= 999).
`LINES RESET`	Returns to the original number of lines: 1 screen size.
`PEND`	Displays only those lines waiting for an operator reply (`WTOR`).

Note: To monitor your system, you can use the `REFRESH` *n* command to automatically update the screen with the latest console messages. See also ***Automatic Screen Refresh*** in the section *Useful Features*.

The console screen is refreshed every time you press ENTER. You can enter any operator command in the command line if you precede it with the session command `OPERATOR`, usually assigned to the magic character stroke (/):

```
/operator-command
```

If you type a plus sign (+) after the command string, a window opens in which you can type an operator command string of up 80 characters in length. This is useful in the following situations:

- if the command line is too short for the operator command;
- if the command string contains special characters that have a special meaning in Natural ISPF (magic characters, delimiters). These characters are not interpreted as Natural ISPF characters when typed in the operator command window.

Note: The Natural ISPF command `OPERATOR` followed by an operator command can be issued from any system screen.

The console is a separate object within Natural ISPF with object type `CON`. You can access the console from any system screen using the function command `BROWSE` in the following format:

```
BROWSE CON TYPE=ALL/PEND NODE=id
```

where *id* is the Entire System Server node number of the CPU whose console is to be displayed (in multi-CPU environments).

System Units

The `UNITS` option on the System Facilities Menu allows you to list system units and request information on a specific unit.

```
------------------------ SYSTEM UNITS - ENTRY PANEL --------------------------
COMMAND ===>


   Volser        ===>
   Class         ===>                    ( DASD,TAPE )
   Status        ===>                    ( ONLINE,OFFLINE,CHANGE )
   Unit          ===>
   Node          ===>









Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right :s
```

You can specify the volume you wish to access in the input fields and enter a function command in the command line.

Meaning of the input fields:

Field	Meaning
`Volser`	Volume serial number. Enter a volume name to display information about a specific volume. Enter the asterisk wildcard (*) to select all volumes or any volume, or a prefix with an * (for example, `ABC*`) to select all volumes starting with that prefix.
`Class`	Select a specific class of units when generating a list of units. Examples of valid classes are: `COMM`, `CTCA`, `DASD`, `DISP`, `TAPE`, `UREC`.
`Status`	Selection criterion when listing units: all units in the specified status are listed. Valid values are: `CHANGE`, `OFFLINE`, `ONLINE`.
`Unit`	Selection criterion when listing units All units with the specified unit address or prefix (for example, `4*`) are selected.
`Node`	Optional. Specify Entire System Server node. Enter a question mark (?) and press ENTER to list available nodes. Leave blank to select the default node.

System units are separate objects in Natural ISPF with object type `UNI`. This means you can issue the relevant function command from any system screen (see the subsection on function commands below).

Function Commands

The following function commands are available for system units:

Command	Object Parameter Syntax
INFORMATION	VOLSER
LIST	VOLSER CLASS=*c* STATUS=*s* UNIT=*u* NODE=*id*

The INFORMATION command is also available as line command (I) from lists of units generated with the LIST command.

Example: LIST (1)

The following list is the result of the command:

```
LIST UNI VOLSER=* CLASS=DASD UNIT=*
```

```
 LIST-UNI:/VOLSER=*/CLASS=DASD/UNIT=* ----------- Row 0 of 470 - Columns 019 076
  COMMAND===>                                                   SCROLL===> CSR
    CLASS UNIT VOLSER          SERIES STATUS  MOUNT      VOLUME   ACTIVITY  DENS
 ** ******************************** top of list ******************************
    DASD  100  BMC003          3380   ONLINE  RESIDENT   PRIVATE
    DASD  101  BMC004          3380   ONLINE  RESIDENT   PRIVATE
    DASD  102                  3380   OFFLINE
    DASD  103                  3380   OFFLINE
    DASD  104                  3380   OFFLINE
    DASD  105                  3380   OFFLINE
    DASD  106                  3380   OFFLINE
    DASD  107                  3380   OFFLINE
    DASD  108                  3380   OFFLINE
    DASD  109                  3380   OFFLINE
    DASD  10A                  3380   OFFLINE
    DASD  10B                  3380   OFFLINE
    DASD  10C                  3380   OFFLINE
    DASD  10D                  3380   OFFLINE
    DASD  10E                  3380   OFFLINE
    DASD  10F                  3380   OFFLINE
    DASD  110  BMCRES          3380   ONLINE  RESIDENT   PRIVATE
    DASD  111  BMC001          3380   ONLINE  RESIDENT   PRIVATE
 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
       Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right :s
```

The list shows all direct access units.

To display more information, issue the RIGHT command (usually assigned to PF11):

```
LIST-UNI:/VOLSER=*/CLASS=DASD/UNIT=* ----------- Row 0 of 470 - Columns 077 134
 COMMAND===>                                                   SCROLL===> CSR
   CLASS UNIT VOLSER DCB FREE(CYL/TRK) CONTIG(CYL/TRK) FREE-EXT TOT-CYL TRK/CYL
** ******************************** top of list ******************************
   DASD  100  BMC003   1   72 ,   29         60 ,    0        7     886      15
   DASD  101  BMC004   1   14 ,   31          5 ,   14       10     886      15
   DASD  102           0    0 ,    0          0 ,    0        0       0
   DASD  103           0    0 ,    0          0 ,    0        0       0
   DASD  104           0    0 ,    0          0 ,    0        0       0
   DASD  105           0    0 ,    0          0 ,    0        0       0
   DASD  106           0    0 ,    0          0 ,    0        0       0
   DASD  107           0    0 ,    0          0 ,    0        0       0
   DASD  108           0    0 ,    0          0 ,    0        0       0
   DASD  109           0    0 ,    0          0 ,    0        0       0
   DASD  10A           0    0 ,    0          0 ,    0        0       0
   DASD  10B           0    0 ,    0          0 ,    0        0       0
   DASD  10C           0    0 ,    0          0 ,    0        0       0
   DASD  10D           0    0 ,    0          0 ,    0        0       0
   DASD  10E           0    0 ,    0          0 ,    0        0       0
   DASD  10F           0    0 ,    0          0 ,    0        0       0
   DASD  110  BMCRES   1   27 ,   36         11 ,    0       12     886      15
   DASD  111  BMC001   1   81 ,  167         69 ,   14       25     886      15
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right :s
```

Meaning of the column headings:

Column	Meaning
CLASS	Device class. Possible classes are: COMM - Communications CTCA - Channel-to-channel adapter DASD - Direct access DISP - Display station TAPE - Tape UREC - Unit record
UNIT	Unit address in EBCDIC.
VOLSER	Volume serial number currently mounted on the unit.
SERIES	Device series.
STATUS	Device status. Possible values are: CHANGE (status is changing), OFFLINE, ONLINE.
MOUNT	Mount status. Possible values: MOUNT PEND, NOT READY, REMOVABLE, RESERVED, RESIDENT.
VOLUME	Volume status. Possible values: PRIVATE, PUBLIC, STORAGE.
ACTIVITY	Activity of device. Possible values: ALLOCATED, BUSY.

Column	Meaning
DENSITY	Tape density. Possible values: 800, 1600, 6250, 800/1600, 1600/6250.
DCB	Number of DCBs currently open on the unit.
FREE	Number of free cylinders and tracks on disk pack.
CONTIG	Number of cylinders and tracks in largest free extent.
FREE-EXT	Number of free extents on disk pack.
TOT-CYL	Total number of cylinders on disk pack.
TRK/CYL	Number of tracks per cylinder.

Example: LIST (2)

The following list is the result of the command:

```
LIST UNI VOLSER=* CLASS=TAPE UNIT=80*
```

The list shows all tape units with addresses starting with 80. To display more information, issue the RIGHT command (usually assigned to PF11).

```
LIST-UNI:/VOLSER=*/CLASS=TAPE/UNIT=80* ---------- Row 0 of 16 - Columns 019 076
 COMMAND===>                                                SCROLL===> CSR
   CLASS UNIT VOLSER          SERIES STATUS  MOUNT      VOLUME   ACTIVITY  DENS
** ******************************** top of list ******************************
   TAPE  801                  3480   OFFLINE                                800
   TAPE  800                  3480   OFFLINE                                800
   TAPE  802  000673          3480   ONLINE  REMOVABLE                      800
   TAPE  803                  3480   ONLINE  NOT READY                      800
   TAPE  804                  3480   OFFLINE                                800
   TAPE  805                  3480   OFFLINE                                800
   TAPE  806                  3480   OFFLINE                                800
   TAPE  807                  3480   OFFLINE                                800
   TAPE  808                  3480   OFFLINE                                800
   TAPE  809                  3480   OFFLINE                                800
   TAPE  80A                  3480   OFFLINE                                800
   TAPE  80B                  3480   OFFLINE                                800
   TAPE  80C                  3480   OFFLINE                                800
   TAPE  80D                  3480   OFFLINE                                800
   TAPE  80E                  3480   OFFLINE                                800
   TAPE  80F                  3480   OFFLINE                                800
** ****************************** bottom of list ******************************

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right :s
```

For an explanation of the information displayed, see the **first LIST example**.

Example: INFORMATION

The following display is the result of the command:

```
INFO UNI VOLSER=BMC003
```

```
---------------------- VTOC SUMMARY INFORMATION DBA003 ------------------------
COMMAND ===>


    SERIES            :  3380
    UNIT              :  5C3
    VOLUME DATA
      TRACKS/CYL      :       15
      %USED           :       100

                        CYLINDER       TRACKS
      TOTAL SPACE     :     2656        39840
      FREE SPACE      :        0  +         0
      LARGEST         :        0  +         0

      FREE EXTENTS    :        0

      SMS CONTROL     :  NO




Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Most of the information provided by this display repeats the information given for the volume in the list of units (see **the example for DASD units**), except that here, the usage of space is shown as a percentage of total space.

Note: Some information displayed in the above screen is available only in z/OS environments and not in z/VSE environments.

Local Commands

When displaying a list of units in Editor format, you can issue the following local commands from the Editor command line in addition to scroll commands: `ALL`, `LAYOUT`, `RELIST` and `SORT`. For detailed information, see the subsections in the section ***Useful Features***.

System Enqueues

The `ENQUE` option on the System Facilities Menu allows you to list and delete system enqueues.

```
------------------------- SYSTEM ENQUE ENTRY PANEL ----------------------------
COMMAND ===>


   Queue          ===>
   User           ===>
   Resource       ===>
   Job Name       ===>
   Node           ===>









Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right :s
```

You can specify the enqueue you wish to access in the input fields and enter a function command in the command line.

Meaning of the input fields:

Field	Meaning
`Queue`	Major name of the resource, which is usually the purpose of the enqueue (for example, `SPFEDIT`).
`User`	Name of the user who holds the queue. Can be used as selection criterion when listing enqueues: enter the asterisk wildcard (*) to select all users or a prefix followed by an asterisk (for example, `ABC*`) to select all users beginning with that prefix.
`Resource`	Minor name of the resource, which is usually the object of the enqueue (for example, a dataset name). Can be used as selection criterion when listing enqueues with the asterisk wildcard (*) (see `User` field).

Field	Meaning
`Job Name`	Name of the job that holds the queue. Can be used as selection criterion when listing enqueues with the asterisk wildcard (*) (see `User` field).
`Node`	Entire System Server node. Enter a question mark (?) and press ENTER to list available nodes. Leave blank to select the default node.

System enqueues are separate objects in Natural ISPF with object type `ENQ`. This means you can issue the relevant function command from any system screen, provided you have appropriate authorization to access Object Type `ENQ`.

Function Commands

The following function commands are available for system enqueues:

Command	Object Parameter Syntax
`DELETE`	Not applicable: available as line command only.
`LIST`	`QUEUE USER=`*`u`* `JOB=`*`j`* `NODE=`*`id`*

The `DELETE` command is available only as line command (`D`) from lists of enqueues generated with the `LIST` command. The `DELETE` command "dequeues" an enqueue, that is, the resource is released.

Example: LIST

The following list is the result of the command:

```
LIST ENQ SPFEDIT* JOB=XCOM148
```

```
LIST-ENQ:SPFEDIT*/JOB=XCOM148 ------------------- Row 0 of 33 - Columns 069 076
 COMMAND===>                                                SCROLL===> CSR
   JOBNAME  QUEUE    RESOURCE-NAME                                   USERID
** ******************************** top of list ******************************
   XCOM148  SPFEDIT  ADABAS.SYSF.QAS.KM.KM02.V600.SMP(KM020000)       KM
   XCOM148  SPFEDIT  ADABAS.SYSF.QAS.KM.KM02.V600.SMP(KM020073)       KM
   XCOM148  SPFEDIT  ADABAS.V61.DEP.SOURCE(SVCMVS)                    RR
   XCOM148  SPFEDIT  SUG.CPS.SRC(MAINC)                               SUG
   XCOM148  SPFEDIT  SUG.CPS.SRC(SECLOAD)                             SUG
   XCOM148  SPFEDIT  KSI.JCL.ALL(KSIRPC1)                             KSI
   XCOM148  SPFEDIT  RR.SYSF.SOURCE(ADAREP)                           RR
   XCOM148  SPFEDIT  RR.SYSF.SOURCE(ADA7200)                          RR
   XCOM148  SPFEDIT  ADABAS.V53.DEP.SOURCE(#GEN)                      RR
   XCOM148  SPFEDIT  KHK.SOURCE(NA22PR32)                             KHK
   XCOM148  SPFEDIT  TST.NAT228.JCL(NATSRCEV)                         DWI
   XCOM148  SPFEDIT  ADASQL.PROBLEM(CEXAM1)                           RSH
   XCOM148  SPFEDIT  HEB.SOURCE(COOR-IMP)                             HEB
   XCOM148  SPFEDIT  COM.RDC.V34.SYSTEM(RDCFRT)                       SAGAWW
   XCOM148  SPFEDIT  ADASQL.TEST.SOURCE(GOALC)                        RSH
   XCOM148  SPFEDIT  COM.SYSF.SAGAWW.SOURCE(VSAMPRT)                  SAGAWW
   XCOM148  SPFEDIT  AVB.V331.SOURCE(MCTAB)                           WGL
   XCOM148  SPFEDIT  NISPF.IV211.DV01.MVSSRCE(NATZAP)                 JWO
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Scroll right to display the STATUS column.

Meaning of the column headings:

Column	Meaning
JOBNAME	The job that enqueued.
QUEUE	Major name of the resource (for example, SPFEDIT). Names with the prefix SYS are reserved for z/OS internal purposes.
RESOURCE-NAME	Minor name of the resource (for example, dataset name if QUEUE is SPFEDIT).
USER	User that holds the enqueue.
STATUS	Status of ownership. Possible values: OWNS - User is owner of resource. WAIT - User is waiting for resource.

You can release a resource for use by other users with the DELETE command (D line command).

Local Commands

When displaying a list of enqueues in Editor format, you can issue the following local commands from the Editor command line in addition to scroll commands: `ALL`, `LAYOUT`, `RELIST` and `SORT`. For detailed information, see the subsections in the section ***Useful Features***.

IDCAMS Services

The IDCAMS services option on the System Facilities Menu allow you to use the interactive execution of IBM's IDCAMS utility program. If you select this option, the following window opens:

```
----------------------------- SYSTEM FACILITIES -------------------------------
OPTION  ===> 6
                                                                Userid   MBE
                                                                Time     08:58:50
    +-------------------------------------------------------------------+ELC521
    !                                                                     !E
    !                        VSAM services                                !8
    !                                                                     !
    ! Select function:     LL -- Listcat Level(level)                    !
    !                      LE -- Listcat Entry(name)                     !
    ! Name or level  :                                                   !
    ! Catalog Name   :                                                   !
    ! List options   : S   ( S for short, D for detail)                  !
    !                                                                     !
    ! Direct Access Method Services Command:                             !
    !  ____________________________________________________________      !
    !  ____________________________________________________________      !
    !  ____________________________________________________________      !
    !  ____________________________________________________________      !
    +-------------------------------------------------------------------+


Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right :s
```

The fields in this screen speak for themselves, but for more details on VSAM services, consult the following IBM publications:

- z/OS Extended Architecture, VSAM Catalog Administration: Access Method Services Reference, GC26-4136-4
- z/OS Extended Architecture, Integrated Catalog Administration: Access Method Services Reference, GC26-4135-5

9 CA Panvalet Members

- CA Panvalet Overview 266
- Function Commands 268
- Line Commands 271
- Local Commands 272
- Handling CA Panvalet Control Cards 272

CA Panvalet Overview

The CA Panvalet facility enables access to CA Panvalet members, which you can maintain using Natural ISPF functions.

If the member consists of job control, you can make use of the Natural ISPF macro facility. You can use all types of macro statements. Macro expansion is performed at submission time (see the `SUBMIT` command below). When creating a new member, you can also use the Edit macro feature to automatically create text lines which can then be modified. For details on the macro facility, see the section *Macro Facility* in the *Natural ISPF Programmer's Guide*.

To enter the CA Panvalet facility, select the `PANVALET` option from the Natural ISPF Main Menu. The CA Panvalet Entry Panel appears:

```
-------------------------- PANVALET - ENTRY PANEL -----------------------------
COMMAND ===>


   Data Set Name ===>
   Member        ===>
   Language      ===>
   User          ===>
   Status        ===>
   Volume        ===>                        ( If not catalogued     )
   Password      ===>                        ( If password protected )
   Expand        ===>                        ( Expand ++include      )
   Comment       ===>
   Scan for      ===>
   Edit macro    ===>
   Node          ===> 148





Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right TOP
```

You can specify the CA Panvalet member you wish to maintain in the input fields and enter a function command in the command line.

Meaning of the input fields:

Field	Meaning
`Data Set Name`	Name of library in which the member resides. This field contains the name of the library last accessed. You can select any other library by overtyping the name.
`Member`	Member name (up to ten alphanumeric characters). Leave blank or use strings and wildcards (* and _) to generate a more selective list of member names. See the subsection ***Selection Windows and Wildcards*** in the section *Command Logic*.
`Language`	Language (format) of member, for example, JCL, ASMB (Assembler), COBOL, PL/1, etc.
`User`	User code (up to four numeric digits).
`Status`	Status of member. Any one or a combination of the following are possible: First character: `T` or `P` - Test/Production Second character: `A` or `I` - Active/Inactive Third character: `E` or `D` - Enabled/Disabled
`Volume`	Volume serial number. Required only for uncataloged datasets.
`Password`	Appropriate access code if dataset is protected.
`Expand`	Specifies expansion of `++INCLUDE` statements when using any of the following commands: `BROWSE`, `SUBMIT`, `PRINT`, `COPY`, `INFORMATION`, `EXPORT`. Possible options: `Y` Perform expansion. `N` No expansion is performed.
`Comment`	Comment to appear in list of members and in the information window. You normally add a comment describing the member when creating a new one. If you type a plus sign + as last character of the comment, a window opens in which you can extend the comment text.
`Scan for`	Selection criterion for listing CA Panvalet members: all members as specified in the above fields are listed which contain the string entered here. When you select a member from this list for `EDIT` or `BROWSE`, the cursor is placed on the first occurrence of this string in the member. Issue the `RFIND` command to find the next occurrence.
`Edit macro`	Name of macro object to be used as a model for the member. The specified macro is executed and loaded into the Editor. See the section *Macro Facility* in the *Natural ISPF Programmer's Guide* for details.
`Node`	Select Entire System Server node. Enter a question mark (?) and press ENTER to open a window in which all node numbers appear with an `ACTIVE` or `INACTIVE` status report. If you do not specify a node, the default node is assumed.

For detailed information on CA Panvalet member characteristics, see the appropriate CA Panvalet documentation.

Apart from the `Data Set Name` and `Member` fields, some other parameter fields may be mandatory when creating a new member with the `EDIT` function command. Which fields are required and

which are optional is specified in the system configuration (see the *Natural ISPF Administration Guide*, or ask your system administrator). Natural ISPF prompts you for required items.

Natural ISPF provides the session command `CONTROL` for CA Panvalet users.

`CONTROL OFF` suppresses the automatic control cards passed to CA Panvalet when a member is saved. Only the edited data (which may include user-defined control cards) are sent to CA Panvalet.

The command format is:

```
CONTROL  [ON]
         [OFF]
```

Meaning of the parameters:

Parameter	Meaning
ON	Re-activates the automatic control cards (default).
OFF	Suppresses the automatic control cards. You can define your own control cards in the member.

You must issue the `CONTROL` command before opening the new Editor session.

Function Commands

The available function commands for CA Panvalet members are:

Command	Parameter Syntax
BROWSE	*dataset*(*member*) VOL=*v* PASSWORD=*p* EXPAND=Y/N NODE=*id*
COPY	*dataset*(*member*) VOL=*v* PASSWORD=*p* EXPAND=Y/N NODE=*id*, *object-type object-parameters*, REP
DELETE	*dataset*(*member*) VOL=*v* PASSWORD=*p* NODE=*id*
EDIT	*dataset*(*member*) LANG=*l* USER=*u* VOL=*v* PASSWORD=*p* MACRO=*name* NODE=*id*
EXPORT	*dataset*(*member*) VOL=*v* PASSWORD=*p* EXPAND=Y/N NODE=*id*, *target-environment*
INFORMATION	*dataset*(*member*) VOL=*v* PASSWORD=*p* EXPAND=Y/N NODE=*id*
LIST	*dataset*(*_*) LANG=*l* USER=*u* ST=*s* VOL=*v* PASSWORD=*p* MACRO=*name* NODE=*id*
OUTPUT	NODE=*id*
PLAY	*dataset*(*member*) VOL=*v* PASSWORD=*p* NODE=*id*
PRINT	*dataset*(*member*) VOL=*v* PASSWORD=*p* EXPAND=Y/N NODE=*id*, *printer-name* CC
RENAME	*dataset*(*member*) VOL=*v* PASSWORD=*p* NODE=*id*, *new-name*
SUBMIT	*dataset*(*member*) VOL=*v* PASSWORD=*p* EXPAND=Y/N NODE=*id1*, TARGET=*id2*

A full description of these commands is contained in the section *Command Reference*. The object parameters correspond to the input fields on the CA Panvalet Entry Panel.

Note: If you issue any of the above function commands from outside the CA Panvalet facility, you must specify the object-type parameter `PAN` before the object parameters.

Below are some examples of function commands using full command syntax.

Example: INFORMATION

The following screen is displayed as a result of the function command

```
INFORMATION PAN SYSA.PANLIB.OPT(MYMEM)
```

A window opens containing information on member `MYMEM` in library `SYSA.PANLIB.OPT`:

```
--------------------------- PANVALET - ENTRY PANEL-----------------------------
COMMAND ===>


   Data Set Name ===> MBE.COMN.SOURCE
   Member        ===>
   Language      ===>
   User      +-------------------------------------------------------------+
   Status    ! INFORMATION-PANVALET                                        !
   Volume    ! DSName  : SYSA.PANLIB.OPT                                   !
   Password  ! Member  : MYMEM                                             !
   Expand    ! Language: ASMB                                              !
   Comment   ! Status  : T   (Test/Production)                             !
   Edit macr !           A   (Active/Inactive)                             !
   Node      !           E   (Enabled/Disabled)                            !
             ! User    : 0007                                              !
             ! Level   :  24                                               !
             ! Comment : NISPF TEST PROGRAM                                !
             !-------------------------------------------------------------+


Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

The fields in the information window correspond to the fields on the CA Panvalet Entry Panel, except the `Level` field.

`Level` is the modification level of the member. When a member is created, it is assigned Level 001. The level number is increased by 1 at each update (see the appropriate CA Panvalet documentation for more information).

You can use the `INFORMATION` command to change certain characteristics of the member. Simply overtype the current value with the new value in the information window. The fields `Member`, `Language`, `Status`, `User`, `Level` and `Comment` are all modifiable.

Example: LIST

The following list of CA Panvalet members is displayed as the result of the command:

```
LIST PAN SYSA.PANLIB.OPT(ISP*)
```

The list contains all members starting with `ISP` in the library `SYSA.PANLIB.OPT`:

```
 LIST-PAN:SYSA.PANLIB.OPT(ISP*)------------------ Row 0 of 12 - columns 010 076
  COMMAND===>                                                  SCROLL===> CSR
    MEMBER             LVL USER LANG   STA MAINTAIN ACCESSED  BLKS STATMTS COMME
 ** ******************************** top of list ******************************
    ISPJCL1234           1 2223 JCL    TAE 21/06/90 21/06/90     1      22 Insta
    ISPPROG1             2 2223 ASMB   TAE 21/06/90 21/06/90     1     107 Test
    ISPPROG2
 1 1420 COBOL  TAE 22/06/90 22/06/90     1      17 Progr
    ISPSYSZ              1 2223 DATA   TAE 22/06/90 22/06/90     1       5 Text

 ** ****************************** bottom of list ******************************









 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
       Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Meaning of the column headings:

Column	Meaning
`MEMBER`	Name of member
`LVL`	Modification level of member
`USER`	User code
`LANG`	Programming language of member
`STA`	Status of member:

Column	Meaning
	A Active I Inactive E Enabled D Disabled T Test P Production
MAINTAIN	Date of last change to member text and/or characteristics (for example, user code, comment)
ACCESSED	Date of last access in read or write mode
BLKS	Number of blocks occupied by member
STATMTS	Number of lines in the member
COMMENT	Comment describing member

To display the whole comment column, issue the Editor RIGHT command (usually assigned to PF11).

Note: The modification level, user code, language, status and comment are modifiable in the information window invoked using the INFORMATION function command (see **the example of the INFORMATION command**).

Line Commands

You can select a CA Panvalet member from a list by typing a line command in the input field preceding the member name and pressing ENTER. Each line command is an abbreviation of a function command. The available line commands are:

Line Command	Corresponding Function Command
BR	BROWSE
CP	COPY
D	DELETE
E	EDIT
EX	EXPORT
I	INFORMATION
OT	OUTPUT
PL	PLAY
PR	PRINT

Line Command	Corresponding Function Command
R	RENAME
SB	SUBMIT

Line commands can also be used as valid abbreviations of function commands entered in the command line.

Local Commands

In Edit Mode:

If you display a CA Panvalet member in Editor format in EDIT mode, you can issue a local command from the Editor command line in addition to Editor commands.

The following local command is available:

Command	Meaning
IMPORT	Imports a PC file or Con-nect document into the CA Panvalet member (see the section ***Useful Features***).

In List Mode:

If you display lists of CA Panvalet libraries or members in Editor format, you can issue the following local commands in addition to Editor scroll commands: ALL, LAYOUT, RELIST and SORT. For detailed information, see the corresponding subsections in the section ***Useful Features***.

Handling CA Panvalet Control Cards

Natural ISPF provides some features for flexible handling of CA Panvalet control cards:

- Using the expand feature, you can specify whether the ++INCLUDE statements in the member are expanded before browsing, printing, submitting, copying, exporting the member, or displaying information;
- With the CONTROL OFF session command, you can submit the edited member to PAM#1 using the Editor SAVE command without the automatic control cards. Only the edited text, including any control cards you have added in the member are processed. You can use the OUTPUT function command to see the output of the member as processed.

10 CA Librarian Members

- CA Librarian Members Overview 274
- Function Commands 348
- Line Commands 280
- Local Commands 281
- Handling CA Librarian Control Cards 282
- Previous Versions 282

CA Librarian Members Overview

The Librarian facility enables access to CA Librarian members, which you can maintain using Natural ISPF functions.

If the member consists of job control, you can make use of the Natural ISPF macro facility. You can use all types of macro statements. Macro expansion is performed at submission time (see the `SUBMIT` command below). When creating a new member, you can also use the Edit macro feature to automatically create text lines which can then be modified. For details on the macro facility, see the section *Macro Facility* in the *Natural ISPF Programmer's Guide*).

To enter the Librarian facility

- Select the `LIBRARIAN` option from the Natural ISPF Main Menu.

  The Librarian Entry Panel appears:

```
------------------------- LIBRARIAN - ENTRY PANEL ---------------------------
COMMAND ===>


   Dataset Name ===>
   Member       ===>
   Member pswd  ===>
   Programmer   ===>
   Language     ===>
   Expand       ===>                        ( Automatic expand of include )
   Record length ===>                       ( If different from 80        )
   Version      ===>                        ( YYMMDD,xxxxx,or -yyy        )
   Volume       ===>                        ( If not catalogued           )
   DS Password  ===>                        ( If password protected       )
   Scan for     ===>
   Edit macro   ===>
   Node         ===> 148
```

  You can specify the CA Librarian member you wish to maintain in the input fields and enter a function command in the command line.

Meaning of the input fields:

Field	Meaning
Dataset Name	Name of library in which the member resides. This field contains the name of the library last accessed. You can select any other library by overtyping the name.
Member	Member name (up to eight alphanumeric characters). Leave blank to list all members or use strings and wildcards (* and _) to generate a more selective list of member names. See the subsection ***Selection Windows and Wildcards*** in the section *Command Logic*.
Member pswd	Member password (up to four alphanumeric characters). The following strings are reserved for CA Librarian and cannot be used as member password: BYPP, EXEC, FILL, LIST, NONE, NOPC, PERM, TEMP, TEST (BYPP is the default password if you do not assign one).
Programmer	Name of programmer for the member.
Language	Language (format) of member, for example: JCL, ASM (Assembler), COBOL, PL/1, etc.
Expand	Specifies expansion of -INC statements when using any of the following commands: BROWSE, SUBMIT, PRINT, COPY, EXPORT. Possible options: 0: Displays -INC statements without expanding them. 1: Displays -INC statements at all levels of nesting, together with the associated data records. 2: Displays -INC statements together with the associated data records. Any nested -INC statements are expanded but not displayed. Associated data is displayed, but the INC statements are displayed only if they are unexpandable. 3: Expands encountered -INC statements and the associated data of statements at all levels of nesting. All -INC statements are displayed only if unexpandable. N: Synonym for 0 (default if selected function is EDIT) Y: Synonym for 3 (default if selected function is not EDIT) (blank) Specifies default value.
Record length	Specify only if different from 80.
Version	Specify the required version of the member, if different from the current version. You can specify a version by date in the format *YYMMDD*, you can specify an absolute version by its number, or you can specify a version by its relative distance from the current version, for example: -12 means the twelfth version back from the current version. See the subsection ***Previous Versions*** for more details.
Volume	Volume serial number. Required only for uncataloged datasets.
DS password	Dataset password, if the library is protected.
Scan for	Selection criterion for listing CA Librarian members: all members as specified in the above fields are listed which contain the string entered here. When you select a member from this list for EDIT or BROWSE, the cursor is placed on the first occurrence of this string in the member. Issue the RFIND command to find the next occurrence.

Field	Meaning
`Edit macro`	Name of macro object to be used as a model for the member. The specified macro is executed and loaded into the Editor. See the section *Macro Facility* in the *Natural ISPF Programmer's Guide* for details.
`Node`	Select Entire System Server node. Enter a question mark (?) and press ENTER to open a window in which all node numbers appear with an `ACTIVE` or `INACTIVE` status report. If you do not specify a node, the default node is assumed.

For detailed information on CA Librarian member characteristics, see the appropriate CA Librarian documentation.

Natural ISPF provides a special `CONTROL` session command for CA Librarian users.

`CONTROL OFF` suppresses the automatic control cards passed to CA Librarian when a member is saved. Only the edited data (which may include user-defined control cards) are sent to CA Librarian.

The command format is:

```
CONTROL  [ON]
         [OFF]
```

Meaning of the parameters:

Parameter	Meaning
`ON`	Re-activates the automatic control cards.
`OFF`	Suppresses the automatic control cards. You can define your own control cards in the member.

You must issue the `CONTROL` command before opening the new Editor session.

Function Commands

The following function commands are available for CA Librarian members.

Command	Parameter Syntax
`BROWSE`	*dataset*(*member*) PSWD=*p* EXPAND=Y/N VERSION=*x* VOL=*v* NODE=*id*
`COPY`	*dataset*(*member*) PSWD=*p* VOL=*v* EXPAND=Y/N NODE=*id*, *object-type* *object-parameters*, REP
`DELETE`	*dataset*(*member*) PSWD=*p* VOL=*v* NODE=*id*
`EDIT*`	*dataset*(*member*) PSWD=*p* LANG=*l* USER=*u* VERSION=*x* VOL=*v* MACRO=*name* NODE=*id*
`EXPORT`	*dataset*(*member*) PSWD=*p* VOL=*v* EXPAND=Y/N NODE=*id*, *target-environment*
`INFORMATION`	*dataset*(*member*) PSWD=*p* VOL=*v* EXPAND=Y/N NODE=*id*

Command	Parameter Syntax
LIST	*dataset*(*_*) PSWD=*p* PGMR=*name* LANG=*l* RECLEN=*r* VOL=*v* MACRO=*name* NODE=*id*
OUTPUT	NODE=*id*
PLAY	*dataset*(*member*) PSWD=*p* VOL=*v* NODE=*id*
PRINT	*dataset*(*member*) PSWD=*p* VOL=*v* EXPAND=Y/N NODE=*id*, *printer-name* CC
RENAME	*dataset*(*member*) PSWD=*p* VOL=*v* NODE=*id*, *new-name*
SUBMIT	*dataset*(*member*) PSWD=*p* VOL=*v* EXPAND=Y/N NODE=*id1*, TARGET=*id2*

* If you request an edit session with a new member, you may be required to type a description in a prompt window before you enter the Editor session. If you request an edit session with an existing member, you may be required to enter a reason for modifying the member in a window before you enter the Editor session. Whether either of these prompt windows appears depends on how the CA Librarian interface has been set up (ask your system administrator).

Note: If you issue any of the above function commands from outside the Librarian facility, you must specify the object-type parameter LIB before the object parameters.

A full description of these commands is contained in the section *Command Reference*. The object parameters correspond to the input fields on the Librarian Entry Panel (the keyword parameter PSWD corresponds to the member password). Below are some examples of function commands using full command syntax:

Example: INFORMATION

The following window opens as a result of the function command:

```
INFORMATION LIB LIB1.MASTER(ASMPROGX)
```

```
 -------------------------- LIBRARIAN - ENTRY PANEL ----------------------------
 COMMAND ===> INFORMATION LIB LIB1.MASTER(ASMPROGX)

          +--------------------INFORMATION-LIBRARIAN---------------------+
    Data  !                                                              !
    Membe ! DSName  : LIB1.MASTER                                        !
    Membe ! Created :   /  /  -  .                                       !
    Progr ! Archive : YES   Levels: 020                                  !
    Langu !                                                              !
    Expan ! Member  : ASMPROGX                                           ! e )
    Recor ! Pswd    : DCMB                                               !   )
    Versi ! Updated : 94/11/14 10:20:45                                  !   )
    Volum ! Lang    : ASM                                                !   )
    DS Pa ! Records :    33                                              !   )
    Scan  ! Blocks  :     1                                              !
    Edit  ! Pgmr    : JOSSI                                              !
    Node  ! Desc.   : NISPF TEST PROGRAM MVS/XA                          !
          !                                                              !
          +--------------------------------------------------------------+



 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
       Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Meaning of the fields:

Field	Meaning
DSName	Dataset name
Created	Date dataset was created
Archive	Archiving active for dataset YES or NO
Levels	Maximum number of possible levels for archiving
Member	Member name
Pswd	Member password
Updated	Date and time of last member update in the format: *YY/MM/DD-HH:MM:SS*
Lang	Language code of member
Records	Number of lines in member
Blocks	Number of blocks occupied by member
Pgmr	Name of programmer for member
Desc.	Description of member

The programmer name and description are modifiable in this window. You can overtype the current values with the new ones.

Example: LIST

The following list of CA Librarian members is displayed as the result of the command:

```
LIST LIB LIB1.MASTER(A*)
```

The list shows all members starting with `A` in the library `LIB1.MASTER`.

```
LIST-LIB:LIB1.MASTER(A*)----------------------- Row 0 of 12 - columns 010 076
COMMAND===>                                                   SCROLL===> CSR
    MEMBER   PSWD          LAST UPDATE       LNG DESCRIPTION
 ** ******************************** top of list ********************************
    ADAEX2T1 PXTX          91/11/21-07:57:51 ASS 'ADABAS - TEST1 PLOG EXIT'
    ADAPL    WVDS          91/11/21-07:56:12 JCL 'ADABAS - PROCEDURE FOR ADARES
    ALTPP    XHJG          94/01/25-13:06:49 ASM 'ALTPP'
    ALT1D    RSLD          94/01/30-14:01:47 ASM ALT TABLE FUER CICSD
    ALT1F    QPPF          94/06/23-09:20:07 ASM ALT FOR CICSY1F1
    ALT1S    CMNP          94/01/29-16:31:24 ASM 'ALT FUER CICSY1S1 '
    ALT1T    LZSH          94/01/30-14:02:38 ASM ALT FUER CISY1T1
    ALT1Z    HHRG          94/01/30-14:04:02 ASM ALT FUER CISY1Z1
    ALT2A    NXLJ          94/01/30-14:04:56 ASM 'ALT2A'
    ALT2P    XKSF          94/01/30-14:06:20 ASM ALT FUER CICSY2P1
    ALT2R    KJLB          94/01/30-14:07:29 ASM 'ALT FUER CICSY2R1'
    ALT2W    STZN          94/01/30-14:09:26 ASM 'CICSTAB'
    ASMHCL   WDBS          93/03/19-11:54:30 PRO 'FROM SYS1.PROCLIB'
    AUSX5    KJFQ          94/09/02-15:42:35
 ** ****************************** bottom of list ******************************

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

To display the whole description column

- Issue the Editor `RIGHT` command (usually assigned to PF11):

  The following screen appears:

```
LIST-LIB:LIB1.MASTER(A*)--------------------------- Row 0 of 12 - Columns 010 076
 COMMAND===>                                                   SCROLL===> CSR
     MEMBER   PSWD LNG DESCRIPTION                      RECS BLKS  PROGRAMMER
  ** ******************************** top of list ******************************
     ADAEX2T1 PXTX ASS 'ADABAS - TEST1 PLOG EXIT'        114     1 JECKE
     ADAPL    WVDS JCL 'ADABAS - PROCEDURE FOR ADARES     16     1 JECKE
     ALTPP    XHJG ASM 'ALTPP'                            28     1 JECKE
     ALT1D    RSLD ASM ALT TABLE FUER CICSD               38     1 WOLFF
     ALT1F    QPPF ASM ALT FOR CICSY1F1                   38     1 WOLFF
     ALT1S    CMNP ASM 'ALT FUER CICSY1S1 '               39     2 ZTS0060
     ALT1T    LZSH ASM ALT FUER CISY1T1                   39     1 ZTS0040
     ALT1Z    HHRG ASM ALT FUER CISY1Z1                   39     1 ZTS0060
     ALT2A    NXLJ ASM 'ALT2A'                            38     1 SCHWARZ
     ALT2P    XKSF ASM ALT FUER CICSY2P1                  39     1 ZTS0040
     ALT2R    KJLB ASM 'ALT FUER CICSY2R1'                38     1 SCHWARZ
     ALT2W    STZN ASM 'CICSTAB'                          38     1 JECKE
     ASMHCL   WDBS PRO 'FROM SYS1.PROCLIB'                23     1 JECKE
     AUSX5    KJFQ                                       303     4 ZTS0080
  ** ****************************** bottom of list ******************************

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

The meaning of the fields corresponds to the fields in the information window: the list shows the member name, member password, date and time of the last update, number or records in the member, number of blocks occupied, programmer name, language code and the description.

Note: The programmer name and description are modifiable in the information window invoked using the `INFORMATION` function command (see **the example of the `INFORMATION` command**).

Line Commands

You can select a CA Librarian member from a list by typing a line command in the input field preceding the member name and pressing ENTER. Each line command is an abbreviation of a function command. The available line commands are:

Line Command	Corresponding Function Command
BR	BROWSE
CP	COPY
D	DELETE
E	EDIT
EX	EXPORT
I	INFORMATION

Line Command	Corresponding Function Command
L	LIST (previous versions of the member)
OT	OUTPUT
PL	PLAY
PR	PRINT
R	RENAME
SB	SUBMIT

Line commands can also be used as valid abbreviations of function commands entered in the command line of any system screen.

Local Commands

In Edit Mode:

If you display a CA Librarian member in Editor format in EDIT mode, you can issue a local command from the Editor command line in addition to Editor commands.

The following local command is available:

Command	Meaning
IMPORT	Imports a PC file or Con-nect document into the CA Librarian member (see the subsection ***Natural Interface to External Environments*** in the section *Useful Features*).
PASSWORD *password*	If the library is password-protected, use this command to enter the valid dataset password in order to update the edited member. If you enter the PASSWORD command without parameter, a window prompts you for the password. Password input in the window is invisible.

In List Mode:

If you display lists of CA Librarian members in Editor format, you can issue the following local commands in addition to Editor scroll commands: ALL, LAYOUT, RELIST and SORT. For detailed information, see the corresponding subsections in the section ***Useful Features***.

Handling CA Librarian Control Cards

Natural ISPF provides some features for flexible handling of CA Librarian control cards:

- Using the Expand feature, you can specify whether the `-INC` statements in the member are expanded before browsing, printing, submitting, copying, exporting the member, or displaying information;
- With the `CONTROL OFF` session command, you can submit the edited member to CA Librarian using the Editor `SAVE` command without the automatic control cards. Only the edited text, including any control cards you have added in the member are processed. You can display the output of the member as processed using the `OUTPUT` function command.

Previous Versions

Previous versions of CA Librarian can be kept and retrieved using Natural ISPF.

Listing Previous Versions

You can list previous versions of a CA Librarian member by either:

- Issuing the `LIST` function command from any system screen, specifying library and member in the command syntax, or:
- Selecting a member from a list of members using the `L` line command.

The following screens show an example of a list of previous versions of a CA Librarian member. The function command used to generate the list was:

```
LIST LIB LIB1.MASTER(ALTIT)
```

The second screen shows the result of a `RIGHT` scroll (usually performed using PF11).

```
LIST-LIB:LIB1.MASTER(ALTIT)-------------------- Row 0 of 12 - columns 010 076
 COMMAND===>                                                 SCROLL===> CSR
    LEVEL          PSWD LAST UPDATE     LNG DESCRIPTION                     RECS
 ** ********************************* top of list ******************************
        4          LZSH 94/01/30-140238 ASM ALT FUER CISY1T1                  39
        3          LZSH 94/01/13-172055 ASM ALT FUER CISY1T1                  30
        2          LZSH 93/05/26-185907 ASM ALT FUER CISY1T1                  19
        1          LZSH 93/05/26-185118 ASM ALT FUER CISY1T1                  19
        0          LZSH 93/05/26-184920 ASM ALT FUER CISY1T1                  19
 ** ****************************** bottom of list ******************************
```

```
LIST-LIB:LIB1.MASTER(ALTIT)-------------------- Row 0 of 12 - columns 010 076
COMMAND===>                                                  SCROLL===> CSR
    LEVEL ATE     LNG DESCRIPTION                     RECS  BLKS PROGRAMMER
 ** ******************************** top of list ******************************
        4 -140238 ASM ALT FUER CISY1T1                  39     1 ZTS0040
        3 -172055 ASM ALT FUER CISY1T1                  30     1 ZTS0040
        2 -185907 ASM ALT FUER CISY1T1                  19     1 ZTS0040
        1 -185118 ASM ALT FUER CISY1T1                  19     1 ZTS0040
        0 -184920 ASM ALT FUER CISY1T1                  19     1 ZTS0040
 ** ****************************** bottom of list ******************************
```

This example shows that there are five versions of the member, Version 4 being the current version. You can select any version for edit or browse using the `E` or `B` line command (see also next subsection).

Retrieving Previous Versions

You can retrieve a previous version of a CA Librarian member in any of the following ways:

- Generate a list of previous versions and use an appropriate line command; possible line commands are `E` (`EDIT`), `B` (`BROWSE`), `CP` (`COPY`).
- Specify a library and member in the CA Librarian Entry Panel, entering data in the `Version` field to identify the required version. This can be a date in the format *`YYMMDD`* to identify the version created on that day, an absolute version number (as it appears in the list of versions), or a relative notation (for example, `-5` selects the fifth version back from the current version.

  Note: If you do not type in a function command, `EDIT` is the default. If you do not specify a version, the current version is the default.

- Enter function command syntax from any Natural ISPF screen to retrieve a version, using the `VERSION` parameter. The available function commands are `EDIT` and `BROWSE`, for example:

```
EDIT LIB LIB1.MASTER(ALTIT) VERSION=4
```

If you edit and save a previous version, it automatically becomes the current version.

11 z/VSE Objects

- z/VSE Members 286
- z/VSE File Maintenance 295
- z/VSE Job Information 305
- z/VSE System Operations 310

This chapter explains all the functions you can perform on objects only available in a z/VSE environment.

z/VSE Members

In the z/VSE members maintenance facility, you can perform maintenance functions on z/VSE objects (for example: `LIST`, `EDIT`, `DELETE`, `COPY`, `RENAME` and `BROWSE` z/VSE members).

If the member consists of job control, you can make use of the Natural ISPF macro facility. You can use all types of macro statements. Macro expansion is performed at submission time (see the `SUBMIT` command below).

When creating a new member, you can also use the Edit macro feature to automatically create text lines which can then be modified. For details on the macro facility, see the section *Macro Facility* in the *Natural ISPF Programmer's Guide*).

To enter the z/VSE objects maintenance facility

- Select the `MEMBERS` option from the Natural ISPF Main Menu.

  The VSE Members Entry Panel appears:

```
------------------------- VSE MEMBERS - ENTRY PANEL --------------------------
COMMAND ===>


   Library/DSN   ===>
   Sub library   ===>
   Member        ===>
   Member type   ===>
   Volume        ===>
   VSAM catalog  ===>
   Password      ===>                        ( If password protected )
   Scan for      ===>
   MSHP protect  ===>
   Inline data   ===>
   Edit macro    ===>
   Node          ===> 33




Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

You can specify the z/VSE object you wish to maintain in the input fields and enter a function command in the command line.

Meaning of the input fields:

Field	Meaning
`Library/DSN`	Displays the dataset name last used. You can select any other dataset by overtyping this name, or generate a list of datasets for a specified volume in the same way as described for the `Member` field below. You can also enter a library name here (standard label), which is the name you are advised to use in direct command syntax.
`Sublibrary`	z/VSE sublibrary.
`Member`	Name of required member. Leave blank or use strings and wildcards (* and _) to generate a more selective list of member names in the specified library. See the subsection ***Selection Windows and Wildcards*** in the section *Command Logic*. See also the field `Member type` below.
`Member type`	Members can be grouped and listed according to a user-defined type specified in this field at creation time.
`Volume`	Volume serial number (non-VSAM installations). Required only if the dataset has no label definition.
`VSAM catalog`	Required only if the dataset has no label definition (VSAM installations).
`Password`	System password if file is protected. Must be specified irrespective of read or write protection (VSAM only).
`Scan for`	Lists members which contain the string specified here. When you select a member from this list for `EDIT` or `BROWSE`, the cursor is placed on the first occurrence of this string in the member. Issue the `RFIND` command to find the next occurrence.
`MSHP protect`	Specify `YES` to bypass MSHP protection; otherwise, leave blank (relevant only for the `EDIT` function).
`Inline data`	Specify `YES` to indicate that the member contains `SYSIPT` data; otherwise, leave blank (relevant only for `EDIT` function).
`Edit macro`	Name of macro object to be used as a model for the member. The specified macro is executed and loaded into the Editor. See the section *Macro Facility* in the *Natural ISPF Programmer's Guide* for details. When used with `LIST`, the list contains all members according to the name criteria that use the specified macro as a model.
`Node`	Select Entire System Server node. Enter a question mark (?) and press ENTER to open a window in which all node numbers are scrolled with an `ACTIVE` or `INACTIVE` status report. If you do not specify a node, the default node is assumed.

Function Commands

The available function commands for z/VSE members are as follows:

Command	Parameter Syntax
BROWSE	*library.sublib(member.type)* VOL/CAT=*n* PASSWORD=*p* NODE=*id*
COPY	*library.sublib(member.type)* VOL/CAT=*n* PASSWORD=*p* NODE=*id*, *object-type object-parms*, REP
DELETE	*library.sublib(member.type)* VOL/CAT=*n* PASSWORD=*p* NODE=*id*
EDIT	*library.sublib(member.type)* VOL/CAT=*n* PASSWORD=*p* MACRO=*name* NODE=*id*
EXPORT	*library.sublib(member.type)* VOL/CAT=*n* PASSWORD=*p* NODE=*id*, *target-environment*
HOLD	*library.sublib(member.type)* VOL/CAT=*n* PASSWORD=*p* NODE=*id*
INFO	*library.sublib(member.type)* VOL/CAT=*n* PASSWORD=*p* NODE=*id*
LIST	*library.sublib(*_*)* VOL/CAT=*n* PASSWORD=*p* SC=*string* MACRO=*name* NODE=*id*
PLAY	*library.sublib(member.type)* VOL/CAT=*n* PASSWORD=*p* NODE=*id*
PRINT	*library.sublib(member.type)* VOL/CAT=*n* PASSWORD=*p* NODE=*id*, *printer-name* CC
RENAME	*library.sublib(member.type)* VOL/CAT=*n* PASSWORD=*p* NODE=*id*, *new-name*
SUBMIT	*library.sublib(member.type)* VOL/CAT=*n* PASSWORD=*p* NODE=*id1*, TARGET=*id2*

The VOL parameter (non-VSAM) or CAT parameter (VSAM) is required only if the specified dataset has no label definition.

A full description of these commands is contained in the section *Command Reference*. The object parameters correspond to the input fields on the VSE Members Entry Panel.

Note: If you issue any of the above function commands from outside the VSE members facility, you must specify the object-type parameter MEM before the object parameters.

Listing z/VSE Objects

The following items can be listed using the LIST command and selection criteria in the parameter input fields of the VSE Members Entry Panel:

- Volumes according to prefix;
- File names according to prefix;
- All files on a specified volume; you can optionally narrow the list down by specifying a prefix, suffix or a string for file names with the wildcards * and _ (see description of Member field);
- Sublibraries in a z/VSE file; you can optionally specify a prefix for sublibrary names;
- Members in a sublibrary; you can optionally specify a prefix for member names;
- Members in a sublibrary according to prefix, but which contain a certain character string (Scan for option);

- Versions of a specific member.

If you wish to generate lists using function command syntax in the command line of any screen, you must address the following object types:

Type of List	Object-Type Notation
List of volumes	DV
List of files	FIL
List of sublibraries	SUB
List of members	MEM
List of member versions	VV

Below are some examples of function commands using full command syntax.

Example: LIST (1)

The following figure shows an example of a list of all sublibraries in a z/VSE library generated using the command:

```
LIST SUB NATPROC.IV131.SYSTEM.LIBRARY(*)
```

```
LIST-SUB:NATPROC.IV131.SYSTEM.LIBRARY.* ---------- Row 0 of 8 - Columns 010 025
 COMMAND===>                                                  SCROLL===> CSR
   SUBLIB            MEMBERS
** ****** top of list ******
   SAVE                  2
   TEST1                 7
   UEDTB1                6
   ESX131                5
   ASM                  21
   ESX132                6
   NPR133              146
   NPR134              140
 ** **** bottom of list *****

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Meaning of column headings:

Column	Meaning
SUBLIB	Name of sublibrary.
MEMBERS	Number of members in the sublibrary

Use the line command L to generate a list of all the members in the sublibrary.

Example: LIST (2)

The following figure shows an example of a list of all members in a sublibrary generated using the command:

```
LIST MEM NPR131.DEV135(*)
```

where NPR131 is the standard label library name and DEV135 is the sublibrary name:

```
LIST-MEM:NPR131.DEV135(*) --------------------- Row 63 of 122 - Columns 018 076
 COMMAND===>                                                   SCROLL===> CSR
   MEMBER   TYPE              CREATED  MODIFIED TIME  BLOCKS    SIZE VV.MM ATTR
   XCOTABE  OBJ               19980802   /  /   00:00      5     111       RECT
   ESYNMAIN PHASE             19980802   /  /   00:00     20   18756       RECT
   MYNODE   PHASE             19980802   /  /   00:00      1     104       RECT
   NATPAMS  PHASE             19980802 19981118 08:57      4    3504       RECT
   NATPBPIT PHASE             19980802 19980802 10:20      2    1660       RECT
   NATPCMDL PHASE             19980802   /  /   00:00     23   22088       RECT
   NATPCNTL PHASE             19980802   /  /   00:00      2    1424       RECT
   NATPEVTM PHASE             19980802   /  /   00:00      1     510       RECT
   NATPLIBR PHASE             19980802   /  /   00:00      5    4190       RECT
   NATPSPLA PHASE             19980802 19981019 09:07     10    9016       RECT
   NATPSPVT PHASE             19980802   /  /   00:00      3    2358       RECT
   NATPVTPH PHASE             19980802   /  /   00:00      2    1164       RECT
   NATPVTPR PHASE             19980802   /  /   00:00      5    4304       RECT
   NPRINIT  PHASE             19980802   /  /   00:00      3    2326       RECT
   NPROPHND PHASE             19980802   /  /   00:00      3    2456       RECT
   NTSTRT   PHASE             19980802   /  /   00:00      2    1308       RECT
   RDSPLDEF PHASE             19980802   /  /   00:00      1      86       RECT
   USERVSEC PHASE             19980802   /  /   00:00      1      85       RECT
   XCOMALCF PHASE             19980802   /  /   00:00      1     836       RECT
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Use the RIGHT command, (usually assigned to PF11) to scroll right and display the attributes in the ATTRIBUTES column:

```
LIST-MEM:NPR131.DEV135(*) --------------------- Row 63 of 122 - Columns 064 122
 COMMAND===>                                                   SCROLL===> CSR
   MEMBER   TYPE    ZE VV.MM ATTRIBUTES
   XCOTABE  OBJ     11       RECTYPF
   ESYNMAIN PHASE   56       RECTYPU  AMODE24  RMODE24
   MYNODE   PHASE   04       RECTYPU  AMODE24  RMODE24
   NATPAMS  PHASE   04       RECTYPU  AMODE24  RMODE24
   NATPBPIT PHASE   60       RECTYPU  AMODE24  RMODE24
   NATPCMDL PHASE   88       RECTYPU  AMODE24  RMODE24
   NATPCNTL PHASE   24       RECTYPU  AMODE24  RMODE24
   NATPEVTM PHASE   10       RECTYPU  AMODE24  RMODE24
   NATPLIBR PHASE   90       RECTYPU  AMODE24  RMODE24
   NATPSPLA PHASE   16       RECTYPU  AMODE24  RMODE24
   NATPSPVT PHASE   58       RECTYPU  AMODE24  RMODE24
   NATPVTPH PHASE   64       RECTYPU  AMODE24  RMODE24
   NATPVTPR PHASE   04       RECTYPU  AMODE24  RMODE24
   NPRINIT  PHASE   26       RECTYPU  AMODE24  RMODE24
   NPROPHND PHASE   56       RECTYPU  AMODE24  RMODE24
   NTSTRT   PHASE   08       RECTYPU  AMODE24  RMODE24
   RDSPLDEF PHASE   86       RECTYPU  AMODE24  RMODE24
   USERVSEC PHASE   85       RECTYPU  AMODE24  RMODE24
   XCOMALCF PHASE   36       RECTYPU  AMODE24  RMODE24
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Meaning of the column headings:

Column	Meaning
MEMBER	Member name.
TYPE	Member type.
CREATED	Date the member was created.
MODIFIED TIME	Date and time of last modification.
BLOCKS	Number of blocks occupied by member.
SIZE	Number of records in member.
VV.MM	Version number and modification level of the current version of the member. If this field is not blank, previous versions for this member exist, and Natural ISPF versioning is active when this member is edited.
ATTR	Attributes of objects and phases

The lists appear in Natural ISPF Editor format in BROWSE mode. This means you can use all available Editor BROWSE commands (UP, DOWN, BOTTOM, TOP, FIND, LOCATE).

Example: LIST (3)

The following figure illustrates a list of members generated using the `SCAN=` option in the command:

```
LIST MEM IJSYSRS.SYSLIB(D*) SC=FCT
```

```
LIST-MEM:IJSYSRS.SYSLIB(D*.*)/SC=FCT ------------ Row 0 of 9 - Columns 010 059
COMMAND===>                                                   SCROLL===> CSR
   MEMBER            TYPE       NUM FIRST FOUND
** ********************** top of list ***********************
   DFHFCTSO          A            8 TITLE 'DFHFCTSO -
   DFHFCTSP          A            8 TITLE 'DFHFCTSP -
   DFHSITSP          A            1 FCT=SP,
   DFHFCTSO          OBJ          4 PHASE DFHFCTSO,*
   DFHFCTSP          OBJ          4 PHASE DFHFCTSP,*
   DFHFCTSO          PHASE        1 *DFHFCTSO*   0170U    PIDL
   DFHFCTSP          PHASE        1 *DFHFCTSP*   0170U    PIDL
   DFHSITSP          PHASE        1 *DFHSITSP*  q0170U    PIDL
   DTSFDUMP          PHASE        1 AUXP7     G     MFH
** ********************* bottom of list ************************
Enter-PF13--PF14--PF15--PF16--PF17--PF18--PF19--PF20--PF21--PF22--PF23--PF24---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

If you select a member from a list generated with the Scan for option for `EDIT` or `BROWSE`, the cursor will be placed on the first occurrence of the string. If you then issue the `RFIND` command, the cursor is placed on the next occurrence.

Example: INFORMATION

The following information screen is displayed as a result of the command:

```
INFO NPR131.TEST1(UEDTB1.A)
```

Information is shown on member `UEDTB1` of Type `A` in the sublibrary `TEST1` in the library `NATPROC.IV131.SYSTEM.LIBRARY` (standard label name `NPR131`):

```
------------------------------ VSE MEMBER INFO --------------------------------
COMMAND ===>


    Dsname/Library:  NATPROC.IV131.SYSTEM.LIBRARY
    Sub library   :  TEST1
    Member        :  UEDTB1
    Member type   :  A


      Version     :  01.03               Maximum no. of versions  : 10_
      Modified    :  19980826 10:28.11   Actual  no. of versions  : 3
      Size        :       32

Enter-PF13--PF14--PF15--PF16--PF17--PF18--PF19--PF20--PF21--PF22--PF23--PF24---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

The information fields in this display speak for themselves. Note that you can modify the maximum number of versions field here to override the default value set by the system administrator.

Line Commands

You select a member from a list by typing in a line command in the input field preceding the member name and pressing ENTER. Each line command is an abbreviation of a function command (but see the special `LIST` line command for a member):

Line Command	Corresponding Function Command
B	BROWSE
CP	COPY
D	DELETE
E	EDIT
EX	EXPORT
HL	HOLD
I	INFORMATION
L	`LIST` versions of the member
PL	PLAY
PR	PRINT
R	RENAME
SB	SUBMIT

Line commands can also be used as valid abbreviations of function commands entered in the command line of any system screen.

Local Commands

In Edit Mode:

If you display a member in Editor format in `EDIT` mode, you can issue local commands from the Editor command line in addition to Editor commands.

The following local commands are available:

Command	Meaning
`DATA YES/NO`	Does the edited member contain SYSIPT data? Specify `YES` or `NO`.
`IMPORT`	Imports a PC file or Con-nect document into the member (see the section ***Useful Features***).
`MSHP YES/NO`	Should MSHP protection be bypassed? Specify `YES` or `NO`.
`PASSWORD` *`password`*	If the member is password-protected, use this command to enter the valid password in order to update the member. If you enter the `PASSWORD` command without parameter, a window prompts you for the password. Password input in the window is invisible.
`REGENERATE`	Available for members written using the Edit macro option. Reexecutes the specified macro object and writes the result in protected lines in the current edit session. Any defined user code remains in place. For details, see the section *Macro Facility* in the *Natural ISPF Programmer's Guide*.

In List Mode:

If you display lists of z/VSE members in Editor format, you can issue the following local commands in addition to Editor scroll commands: `ALL`, `LAYOUT`, `RELIST` and `SORT`. For detailed information, see the corresponding subsections in the section ***Useful Features***.

Previous Versions

Previous versions of z/VSE members can be listed and retrieved (see the line command `L` for `LIST`). They are separate objects in Natural ISPF, accessible via the VSE Members Entry Panel or using function commands that address object type `VV` from any screen. To activate the versioning feature, you must issue the command `VERSIONS ON` before starting your edit session.

Note that previous versions of z/VSE members are kept for the physical dataset name. This means that you cannot use the same dataset name for files on different volumes or VSAM catalogs if you wish to use the versioning feature.

For details on versioning in Natural ISPF, see the subsection ***Versioning*** in the section *Useful Features*.

Concurrent Editing of z/VSE Members

When you save a z/VSE member, Natural ISPF checks if the member has been modified by another user or another session while you were editing. If this is the case, you are notified by a message and the `SAVE` operation is not executed.

You can use the `BROWSE` command to inspect the member and you can decide whether to override it with your latest modifications or not. To override it, you can either:

- use the `REPLACE` command for the existing member, or
- delete the existing member and then save the version with your latest changes.

Note: Checking for concurrent editing of z/VSE members is an optional feature of Natural ISPF which may or may not have been activated by your system administrator.

z/VSE File Maintenance

The z/VSE file maintenance facility provides functions that include `LIST`, `BROWSE`, `RENAME`, `ALLOCATE` and `DELETE` datasets. You can also display dataset information and access sublibraries and members from a list of datasets.

Note: You can use this facility to access volume information.

To enter the z/VSE file maintenance facility

- Select the `VSE FILES` option from the Natural ISPF Main Menu.

  The VSE Files Entry Panel appears:

```
---------------------------VSE FILES - ENTRY-PANEL-----------------------------
COMMAND ===>
   Library/DSN   ===>
   Volume        ===>
   VSAM catalog  ===>
   Password      ===>                        ( If password protected )
   Record format ===>
   Record length ===>
   Block size    ===>
   Node          ===>

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

You can specify the file you wish to maintain in the input fields and enter a function command in the command line.

Meaning of the input fields:

Field	Meaning
`Library/DSN`	Displays the file name last used. You can select any other dataset by overtyping this name, or generate a selective list of datasets on the specified volume by using strings and wildcards (* and _) as described in the subsection ***Selection Windows and Wildcards*** in the section *Command Logic*. You can also use the library name (standard label) in this field.
`Volume`	For non-VSAM installations: Required only if the dataset has no label definition. To list volume serial numbers, leave the `Library/DSN` field blank and use the asterisk wildcard (*) as described above.
`VSAM catalog`	For VSAM installations: Required only if the dataset has no label definition.
`Password`	System password if dataset is protected.
`Record format`*	Required when issuing `EDIT` or `BROWSE` for a sequential file.
`Record length`*	Required when issuing `EDIT` or `BROWSE` for a sequential file.
`Block size`*	Required when issuing `EDIT` or `BROWSE` for a sequential file.
`Node`	Select Entire System Server node. Enter a question mark (?) and press ENTER to open a window in which all node numbers are scrolled with an `ACTIVE` or `INACTIVE` status report. If you do not specify a node, the default node is assumed.

* These parameters are not available as keyword parameters in command syntax and must therefore be entered in the VSE Files Entry Panel. In general, these parameters are also required for the functions `PRINT`, `COPY` and `EXPORT`.

Function Commands

The available function commands are:

Command	Object Parameter Syntax
`ALLOCATE`	*library/dsn* `VOL/CAT=`*n*
`BROWSE`*	*library/dsn* `VOL/CAT=`*n* `NODE=`*id*
`COPY`*	*library/dsn* `VOL/CAT=`*n* `NODE=`*id*, *object-type object-parameters*, `REP`
`DELETE`	*library/dsn* `VOL/CAT=`*n* `NODE=`*id*
`EDIT`*	*library/dsn* `VOL/CAT=`*n* `NODE=`*id*
`EXPORT`*	*library/dsn* `VOL/CAT=`*n* `NODE=`*id*
`EXTENTs`	*library/dsn* `VOL/CAT=`*n* `NODE=`*id*

Command	Object Parameter Syntax
INFORMATION	*library/dsn* VOL/CAT=*n* NODE=*id*
LIST	(*_*) VOL/CAT=*n* NODE=*id*
PRINT*	*library/dsn* VOL/CAT=*n* NODE=*id*, *printer-name* CC NO
RENAME	*library/dsn* VOL/CAT=*n* NODE=*id*, *new-name*

* Apply to sequential files only.

The VOL parameter (non-VSAM) or CAT parameter (VSAM) is required only if the specified dataset has no label definition.

A full description of these commands, including the function parameters, is contained in the section *Command Reference*. The object parameters correspond to the input fields on the VSE Files Entry Panel.

Note: If you issue any of the above function commands from outside the dataset facility, you must specify the object-type parameter FIL before the object parameters.

Example: ALLOCATE

With the ALLOCATE command, you can allocate a new dataset specified in the dataset name parameter. Natural ISPF provides a special feature here. If you issue the ALLOCATE command for an existing (allocated) dataset, the information for the dataset is displayed in the following format:

```
--------------------------- ALLOCATE VSE DATASET ------------------------------
COMMAND ==>

   Data set name ===> VSE.SYSRES.LIBRARY
      Volume Serial         ===> VSERES
   or Generic Unit          ===>
      Dataset Organization  ===> SD
      Extent start          ===>              (Leave BLANK for any free space)
      Extent size           ===>
      Expiration Date       ===> 99365        ('YYDDD')
      Node                  ===>

Press ENTER to allocate
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

You can enter in the name of the dataset to be allocated by overtyping the value in the Data set name field. Modify any other value as required and press ENTER to perform the allocation.

Meaning of the fields:

Field	Meaning
`Volume Serial`	Serial number of the volume on which the dataset is to be allocated.
`Generic Unit`	If you do not specify a volume, specify the generic identifier from which a volume is to be selected (for example: `3380`).
`Dataset Organ.`	For example: `DA` - Direct access `SD` - Sequential disk `UN` - Unknown or unidentified `VS` - VSAM
`Extent start`	Specify track number on CKD disks and block number on FBA disks on which extent is to start. If you leave this field blank, any free space is found and allocated.
`Extent size`	Specify tracks on CKD disks and blocks on FBA disks.
`Expiration Date`	Date the file will expire. Until this date is reached, each attempt to update or delete the file will cause a console message, requiring an operator reply.
`Node`	Node number for dataset.

Example: INFORMATION

The following figure is the result of the `INFORMATION` command issued for a z/VSE dataset.

The fields in the information screen reflect the specification of the allocation parameters described for the `ALLOCATE` command above, with additional information such as current number of cylinders and tracks, as well some information about allocated extents:

```
--------------------------- DATA SET INFORMATION -----------------------------
COMMAND ===>

     DATA SET NAME :  VSE.SYSRES.LIBRARY

     GENERAL DATA
       VOLUME SERIAL     : DOSRES
       DEVICE TYPE       : 3380
       ORGANIZATION      : SD
       RECORD FORMAT     :
       RECORD LENGTH     :
       BLOCK SIZE        : 8192
       ALLOCATION        : 63      CYL   0   TRK
       ALLOCATED EXTENTS : 1
       SECURITY          :

     CREATION DATE       : 19980393
     EXPIRATION-DATE     : 31121999

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

To display the file extents, use the command `EXTEND` or the abbreviation `ET`. See the following example.

Example: EXTENT

The following figure is the result of the `EXTENT` command issued for a z/VSE dataset:

```
EXTENTS-FIL:MKH.DATA.TEST ------------------------------------- Columns 001 052
 COMMAND===>                                                   SCROLL===> CSR
   VOLSER NR. BEGIN   END     SIZE    RELATIVE
** ******************** top of list *******************
   ESAPU5   1 321-00  327-09  6-10        4815/100
   ESAPU5   2 947-00  950-04  3-05       14205/50
** ****************** bottom of list ******************

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right TOP
```

The `EXTENT` fields have the following meaning:

Field	Meaning
Begin	Beginning of extent in cylinder and track number.
End	End of extent in cylinder and track number.
Size	Extent size in cylinder and track.
Relative	Starting track number and total number of tracks.

Example: LIST Data Sets

The following list of datasets is the result of the command:

```
LIST FIL VSE*
```

```
LIST-FIL:VSE* ---------------------------------- Row 0 of 20 - Columns 053 076
 COMMAND===>                                                  SCROLL===> CSR
   DATA SET NAME                                  VOLSER          TYPE   F.NAME
** ******************************** top of list ******************************
   VSE.SYSRES.LIBRARY                             VSERES          LIBR   IJSYSRS
                                                  VSERES                 IJSYSR2
   VSE.POWER.QUEUE.FILE.RAW                       VSERES                 IJQFILE
   VSE.SYSTEM.HISTORY.FILE.RAW                    VSERES                 IJSYSHF
   VSE.DUMP.LIBRARY                               SYSWK1          LIBR   SYSDUMP
   VSESP.JOB.MANAGER.FILE.RAW                     SYSWK1                 VSEJMGR
   VSE.HARDCOPY.FILE.RAW                          SYSWK1                 IJSYSCN
   VSE.RECORDER.FILE.RAW                          SYSWK1                 IJSYSRC
   VSE.POWER.DATA.FILE.RAW                        SYSWK1                 IJDFILE
   VSE.CONTROL.FILE                                                      IESCNTL
   VSE.TEXT.REPSTORY.FILE.RAW                                            IESTRFL
   VSE.TEXT.REPSTORY.WORKFILE.RAW                                        IESTRWF
   VSE.MESSAGES.ONLINE                                                   IESMSGS
   VSE.ONLINE.PROB.DET.FILE.RAW                                          IESPRB
   VSE.MESSAGE.ROUTING.FILE.RAW                                          IESROUT
   VSE.PRD1.LIBRARY                                                      PRD1
                                                                         PRDPRIM
   VSE.PRD2.LIBRARY                                               LIBR   PRD2
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

The list lists all datasets starting with z/VSE on the node specified in the Main Menu of the current session.

Meaning of the data provided for each dataset (scroll right to display all columns):

Column	Meaning
VOLSER	Volume serial number
TYPE	Library type. Possible values: (blank) - No Librarian file. LIBR - z/VSE Librarian file.
F.NAME	Library name (standard label).
CLASS	Device class, for example: DASD - direct access TAPE - tape
SERIES	Device series

Example: LIST a VTOC

The following shows an example VTOC generated using the command:

```
LIST FIL * VOL=SYSWK1
```

```
LIST-FIL:*/VOL=SYSWK1 --------------------------- Row 0 of 57 - Columns 046 076
 COMMAND===>                                                  SCROLL===> CSR
   DATA SET NAME                                          RELAT/SIZE  DO X BEGIN
** ******************************** top of list ******************************
   ** VTOC EXTENT **                                       4755/15    UN 0  317-
   VSE.RECORDER.FILE                                       8640/45    UN 0  576-
   VSE.POWER.DATA.FILE                                     6795/1740  DA 0  453-
   VSE.POWER.ACCOUNT.FILE                                  8535/60    DA 0  569-
   VSESP.JOB.MANAGER.FILE                                  8595/15    SD 0  573-
   VSE.SYSTEM.HISTORY.FILE                                 8685/75    UN 0  579-
   Z9999992.VSAMDSPC.T9DB7B95.T8C90C94                      975/2205  VS 0   65-
   Z9999994.VSAMDSPC.T9DB7B95.TC3F350C                     3780/135   VS 0  252-
   Z9999992.VSAMDSPC.T9DB7B96.T30EBED4                     3915/840   VS 0  261-
   VSE.DUMP.LIBRARY                                        3180/600   SD 0  212-
   ICCF.LIBRARY                                            4770/1785  DA 0  318-
   INFO.ANALYSIS.DUMP.MGNT.FILE                            8760/10    SD 0  584-
   INFO.ANALYSIS.EXT.RTNS.FILE                             8770/5     SD 0  584-
   WORK.HIST.FILE                                           900/75    UN 0   60-
   VSE.HARDCOPY.FILE.SHR2                                  6555/30    UN 0  437-
   VSE.HARDCOPY.FILE                                       8610/30    UN 0  574-
   COM441.VSE.HISTORY                                        15/30    UN 0    1-
   VTAM.TRACE.FILE                                         8775/15    SD 0  585-
   VSE.RECORDER.FILE.SHR2                                  6585/45    UN 0  439-
   VSE.RECORDER.FILE.TFS                                   6660/45    UN 0  444-
```

The list contains all files on volume SYSWK1. You must scroll right to display all the information provided (RIGHT command, usually assigned to PF11).

Meaning of the data provided for each file:

Column	Meaning
DO	File organization type. Possible types are: DA - Direct access SD - Sequential disk UN - Unknown or unidentified VS - VSAM
X	Extent number.
RELAT/SIZE	For a CKD disk: starting track/block and size of the space on which the file resides. For an FBA disk: size of file in blocks.
BEGIN	Starting cylinder head or PBN on which the file resides.
END	Ending cylinder head or PBN on which the file resides.
EXP-DATE	Date the file will expire. Until this date is reached, each attempt to update or delete the file will cause a console message, requiring an operator reply.

Example: LIST Volumes

Volumes are separate objects in Natural ISPF, but have no Entry Panel. You can access volume information using the VSE File Entry Panel, or using a function command that addresses object type DV.

The following figure illustrates a list of all volumes generated using the command:

```
LIST DV *
```

```
LIST-DV: ---------------------------------------- Row 0 of 14 - Columns 001 044
 COMMAND===>                                                   SCROLL===> PAGE
   UNIT VOLSER CLASS SERIES STATUS  MOUNT
** **************** top of list ***************
   225  TFS801 DASD  3380   ONLINE
   226  TFS802 DASD  3380   ONLINE
   227  TFS803 DASD  3380   ONLINE
   228  TFS804 DASD  3380   ONLINE
   22A  PQA001 DASD  3380   ONLINE
   A00  VSERES DASD  3380   ONLINE
   A01  SYSWK1 DASD  3380   ONLINE
   A02  VSE410 DASD  3380   ONLINE
   A03  VSE420 DASD  3380   ONLINE
   A04  VSE430 DASD  3380   ONLINE
   A05  VSE440 DASD  3380   ONLINE
   A08  VSE470 DASD  3380   ONLINE
   A14  VSEUC3 DASD  3380   ONLINE
   A18  VSERAW DASD  3380   ONLINE
** ************** bottom of list **************


Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Meaning of the column headings:

Column	Meaning
UNIT	Unit address
VOLSER	Volume serial number
CLASS	Device class, for example: DASD - Direct access TAPE - Tape
SERIES	Device series
STATUS	Device status. Possible values: CHANGE - Device status is changing. OFFLINE - Device is offline. ONLINE - Device is online.
MOUNT	Device mount status. Possible values: MOUNT PEND - Mount is pending. NOT READY - Device not ready. REMOVABLE - Device is removable (e.g., a tape).

Column	Meaning
	RESERVED - Device is reserved. RESIDENT - Device is resident (e.g., a hard disk).

You can select a volume and list its contents (VTOC) by entering the line command L in the input field preceding the UNIT notation.

Line Commands

When listing datasets, the line commands in the following table are available. When listing volumes, only the L line command is available.

Line Command	Corresponding Function Command	Remarks
A	ALLOCATE	
B	BROWSE	Sequential datasets only.
CP	COPY	Sequential datasets only.
D	DELETE	
E	EDIT	Sequential datasets only.
ET	EXTENTS	
EX	EXPORT	Sequential datasets only.
I	INFORMATION	
L	LIST	From a list of volumes, lists datasets on a volume; from a list of files, lists sublibraries.
PR	PRINT	Sequential datasets only.
R	RENAME	

Line commands can also be used as valid abbreviations for function commands entered in the command line of any screen.

Local Commands

In Edit Mode:

If you display a file in Editor format in EDIT mode, you can issue local commands from the Editor command line in addition to Editor commands.

The following local commands are available:

Command	Meaning
IMPORT	Imports a PC file or Con-nect document into the sequential dataset (see the section *Useful Features*).
PASSWORD *password*	If the file is password-protected, use this command to enter the valid password in order to update the file. If you enter the PASSWORD command without parameter, a window prompts you for the password. Password input in the window is invisible.

In List Mode:

If you display lists of z/VSE files in Editor format, you can issue the following local commands in addition to Editor scroll commands: ALL, LAYOUT, RELIST and SORT. For detailed information, see the corresponding subsections in the section *Useful Features*.

z/VSE Job Information

The job information facility allows you to display job information.

To enter the job information facility

- Select the JOBS option from the Natural ISPF Main Menu.

  The Jobs Entry Panel appears:

```
---------------------------- JOBS - ENTRY PANEL -------------------------------
COMMAND ===>

   Job Name      ===>
   Job Number    ===>
   Queue         ===>                          (C,R,L,P,X                )
   Class         ===>                          ( Output/Execution class )
   Disposition   ===>                          ( *,L,D,K )
   User          ===>
   Destination   ===>
   System Id     ===>
   Node          ===> 33


Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

  You can specify the job you wish to handle in the input fields and enter a function command in the command line.

The meaning of the input fields is explained in the following table:

Field	Meaning
`Job Name`	Job name on the job card. Leave blank or use strings and wildcards (* and _) to generate a more selective list of member names in the specified library. See the subsection ***Selection Windows and Wildcards*** in the section *Command Logic*.
`Job Number`	Job number assigned. Only the combination of job name and number uniquely identifies a submitted job (if more than one copy of the same job is active).
`Queue`	Job entry system queue. Possible options: `C` - CRE queue `L` - List queue `P` - Punch queue `R` - Reader queue `X` - Transmit queue
`Class`	Installation-specific job class. Combinations are possible to display a list of jobs of several classes
`Disposition`	How job is handled in `R` queue. Possible options: `*` - Active `D` - Delete after processing (job starts automatically). `H` - Hold job (must be started explicitly). `K` - Keep after processing. Starts automatically as `D`, after run disposition is `L`. `L` - Leave in queue after processing (must be started explicitly).
`User`	When using the `LIST` function, you can restrict the list of jobs by specifying a user ID in this field. You can use the asterisk wildcard (*). Only the jobs submitted by the specified user(s) are selected for display.
`Destination`	When using the `LIST` function, you can restrict the list of jobs by specifying a destination in this field. You can use the asterisk wildcard (*). Only the jobs for the specified destination(s) are selected for display.
`System Id`	When using the `LIST` function, you can restrict the list of jobs by specifying a system ID in this field. You can use the asterisk wildcard (*). Only the jobs under the specified system(s) are selected for display.
`Node`	Node ID of machine on which the job ran. Enter a question mark (?) and press ENTER for a list of available nodes. Leave blank to select the default node.

Function Commands

The available function commands are:

Command	Object Parameter Syntax
BROWSE	*job-number job-name* NODE=*id*
CC	*job-number job-name* NODE=*id*
CHANGE	*job-number job-name* NODE=*id*, NEWCLASS=*c* NEWDEST=*dest* COPIES=*nnn* DISP=*disp* PRIORITY=*p*
COPY	*job-number job-name* NODE=*id*, *object-type object-parms*, REP
EXPORT	*job-number job-name* NODE=*id*, *target-environment*
FOLLOW	*job-number job-name* NODE=*id*
HOLD	*job-number job-name* NODE=*id*
LIST	.*_* QUEUE=*q* CLASS=*c* DISP=*d* USER=*userid* DEST=*dest* SYSID=*sysid* NODE=*id*
PRINT	*job-number job-name* NODE=*id*, *printer-name* CC NO
PURGE	*job-number job-name* NODE=*id*
RELEASE	*job-number job-name* NODE=*id*
STATUS	*job-number job-name* NODE=*id*

You can specify a job by name, number or both. If you use both, the job number must come first.

A full description of these commands (including the function parameters) is contained in the section *Command Reference*. The object parameters correspond to the input fields on the Jobs Entry Panel.

Note: If you issue any of the above function commands from outside the JOBS facility, you must specify the object type DJ before the object parameters.

Below are some examples of function commands using command syntax.

Example: LIST Jobs

The following figure shows the result of the command:

```
LIST DJ * DISP=*
```

The list contains all jobs in the Active queue:

```
LIST-DJ:*/DISP=* --------------------------------- Row 0 of 9 - Columns 029 076
 COMMAND===>                                                  SCROLL===> CSR
   JOBNAME  JOBNO Q NUM C D PR  STAT FROM     TO       CP  PAGES RECORDS ID  CP
** ******************************** top of list ******************************
   CICSWMZ2 23871 R     2 * 003                          1     0      99 F2  0:
   VTAMWMZ  21742 R     3 * 003      SYSA                1     0      16
   COMPWMZ  23671 R     4 * 003                          1     0     156 F4  3:
   ADABAS   23668 R     5 * 003                          1     0      48 F5  1:
   SQLSTART 21746 R     6 * 003      SYSA                1     0      10 F6  0:
   NETPASS  23670 R     8 * 003                          1     0      16 F8  0:
   NETW531  23669 R     9 * 003                          1     0      39
   DELACC   23800 R     C * 003                          1     0      11 C1  0:
   NOM      23700 R     T * 003      PJ                  1     0      41 FB  1:
** ****************************** bottom of list ******************************









Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

You must scroll right to display all available information items (use the `RIGHT` command, usually assigned to PF11).

The list appears in Editor format in `BROWSE` mode.

Meaning of the column headings:

Field	Meaning
`JOBNAME`	Name of job on job card
`JOBNO`	Number of job
`Q`	Job entry system queue
`NUM`	Number of output dataset, if segmented
`C`	Job class
`D`	Disposition
`PR`	Job priority
`STAT`	Job status
`FROM`	Owner of the job, that is, user who issued `SUBMIT` (corresponds to the `USER` parameter on the Jobs Entry Panel)
`TO`	Destination of job.
`CP`	Number of copies of the job.

Field	Meaning
PAGES	Number of pages of the job.
RECORDS	Number of records
ID	Partition ID
CPU	CPU time consumed in minutes
IO	Number of I/O operations performed so far.
STEP	Name of step currently executed
PROC	Name of procedure currently executed
REG	Real storage used by address space in Kbytes
SYSID	System ID

This list of submitted jobs is refreshed every time you press ENTER.

Example: BROWSE

The command:

```
BROWSE DJ CICS*
```

displays an active help window with a list of all jobs that start with CICS. Select a job by marking it with any character and press ENTER to display the job output.

Example: CHANGE

The command:

```
CH DJ MYJOB CLASS=C
```

changes the class of the job MYJOB in a z/VSE environment from C to a new class, and/or assigns other job attributes, which are prompted. If there is more than one copy of the job, use the job number parameter.

Line Commands

The following line commands are available from lists of jobs:

Line Command	Corresponding Function Command
B	BROWSE
CC	CC
CH	CHANGE
CP	COPY
EX	EXPORT

Line Command	Corresponding Function Command
FL	FOLLOW
HL	HOLD
PG	PURGE
PR	PRINT
RL	RELEASE
ST	STATUS

You can also use line commands as valid abbreviations for function commands entered in the command line of any screen.

Local Commands

In List Mode:

If you display lists of z/VSE jobs in Editor format, you can issue the following local commands in addition to Editor scroll commands: ALL, LAYOUT, RELIST and SORT. For detailed information, see the corresponding subsections in the section *Useful Features*.

z/VSE System Operations

With the system operations facility, you can perform certain monitoring and administrative functions: you can display the system log, active jobs, the console, and you can issue operator commands.

To enter the system operations facility, select the SYSTEM option from the Natural ISPF Main Menu. The System Facilities Menu appears:

```
-----------------------------SYSTEM-FACILITIES------------------------------
OPTION  ===>
                                                          Userid   MBE
                                                          Time     16:33:21
   1    LOG        - Display system log                   Terminal 1
   2    VSE ACTIVE - Display address space information     Library  MBE
   3    CONSOLE    - Console display and commands         Node
   4    UNITS      - Display unit information
```

The System Facilities Menu provides four options:

- LOG: display system log;

- `VSE ACTIVE`: display active job information according to selection criteria;
- `CONSOLE`: display console and issue operator commands.
- `UNITS`: list units and display unit information.

For descriptions of the `LOG`, `CONSOLE` and `UNITS` options, see the corresponding subsections in the subsection ***z/OS System Operations*** in the section *z/OS Objects* in this documentation. The `VSE ACTIVE` option is described in the following subsection.

Note: Other than the system log that is described for z/OS, the z/VSE system log shows the date in the very left column in Julian format (YYDDD). For example, "13277" means that day 277 of year 2013 is shown.

Active Jobs Display

You can access active job information by using the System Facilities Menu. Active jobs are also separate Natural ISPF objects with object type `DA` (see the subsection ***Function Commands***).

If you select the `VSE ACTIVE` option from the System Facilities Menu, the Active Jobs Entry Panel appears:

```
-------------------------ACTIVE-JOBS--ENTRY-PANEL----------------------------
COMMAND ===>

   Job            ===>
   Node           ===>
```

Meaning of the input fields:

Field	Meaning
`Job`	Job name. Use strings and wildcards (* and _) to generate a more selective list of jobs as described in the subsection ***Selection Windows and Wildcards*** in the section *Command Logic*.
`Node`	Optional. Specify Entire System Server node.

You can enter selection criteria in the input fields and enter a function command in the command line.

Function Commands

The following function command is available for active jobs:

Command	Object Parameter Syntax
LIST	*_ _* NODE=*id*

Note: If you issue this command from outside the active jobs facility, you must specify object type DA before the object parameters in the command syntax.

Below is an example of the LIST command.

Example: LIST Command

The following figure shows a list of active jobs generated using the command:

```
LIST DA *
```

```
 LIST-DA:* ------------------------------------- Row 0 of 12 - Columns 001 076
  COMMAND===>                                                  SCROLL===> CSR
    ID JOBNAME  CLASSES STEP     PROC     JOBNO  DP VSIZ RSIZ CPU    IO      RID
 ** ******************************** top of list ******************************
    BG NO NAME                                   12 1536    0 0:00.00      3
    FB NO NAME                                   11 2048    0 0:00.00      3
    FA NUC002           ADARUN            00032  10 5120  128 2:05.89  29254
    F9 SMAI040          ADARUN            01150   9 2048    0 0:00.90   1158
    F8 PQACOMPL         PQACOMPL          05077   8 4096 1024 3:45.04  12381 024
    F7 NO NAME                                   7 1024    0 0:00.07     68
    F6 NO NAME                                   6 1024    0 0:00.07     68
    F5 NPRNUC           NPRINIT           00035   5 2048    0 2:42.39  44925 005
    F4 NO NAME                                   4  576  128 0:00.07     68
    F3 VTAMRAW          ISTINCVT          00030   2 3072  424 0:06.98   5444
    F2 NO NAME                                   3 7424 1024 0:00.00      3
    F1 POWSTART         RAWPOWER POWSTRT6        1  768   64 0:03.86   3678
 ** ****************************** bottom of list ******************************




 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
       Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

The list shows all active jobs. Meaning of the column headings:

Column	Meaning
ID	Partition number (ID).
JOBNAME	Job name.
CLASSES	Job class(es) serviced by this partition under the control of z/VSE/POWER.
STEP	Name of step currently executed.
PROC	Name of procedure currently executed.
JOBNO	Job number.
DP	Dispatching priority.
VSIZ	Partition size in Kbytes.
RSIZ	Amount of real storage allocated by z/VSE to the partition in Kbytes.
CPU	CPU time consumed by address space in minutes.
IO	Number of I/O operations performed so far.
RID	Outstanding operator reply ID associated with the job.

The list is refreshed every time you press ENTER.

12 BS2000 Objects

- BS2000 Files 316
- LMS Library Elements 329
- BS2000 Job Information 341
- BS2000 Job Variables 347
- Console Display 356

This chapter explains all the functions you can perform on objects only available in a BS2000 environment.

BS2000 Files

The BS2000 file maintenance facility enables you to perform functions on cataloged disk files of type SAM, ISAM and PAM, including LMS libraries. You can:

- `LIST`, `COPY`, `RENAME` and `DELETE` all types of files, display `INFORMATION` for them and `CHANGE` file attributes;
- `ALLOCATE` all types of files except LMS libraries;
- `BROWSE`, `EDIT`, `SUBMIT`, `PRINT` and `EXPORT` files of type SAM and ISAM.

If the file consists of job control, you can make use of the Natural ISPF macro facility. You can use all types of macro statements. Macro expansion is performed at submission time (see the **`SUBMIT` command** below).

When creating a new file, you can also use the Edit macro feature to automatically create text lines which can then be modified. For details on the macro facility, see the section *Macro Facility* in the *Natural ISPF Programmer's Guide*).

To enter the BS2000 files maintenance facility

- Select the `BS2000 FILES` option from the Natural ISPF Main Menu.

  The BS2000 Files Entry Panel appears:

```
------------------------ BS2000 FILES - ENTRY PANEL ---------------------------
COMMAND ===>


   File Name     ===> *
   Volume        ===>                        ( for selection list    )
   Password      ===>                        ( If password protected )
   Edit macro    ===>
   Shared file   ===>                        ( for selection list    )
   More criteria ===>                       ( mark for special selection )
   Node          ===> 31
```

  You can specify the file you wish to maintain in the input fields and enter a function command in the command line.

Meaning of the input fields:

Field	Meaning
File Name	Displays the file name last used. You can select any other file by overtyping this name. Leave blank or use strings and wildcards (* and _) to generate a more selective list of files. See the subsection ***Selection Windows and Wildcards*** in the section *Command Logic*.
Volume	Volume serial number. Usually not needed, unless you want to restrict a list of files to those on a specific volume.
Password	System password if file is protected. Must be specified irrespective of read or write protection.
Edit macro	Name of macro object to be used as a model for the file. The specified macro is executed and loaded into the Editor. See the section *Macro Facility* in the *Natural ISPF Programmer's Guide* for details. When used with LIST, the list contains all files according to the name criteria that use the specified macro as a model.
Shared file	Enter YES to restrict the LIST function to shared (public) files or NO to restrict it to non-shared files.
More criteria	Option. Can be marked with any character to open a window where additional criteria for LIST selection can be specified. See the corresponding subsection.
Node	Select Entire System Server node. Enter a question mark (?) and press ENTER to open a window in which all node numbers are scrolled with an ACTIVE or INACTIVE status report. If you do not specify a node, the default node indicated on the Main Menu is assumed.

Apart from the input fields listed above, some parameters of the LIST command are supported which do not correspond to an input field of the panel, but which can be specified either by direct command syntax (see below) or by means of the More criteria option:

```
------------------------ BS2000 FILES - ENTRY PANEL ---------------------------
COMMAND ===>

+-------------------------Options for LIST-BS2FILE:-------------------------+
!                                                                            !
! File Name      ==> *_____________________________________________________ !
! Volume         ==> ______                                                 !
! Support Type   ==> ____        ( blank, PUB, PRD, ANYDisk or TAPE    )    !
! Access Type    ==> _____       ( blank, READ or WRITE                )    !
! Shared File    ==> ___         ( blank, YES  or NO                   )    !
! FGGs only      ==> _           ( mark to restrict search             )    !
! Unclosed only  ==> _           ( mark to restrict search             )    !
! FCB Type       ==> _____       ( P, S, I, N or combination           )    !
! Passw.security ==> _____       ( R, W, E, N or combination           )    !
! Range keyword  ==> ___         ( SI, FR, CR, LA, MD, EXD, EXT or BAC )    !
! Range value    ==> ___________________                                    !
! Node           ==> 31_                                                    !
!                                                                            !
+----------------------------------------------------------------------------+
```

Meaning of the input fields:

Field	Meaning
`Support Type`	Select BS2000 files that reside on a specific storage medium. If you omit this parameter, or if you enter the asterisk wildcard (*), all kinds of files are selected, including tape files. Valid options are: ** `ANYDISK` - Files residing on any kind of disk. `PRDISK` - Files on private disks. `PRIVATE` - Synonym for PRDISK. `PUBLIC` - Files on public disks. `TAPE` - Tape files.
`Access Type`	Select BS2000 files that are - or are not - protected against modification. Valid options are: ** `READ` - Protected files only. `WRITE` - Non-protected files only.
`FGGs only`	Mark this field with any non-blank character if only file generation groups and the related individual generations are to be selected and included in the resulting list of BS2000 files.
`Unclosed only`	Mark this field with any non-blank character if you want the resulting list of BS2000 files to contain only files that were opened for output (write, extend or update) and that have not yet been closed.
`FCBType`	Select BS2000 files that correspond to one or more file organization attributes (SAM, PAM, ISAM, NONE or combination). To select for a combination, concatenate the first characters of the requested attributes to form a string, for example: `PI` for PAM and ISAM)

* Fields not mentioned here (`File Name`, `Volume`, `Shared File` and `Node`) contain the values specified in the corresponding fields of the BS2000 Files Entry Panel and have the same meaning as described above.

** Options can be abbreviated.

Field	Meaning
`Passw. security`	Fill in the appropriate specification if you want the resulting list of BS2000 files to contain only files that are (or are not) password protected. Possible values: `E`, `EXEC` or `EXPASS`: Selects only files protected against unauthorized execution `N` or `NONE`: Selects only files without any kind of password protection `R`, `READ` or `RDPASS`: Selects only files protected against unauthorized read access `W`, `WRITE` or `WRPASS`: Selects only files protected against unauthorized modification

Field	Meaning
	Combinations of the above values can also be specified as selection criteria simply by concatenating the value abbreviations. Examples: RWES: Selects all files that are subject to any kind of password protection NW: Selects all files that are either not protected by any password definition, or that are protected against modification.
Range keyword	The range options listed below narrow down a list of files. To use one of these, enter one of the following abbreviations: BAC Backup class option: selects files with a backup frequency attribute that corresponds to the specified value or list of values. Each value must be a character A, B, C, D or E. CR Creation date option: selects files created (or updated) in the period specified in the range field. FR Freesize option: selects files with a number of unused PAM pages within the specified range. EXD Expiration date option: selects files with an expiration date corresponding to the specified time range. EXT Extents option: selects files with an extent count value corresponding to the specified range. LA Last-access-date option: selects files last accessed in the period specified in the range field. MD Modification-date option: selects files last modified in the period specified in the range field. SI Size option: selects files with space allocation (number of PAM pages) corresponding to the given value or range of values. **Note:** The above options cannot be combined in the same selection, that is, only one of these options can be specified in one selection. If you specify one of these options, you must specify the corresponding range itself (either single value or list of values) in the Range value field below.
Range value	Use this field to specify: ■ a range of numbers or dates (i.e. a time period); or ■ a single value or a list of values. The format of this specification depends on the Range option entered in the previous field and is identical to the notation used with a BS2000 FSTATUS command. The only exception is that a range can be given either in the notation (start-v,end-v) or as (start-v:end-v), that is, both a comma and a colon are accepted as separators.

Field	Meaning
	Examples: Range keyword: CR Range value: (1994-06-01,1994-09-30) Range keyword: CR Range value: Yesterday created on previous day Range keyword: LA Range value: (-30:Today) last 30 days Range keyword: SI Range value: (80:) 80 or more PAM pages

Function Commands

The available function commands for BS2000 files are as follows:

Command	Parameter Syntax
ALLOCATE	*file-name* PASSWORD=*p* NODE=*id*
BROWSE	*file-name* PASSWORD=*p* NODE=*id*
CHANGE	*file-name* PASSWORD=*p* NODE=*id*, SHARE=YES/NO ACCESS=READ/WRITE RETPD=*nnnn* READPSWD=*rpwd* WRITEPSWD=*wpwd* EXECPSWD=epwd BACKUP=A/B/C/D/E BACK-VERS=*nn*[1]
COPY	*file-name* PASSWORD=*p* NODE=*id*, *object-type object-parms*, REP
DELETE	*file-name* PASSWORD=*p* NODE=*id*
EDIT	*file-name* PASSWORD=*p* NODE=*id* RECLEN=*nnn* MACRO=*name*
EXPORT	*file-name* PASSWORD=*p* NODE=*id*, *target-environment*
INFORM	*file-name* PASSWORD=*p* NODE=*id*
LIST	*_* NODE=*id* SHARE=YES/NO VOLSER=*vol* ACCESS=READ/WRITE SUPPORT=PUB/PRD/ANY/TAPE FCBTYPE=*ttt* STATE=NOCLOSE/* < SIZE=*s-range* / CREATION=*c-range* / LAST-ACCESS=*la-range* >[2]
PRINT	*file-name* PASSWORD=*p* NODE=*id*, *printer-name* CC
RENAME	*file-name* PASSWORD=*p* NODE=*id*, *new-name*
SUBMIT	*file-name* PASSWORD=*p* NODE=*id1*, TARGET=*id2*

[1] For the CHANGE command, see also the subsection *Changing the Attributes of a BS2000 File*.

[2] Enter only one of the parameters enclosed in brackets < >. For the LIST command, see also the subsection *Listing BS2000 Files*.

A full description of these commands is contained in the section *Command Reference*. For all function commands except LIST and CHANGE, the object parameters correspond to the input fields on the BS2000 Files Entry Panel.

Notes:

1. If you issue any of the above function commands from outside the BS2000 files facility, you must specify the object-type parameter BF before the object parameters.
2. The RECLEN parameter can be specified on the EDIT command to reduce storage requirements. For example, you can specify RECLEN=80 if you issue an EDIT command for a file with record format V which does not contain lines that exceed length 80. If you omit the RECLEN parameter, the value from the catalog information is assumed, if available. Otherwise, the maximum line length for BROWSE or EDIT sessions is taken, that is, 253.
3. The ALLOCATE command applies only to non-standard files. A file with standard attributes (that is, SAM files with record format V) is automatically allocated simply by starting an edit session and issuing the SAVE command. The ALLOCATE command also applies to LMS libraries. LMS libraries can be allocated using the COPY command (for example, copy an empty library) or the ALLOCATE command.
4. For all commands except LIST, if you do not specify the full file name, but only a prefix followed by the asterisk wildcard (*), a selection window opens that lists the existing files which start with that prefix. If you specify only the asterisk wildcard (*) as file name without a prefix, the selection window lists file names for which short IDs are defined, that is, in this case the selection is based on profile definitions rather than on catalog information (see the subsection *Library Definition* in the section *Profile Maintenance*). In both kinds of selection window, you can select a file for further processing by marking it with any character and pressing ENTER.

Listing BS2000 Files

Selecting Files by a Pathname Pattern and/or by Descriptive Parameters

Lists of BS2000 files can be generated using the LIST command and selection criteria in the parameter input fields of the BS2000 Files Entry Panel. Alternatively, you can issue the LIST command with appropriate parameters from any system screen. In the FILE-NAME parameter, you can optionally specify a catalog ID and/or a user ID. This lists shared files only. If you omit the catalog ID / user ID specification, or if you specify the current user ID, all files for which the user ID is authorized are listed.

Notes:

1. If a catalog ID is specified, it must be embedded between colons (:).
2. If a user ID is specified, it must be preceded by the dollar sign ($).

Keyword Syntax for Descriptive Parameters

Descriptive parameters can be specified with usual keyword syntax.

- The keyword parameters `SHARE` and `VOLSER` correspond to the fields on the BS2000 Files Entry Panel (see above).
- The keyword parameters `ACCESS`, `SUPPORT` and `FCBTYPE` correspond to the fields on the More criteria options screen; the keyword parameter `STATE` corresponds to the field Unclosed only on that screen. Specify `STATE=NOCLOSE` to restrict the list to files that were opened for output (`WRITE`, `EXTEND` or `UPDATE`) and have not yet been closed.
- The keyword parameter `SIZE` can be used to select BS2000 files with space allocation (number of PAM pages) corresponding to the given value or range of values. A range must be specified as `(n:m)`, for example: `(24:999)`. If only one of the boundaries n and m is relevant, you can omit the other.
- The keyword parameter `CREATION` can be used to select BS2000 files that were created (or updated) on a specific day, or within a range of days. A range must be specified as `(from:to)`. If only one of the boundaries (*from* or *to*) is relevant, you can omit the other one. The boundaries, as well as a specific day, can both be specified in one of the following notations:

Notation	Description
YYYY-MM-DD	Long date notation
YYMMDD	Short date notation
-n	Relative date notation
`T` or `TODAY`	(relative to the current day) Current day
`Y` or `YESTERDAY`	Previous day

- The keyword parameter `LAST-ACCESS` can be used to select BS2000 files that were last accessed on a specific day, or within a range of days. The syntax conventions for specifying the day or time range are the same as for `CREATION`.
- The keyword parameters `SIZE`, `CREATION` and `LAST-ACCESS` are mutually exclusive, that is, these must not occur in the same command.

Example: LIST BF (1)

Below are some examples of the `LIST` function commands using full command syntax.

1. `LIST BF ADA*`

   lists all files starting with `ADA` owned by current user ID.

2. `LIST BF $DEMO.*ADA*`

   lists all public files owned by user ID `DEMO` that have `ADA` as part of their name.

3. `LIST BF :K:$DEMO.*`

   lists all public files in catalog `K` owned by user ID `DEMO`.

4. `LIST BF *LIS* LAST-ACCESS=(940601:940930)`

   lists all files owned by current user ID with string `LIS` occurring as part of the file name, and last accessed within the specified period.

5. `LIST BF $DEMO.* NODE=38`

   lists all public files owned by user ID `DEMO` on Entire System Server Node 38 in a multi-node environment.

An example of a list of BS2000 files follows.

Example: LIST BF (2)

The following figure shows an example of a list of BS2000 files generated using the command:

```
LIST BF :NAT:$ABC.*TEST*
```

The list generated shows all public files owned by user ID `ABC` in catalog `NAT` that have the string `TEST` in their names:

```
LIST-BF:(125)*TEST* ---------------------------- Row 0 of 37 - Columns 056 076
 COMMAND===>                                                  SCROLL===> PAGE
   PATH-NAME OF FILE                                             PAM-PAGES  M
** ******************************** top of list ********************************
   :NAT:$ABC.D.TEST.JOBLIB                                       0000005475 2
   :NAT:$ABC.D.TEST.NPR.JOBLIB                                   0000004800 2
   :NAT:$ABC.D.TEST.PAM                                          0000000450 2
   :NAT:$ABC.D.TEST.SAM                                          0000000192 2
   :NAT:$ABC.E.ISP.TEST                                          0000000003 2
   :NAT:$ABC.E.TEST.NPR.SPOOL                                    0000000003 2
```

Use the RIGHT command (usually assigned to PF11) to scroll right in order to display more information:

```
LIST-BF:(125)*TEST* ---------------------------- Row 0 of 37 - Columns 071 091
 COMMAND===>                                                  SCROLL===> PAGE
   PATH-NAME OF FILE                                       GES  MODIF-DATE TIME
** ******************************** top of list ******************************
   :NAT:$ABC.D.TEST.JOBLIB                                 5475 2012-11-16 13:11
   :NAT:$ABC.D.TEST.NPR.JOBLIB                             4800 2012-11-16 13:11
   :NAT:$ABC.D.TEST.PAM                                    0450 2000-08-31 10:08
   :NAT:$ABC.D.TEST.SAM                                    0192 2000-08-31 10:08
   :NAT:$ABC.E.ISP.TEST                                    0003 2000-05-25 17:05
   :NAT:$ABC.E.TEST.NPR.SPOOL                              0003 2000-02-15 16:02
```

Meaning of column headings:

Column	Meaning
PATH-NAME OF FILE	Full path name including catalog ID and user ID specification.
PAM-PAGES	Number of disc pages allocated for the file.
MODIF-DATE	Date when the file was last modified.
TIME	Time when the file was last modified.

Session Header Information

The session header reflects the path name pattern used for selection and, optionally, additional criteria specified by means of function command syntax in the BS2000 Files Entry Panel or in the LIST options window. If this information does not fit into a screen line, it is truncated.

Example: Session Header Line

```
LIST-BF:*LIB*/LAST=(940601:940930) ----------- Row 0 of 2 - Columns 056 071
```

Line Commands

You select a file from a list by typing in a line command in the input field preceding the file name and pressing ENTER. Each line command is an abbreviation of a function command.

Line Command	Function Command	Meaning
A	ALLOCATE	Allocate a new file. The attributes of the selected file are displayed, which you can modify for the new file.
B	BROWSE	Display sequential file (SAM or ISAM). No modification is possible.
CH	CHANGE	Change attributes of file, for example, define a password.
CP	COPY	Copy the file into another object (target object type and name will be prompted).
D	DELETE	Erase the file from catalog and/or disk.
E	EDIT	Edit sequential file (SAM or ISAM).

Line Command	Function Command	Meaning
EX	EXPORT	Export sequential file (SAM or ISAM) to external environment (PC or Con-nect).
I	INFORM	Display file attributes.
L	LIST	List elements of the file, if LMS-type file.
PR	PRINT	Create a report of sequential file (SAM or ISAM).
R	RENAME	Rename the file.
SB	SUBMIT	Submit the file (an ENTER job; if applicable, after macro expansion).

Line commands can also be used as valid abbreviations of function commands entered in the command line of any system screen.

Local Commands

If you display a file in Editor format, you can issue some local commands from the Editor command line in addition to Editor commands.

The following local commands are available.

In Edit Mode:

Command	Meaning
IMPORT	Imports a PC file or Con-nect document into the file (see the section ***Useful Features***).
PASSWORD *password*	If the file is password-protected, use this command to enter the valid password in order to update the file. If you enter the PASSWORD command without parameter, a window prompts you for the password. Password input in the window is invisible.
REGENERATE	Available for files written using the Edit macro option. Reexecutes the specified macro object and writes the result in protected lines in the current edit session. Any defined user code remains in place. For details, see the section *Macro Facility* in the *Natural ISPF Programmer's Guide*.

In List Mode:

If you display lists of BS2000 files in Editor format, you can issue the following local commands in addition to Editor scroll commands: ALL, LAYOUT, RELIST and SORT. For detailed information, see the corresponding subsections in the section ***Useful Features***.

Changing the Attributes of a BS2000 File

The CHANGE function command allows you to modify attributes of a file that are stored in the system catalog. You can specify one or more new attribute values as function parameters, using keyword parameter syntax (see Example 3).

If you issue the CHANGE command with or without object parameters, but without any function parameters, or if you issue the line command CH from a list of files, a window opens that displays the current attribute values, except for passwords. To modify one or more attributes, overtype the values in the input fields and press ENTER.

Example 1:

The following window opens if you enter the command:

```
CHANGE BF D.CH.TEST
```

```
------------------------ BS2000 FILES - ENTRY PANEL --------------------------
COMMAND ===> CHANGE BF D.CH.TEST


   File Name     ===>
   Vol +-------------------CHANGE-BS2FILE: D.CH.TEST-------------------+
   Pas !                                                                !
   Edi ! Modify attributes as desired:                                  !
   Sha !                                                                !
   Mor !  SHARED FILE      :  NO_       (YES/NO)                        ! n )
   Nod !  ACCESS-TYPE      :  WRITE     (READ/WRITE)                    !
       !  RETENTION-PERIOD :  ____      (in days)                       !
       !  READ-PASSWORD    :                                            !
       !  WRITE-PASSWORD   :                                            !
       !  EXEC-PASSWORD    :                                            !
       !  BACKUP-TYPE      :  D                                         !
       !  #BACK-VERS       :  5                                         !
       !                                                                !
       +----------------------------------------------------------------+




Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Meaning of the input fields:

Field	Meaning
SHARED FILE - SHARE	Enter Y or YES to make the file accessible to other users. Enter N or NO to restrict access to the owner user ID.
ACCESS-TYPE - ACCESS	Enter W or WRITE to allow write access to the specified file, which also implicitly includes authorization for read access. Enter R or READ to allow read access only.
RETENTION-PERIOD - RETPD	Enter a number representing a retention period (in days) that is to be assigned to the specified file: This means it can neither be modified nor deleted within the specified period of time.
READ-PASSWORD - READPSWD	Enter a password to protect the file against unauthorized read accesses. Enter *NONE to reset an existing password, provided that you specified that password together with the current CHANGE function.
WRITE-PASSWORD - WRITEPSWD	Enter a password to protect the file against unauthorized modification or deletion. Enter *NONE to reset an existing password, provided that you specified that password together with the current function.
EXEC-PASSWORD - EXECPSWD	Enter a password to protect the file against unauthorized execution; this applies to procedure files and load modules. Enter *NONE to reset an existing password, provided that you specified that password together with the current CHANGE function.
BACKUP-TYPE - BACKUP	Enter a backup class to be assigned to the specified file, thus controlling how the file is to be handled by (periodic) runs of the ARCHIVE and HSMS utility routines. Possible values: A - Saved with every ARCHIVE run (default). B - Saved with runs for Level B, C and D. C - Saved with runs for Level C and D. D - Saved with runs for Level D. E - Not automatically saved. E is the only value which is valid for temporary files.
#BACK-VERS - BACK-VERS	Enter a value from 0-32 for the maximum number of file versions to be stored in the version backup archive. Entering the value 0 is the equivalent of ignoring the version backup, as no file versions will be stored then.

Example 2:

If the file you wish to maintain is password-protected, correct password specification is also required for the `CHANGE` command. For example, if you enter the command:

```
CHANGE BF T.TEST PASSW=HUGO
```

the following window opens:

```
------------------------ BS2000 FILES - ENTRY PANEL ---------------------------
COMMAND ===> CHANGE BF T.TEST PASSW=HUGO


   File Name     ===>
   Vol +--------------------CHANGE-BS2FILE: T.TEST---------------------+
   Pas !                                                                !
   Edi ! Modify attributes as desired:                                  !
   Sha !                                                                !
   Mor !  SHARED FILE      :  YES       (YES/NO)                        ! n )
   Nod !  ACCESS-TYPE      :  WRITE     (READ/WRITE)                    !
       !  RETENTION-PERIOD :  ____      (in days)                       !
       !  READ-PASSWORD    :                                            !
       !  WRITE-PASSWORD   :            (Enter *NONE to reset)          !
       !  EXEC-PASSWORD    :                                            !
       !  BACKUP-TYPE      :  B                                         !
       !  #BACK-VERS       :  3                                         !
       !                                                                !
       +----------------------------------------------------------------+



Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Enter `*NONE` to reset the password.

To avoid visible input of the old password on your screen, issue the `CHANGE` command from the BS2000 Files Entry Panel and specify the password in the corresponding input field of the `CHANGE` window.

Example 3:

To avoid being prompted for the new attribute values (for example, during a batch run or during execution of a command script), specify them as function parameters using the keywords indicated above. For example, the command:

```
CH BF DEMO.PHASE, ACCESS=READ BACKUP=E
```

inhibits modification of the file `DEMO.PHASE` and assigns backup type `E`, that is, the file is not saved automatically with any periodic `ARCHIVE` run.

Concurrent Editing of Files - BS2000 Version 11 or higher

When you save a BS2000 file, Natural ISPF checks if the file has been modified by another user or another session while you were editing. If this is the case, you are notified by a message and the `SAVE` operation is not executed. However, this check can only be performed if Version 11 (OSD V1) or higher of BS2000 is installed at your site. For lower versions of BS2000, Natural ISPF does not check for concurrent editing. If your `SAVE` operation is rejected due to concurrent editing, you can use the `BROWSE` command to inspect the file and you can decide whether to override it with your latest modifications or not. To override it, you can either:

- use the `REPLACE` command for the existing file, or
- delete the existing file and then save the version with your latest changes.

LMS Library Elements

The BS2000 LMS library elements maintenance facility allows you to perform functions on elements of LMS programming libraries.

If the element consists of job control, you can make use of the Natural ISPF macro facility. You can use all types of macro statements. Macro expansion is performed at submission time (see the `SUBMIT` command below). When creating a new element, you can also use the Edit macro feature to automatically create text lines which can then be modified. For details on the macro facility, see the section *Macro Facility* in the *Natural ISPF Programmer's Guide*).

To enter the LMS element maintenance facility

- Select the `LMS` option from the Natural ISPF Main Menu.

  The LMS Elements Entry Panel appears:

```
------------------------- LMS ELEMENTS - ENTRY PANEL ---------------------------
COMMAND ===>


File Name  =>
Element    =>
Type       =>                          ( S,M,J,D,H,P,R,C,X or combination   )
Version    =>
Rec.length =>                          ( If different from default (80/133) )
Password   =>                          ( If library is password protected   )
Scan for   =>
Edit macro =>
Node       => 31





Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right :s
```

You can specify the element you wish to maintain in the input fields and enter a function command in the command line.

Meaning of the input fields:

Field	Meaning
`File Name`	Displays the file name last used. You can select any other file by overtyping this name. Leave blank or use strings and wildcards (* and _) to generate a more selective list of PAM files. See the subsection ***Selection Windows and Wildcards*** in the section *Command Logic*. When requesting a list of files, the command is automatically transferred to object type `BF` (see the subsection ***BS2000 Files***).
`Element`	Name of the required element in the specified file. Leave blank or use selection criteria as described for the `File Name` field to generate a list of elements. If more than one version exists for an element, the highest version is listed.
`Type`	Type of required element. This is optional if the element name is unique. If you are creating a new element and omit this field, type `S` (source) or the value from your profile is taken as default. When listing elements, combinations of type notation are possible, for example, `JD` for jobs and data elements.
`Version`	You can specify a previous version of the element in this field. If this is omitted, either the highest version is selected, or a version according to the value in your user profile (default is the highest possible version, indicated by the character §, or the character that corresponds to hexadecimal 7C) .
`Record length`	This must be specified if the actual record length exceeds the default length according to element type. The defaults are:

Field	Meaning
	80 for types S, M, J and D. 133 for type P. 253 for all others.
Password	System password if library or element is protected. Must be specified irrespective of read or write protection.
Scan for	Lists elements which contain the string specified here. When you select a member from this list for EDIT or BROWSE, the cursor is placed on the first occurrence of this string in the element. Issue the RFIND command to find the next occurrence.
Edit macro	Name of macro object to be used as a model for the new element. The specified macro is executed and loaded into the Editor. See the section *Macro Facility* in the *Natural ISPF Programmer's Guide* for details. When used with LIST, the list contains all elements according to the name criteria that use the specified macro as a model.
Node	Select Entire System Server node. Enter a question mark (?) and press ENTER to open a window in which all node numbers are scrolled with an ACTIVE or INACTIVE status report. If you do not specify a node, the default node indicated on the Main Menu is assumed.

Function Commands

The available function commands for LMS elements are as follows:

Command	Parameter Syntax
BROWSE	*file-name(element)* TYPE=*t* VERSION=*vvv* RECLEN=*nnn* PASSWORD=*p* NODE=*id*
COPY	*file-name(element)* TYPE=*t* VERSION=*vvv* PASSWORD=*p* NODE=*id*, *object-type object-parms*, REP
DELETE	*file-name(element)* TYPE= *t* VERSION=*vvv* PASSWORD=*p* NODE=*id*
EDIT	*file-name(element)* TYPE=*t* VERSION=*vvv* RECLEN=*nnn* PASSWORD=p NODE=*id* MACRO=*name*
EXPORT	*file-name(element)* TYPE=*t* VERSION=*vvv* PASSWORD=*p* NODE=*id*, *target-environment*
LIST	*file-name* (*_*) TYPE=*t* PASSWORD=*p* NODE=*id* SCAN=*string* MACRO=*name*
PLAY	*file-name(element)* TYPE=*t* VERSION=*vvv* PASSWORD=p NODE=*id*
PRINT	*file-name(element)* TYPE=*t* VERSION=*vvv* PASSWORD=*p* NODE=*id*, *printer-name* CC
RENAME	*file-name(element)* TYPE=*t* VERSION=*vvv* PASSWORD=*p* NODE=*id*,*new-name*
SUBMIT	*file-name(element)* TYPE=*t* VERSION=*vvv* PASSWORD=p NODE=*id1*, TARGET=*id2*

A full description of these commands is contained in the section ***Command Reference***. The object parameters correspond to the input fields on the LMS Elements Entry Panel.

Notes:

If you issue any of the above function commands from outside the LMS elements facility, you must specify the object-type parameter LMS before the object parameters.

The file-name parameter is optional, Natural ISPF then takes the current file name, or if you issue a command from outside the LMS elements facility, from your profile.

Instead of typing the keywords `TYPE` and `VERSION` when specifying an element, the following notations are valid:

```
FILENAME(ELNAME)T/VVVVV
FILENAME(element-selection)
element-selection              /* Current file name is assumed
```

where *`element-selection`* can be any of the following:

```
(T)ELNAME/VVVVV
ELNAME/VVVVV(TTT)
ELNAME/VVVVV
```

Note that `TTT` represents a string of valid element types when requesting a list of elements. The version notation (`/VVVVV`) is required only when selecting a version other than the highest one.

Listing LMS Elements

Lists of elements in an LMS-type library can be generated using the `LIST` command and selection criteria in the parameter input fields of the LMS Elements Entry Panel. Alternatively, you can issue the `LIST` command with appropriate parameters from any system screen. The list can be restricted to selected elements by specifying certain selection criteria:

- Element name prefix or pattern using wildcards (* and _);
- Element type or multiple types;
- Specific element version;
- Scan value, listing only those elements that contain the specified string;
- Model name, listing only those elements that were generated by the specified Edit macro.

Example: LIST LMS (1)

Below are some examples of the `LIST` function commands using full command syntax.

1. `LIST LMS $DEMO.DEMOLIB(*OC*)`

   lists all elements in the specified library that have the string `OC` in their name.

2. `LIST LMS $DEMO.DEMOLIB(ED*) TYPE=SM`

   lists elements in the specified library with names starting with `ED`; only `S` (source) and `M` (macro) type elements are listed.

3. `LIST LMS $DEMO.DEMOLIB(*/120) TYPE=D`

   lists all library elements of Type `D` that have Version Name `120` assigned.

4. `LIST LMS $DEMO.DEMOLIB(E.*) SCAN=EXAMPLE`

   lists all library elements that start with `E.` and which contain the string `EXAMPLE`.

5. `LIST LMS $DEMO.DEMOLIB(E.*) MODEL=EXB5`

   lists all library elements that start with `E.` and that were generated using Edit macro `EXB5`.

An example list of LMS elements follows.

Example: LIST LMS (2)

The following figure shows an example of a list of LMS elements generated using the command:

```
LIST LMS :D:$ASF.ASF.LMSLIB(*)
```

```
LIST-LMS::D:$ASF.ASF.LMSLIB(*) ------------------ Row 0 of 18 - Columns 018 072
 COMMAND===>                                                  SCROLL===> CSR
   TYP ELEMENT-NAME          S V-C HIGH-VERSION CREATED    MODIF-DATE TIME
** ****************************** top of list ******************************
   (J) Z132B005              V   1 000          1993-08-05 1993-08-05 15:31
   (R) ACLTEST               V   1 §            1992-10-06 1992-10-08 12:53
   (R) ARCTEST               V   1 §            1992-08-05 1992-08-05 13:13
   (R) ASFJOIN               V   1 §            1993-03-31 1993-03-31 16:43
   (R) ASFJV                 V   1 §            1993-02-18 1993-02-26 17:08
   (R) ASFMACT               V   1 §            1993-01-20 1993-01-22 15:49
   (R) ASFMP                 V   1 §            1993-04-06 1993-04-08 11:32
   (R) ASFSW                 V  +2 §            1993-03-18 1993-03-25 18:34
   (R) ASFTEST               V   1 §            1992-09-29 1993-02-01 16:50
   (R) ASFTYPIO              V   1 §            1992-12-08 1993-05-05 11:00
   (R) FSTAT                 V   1 §            1992-09-29 1992-09-29 14:59
   (R) GUARDTST              V   1 §            1992-10-28 1992-10-28 18:05
   (R) MACTEST               V   1 §            1992-04-14 1992-04-14 15:16
   (R) MAC31                 V   1 §            1992-10-23 1992-10-23 14:46
   (R) SAMTEST               V   1 §            1993-03-18 1993-03-18 11:24
   (S) S.SAMTEST             V   1 §            1990-10-28 1990-10-28 15:39
   (S) SAMTEST               V   1 §            1990-10-28 1990-10-28 17:45
   (X) L.SAMTEST             V   1 §            1990-10-28 1990-10-28 15:39
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right :s
```

The list generated shows all elements in the specified library. The list appears in Editor format. This means you can use Editor `BROWSE` commands (`UP`, `DOWN`, `TOP`, `BOTTOM`, `LEFT`, `RIGHT`, `FIND`, `LOCATE`), as well as `SORT` and `LAYOUT`.

Meaning of column headings (without the `SCAN` option):

Column	Meaning
`TYP`	Element type, enclosed in parentheses
`ELEMENT-NAME`	Name of element
`S`	Storage mode (`D`=Delta, `V`=Full versions)
`V-C`	Number of versions for the element (version count)
`HIGH-VERSION`	Highest version encountered
`CREATED`	Creation date of highest version
`MODIF-DATE`	Date the highest version was last modified or renamed
`TIME`	Time the highest version was last modified or renamed

If the `SCAN` option were used to create a list, the following columns would be displayed:

Column	Meaning
TYP	Element type, enclosed in parentheses
ELEMENT-NAME	Name of element
NUM	Number of occurrences of the scan string in the highest version
FIRST FOUND	First occurrence of the scan string in the highest version.

Line Commands

You select an element from a list by typing in a line command in the input field preceding the element name and pressing ENTER. Each line command is an abbreviation of a function command (but see the special LIST line command for an element):

Line Command	Function	Meaning
B	BROWSE	Display element. Not available for Type C. No modification is possible.
CP	COPY	Copy the element into another object (target object type and name will be prompted).
D	DELETE	Erase the element from the library. Note that all versions are deleted.
E	EDIT	Edit the element. Not available for types C and R. Elements stored in Delta mode cannot be updated (the SAVE command is rejected).
EX	EXPORT	Export element to external environment (PC or Con-nect). Not available for type C.
L	LIST	List all versions of the element. See the subsection *Versions of LMS Elements* below.
PL	PLAY	Run the element as a Natural ISPF command script. Not available for Types C and R.
PR	PRINT	Create a report of element. Not available for Type C.
R	RENAME	Rename the element (not available for elements stored in Delta mode). Note that all versions are renamed.
SB	SUBMIT	Submit the element (an ENTER job; if applicable, after macro expansion). Only available for Type J.

Line commands can also be used as valid abbreviations of function commands entered in the command line of any system screen.

Local Commands

If you display an LMS element in Editor format, you can issue some local commands from the Editor command line in addition to Editor commands.

The following local commands are available:

In Edit Mode:

Command	Meaning
IMPORT	Imports a PC file or Con-nect document into the element (see the section ***Useful Features***).
NEWNAME *elname/version*	Specifies a new element name and/or version name to be used on subsequent SAVE operations. If you enter the NEWNAME command without parameters, a window prompts you for the new element and version name(s).
PASSWORD *password*	If either the library or the edited element is password-protected, use this command to enter the valid password in order to update the element. If you enter the PASSWORD command without parameter, a window prompts you for the password. Password input in the window is invisible.
REGENERATE	Available for elements written using the Edit macro option. Reexecutes the specified macro object and writes the result in protected lines in the current edit session. Any defined user code remains in place. For details, see the section *Macro Facility* in the *Natural ISPF Programmer's Guide*.

Hint for use of NEWNAME:

You may find it useful to define the greater than character (>) as a magic character in your user profile and assign it to the command ;NEWNAME_. This allows you to use a direct command notation very similar to the command syntax of LMS when used in a TIAM dialog session. For example:

1. `EDIT (D)DOCTX/002>DOCU-NEW`

   opens an edit session on data element DOCU-NEW, filled with the contents of data element DOCTX, version 002. The new element will have the same version identification 002 assigned.

2. `E $VG.CVRP(CVPROT)>/V-TEST`

   opens a session with element CVPROT of any type (provided this element exists and is of unique type) in library $VG.CVRP. Data is read from the highest version encountered and is saved as version V-TEST.

In List Mode:

If you display lists of LMS elements in Editor format, you can issue the following local commands in addition to Editor scroll commands: ALL, LAYOUT, RELIST and SORT. For detailed information, see the corresponding subsections in the section ***Useful Features***.

Concurrent Editing of LMS Elements

When you save an LMS library element, Natural ISPF checks if the same element has been modified by another user or another session while you were editing. If this is the case, you are notified by a message and the `SAVE` operation is not executed. You can use the `BROWSE` command to inspect the element and you can decide whether to override it with your latest modifications or not. To override it, you can either:

- use the `REPLACE` command for the existing element, or
- delete the existing LMS element and then save the version with your latest changes.

Versions of LMS Elements

Previous versions of LMS elements can be kept and retrieved using Natural ISPF. The way versioning is done for LMS elements is slightly different from objects on other platforms, therefore most of the information given in the subsection ***Versioning*** in the section *Useful Features* does not apply.

Listing Previous Versions

You can list previous versions of an LMS element by either:

- Issuing the `LIST` function command from any system screen, specifying the library and element selection criteria in the command syntax (note the version and type selection criteria described in the subsection ***Function Commands*** above), or:
- Selecting an element from a list of elements using the `L` line command.

Examples using function command syntax:

1. `LIST LMS $DEMO.DEMOLIB(E.EXAM/*)`

   lists all versions of the specified library element.

2. `LIST LMS $DEMO.DEMOLIB(EXAMPLE/*N*) TYPE=S`

   lists those versions of source-type element `EXAMPLE` that have the letter `N` in their names. The `TYPE` parameter is required only if the element name `EXAMPLE` is not unique in the library (for example, if an element `EXAMPLE` of Type `J` is contained in the same library).

3. `LIST LMS $DEMO.DEMOLIB(EXAMPLE) VERSION=V* SCAN=EXAM`

   lists those versions of library element `EXAMPLE` whose names start with `V` and which contain the string `EXAM`.

The following screen shows an example of a list of library element versions. The function command used to generate the list was:

```
LIST LMS :D:$ASF.ASF.LMSLIB(ASFSW)R/*
```

The list displays all versions of the element ASFSW in the specified library:

```
LIST-LMV::D:$ASF.ASF.LMSLIB(ASFSW)R/* ------------ Row 0 of 2 - Columns 014 071
 COMMAND===>                                                  SCROLL===> CSR
   VERSION-NAME          S CREATED    MODIFIED-DATE TIME  USER-DATE  TIME
** ***************************** top of list ******************************
   000                   V 1993-03-25    1993-03-25 18:27 1993-03-18 09:40
   §                     V 1993-03-18    1993-03-25 18:34 1993-03-25 18:34
** *************************** bottom of list ****************************









Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right :s
```

Meaning of column headings:

Column	Meaning
VERSION-NAME	Name of element version.
S	Storage mode (D=Delta, V=Full versions)
CREATED	Creation date of version
MODIF-DATE	Date when version was last modified or renamed
TIME	Time when version was last modified or renamed.
USER-DATE	Date supplied by user on last modification
TIME	Time supplied by user on last modification

If the SCAN option were used to create a list, the following columns would be displayed:

Column	Meaning
VERSION-NAME	Name of element version.
NUM	Number of occurrences of the scan string in the version.
FIRST FOUND	First occurrence of the scan string in the version.

Available Functions for Previous Versions

A number of functions can be performed on versions of LMS elements. A version can be selected in either of the following ways:

- Select a version by entering a line command in the input field preceding the version name in the list of element versions;
- Perform the function from any system screen by issuing a function command directed at object type LMV, identifying the version by library name, element name, and version name. The TYPE parameter is required only if the element name is not unique in the library.

Line commands can also be used as abbreviations of function commands in command syntax. Available functions for LMS element versions are:

Line Command	Function	Meaning
B	BROWSE	Display element version. Not available for Type C. No modification is possible.
CP	COPY	Copy the version into another object (target object type and name will be prompted).
D	DELETE	Erase the element version.
E	EDIT	Edit the element version. Not available for Types C and R. Versions of elements stored in Delta mode cannot be updated (the SAVE command is rejected).
EX	EXPORT	Export element version to external environment (PC or Con-nect). Not available for Type C.
PL	PLAY	Run the element version as a Natural ISPF command script. Not available for Types C and R.
PR	PRINT	Create report of the version. Not available for Type C.
R	RENAME	Rename the element name, the version name, or both (V-storage type elements only).
SB	SUBMIT	Submit the element version (an ENTER job; if applicable, after macro expansion). Only available for Type J.

Updating Versioned LMS Elements in Full Storage Mode

Other Natural ISPF objects such as Natural programs or PDS members can be updated in place (with versioning disabled), or continuous versioning can be in effect for each update. In contrast, LMS elements kept in Full Storage Mode can be updated in place, even if there already are several versions of the element being updated. Additionally, for LMS elements in Full Storage Mode, no version identifier is automatically assigned. This means that every time you are about to update an LMS element, you can decide whether you wish to keep the current version or not, and if you do, you can specify a name for the new version according to your site's naming conventions.

Specifically, if you intend to update a certain LMS element and wish to keep the current version, you can proceed in any of the following ways:

1. Start an edit session with the current version, for example, by selecting it from a list of LMS elements with the `E` line command. Modify the data and use the `NEWNAME` local command to assign a new name to the version. This version is stored with the `SAVE` command, or when the session is ended (provided `AUTOSAVE=ON` is specified in your user profile).
2. If the LMS Override option has been set to `*` in your BS2000 Defaults profile, a prompt window opens every time you save an LMS element or an existing element version. This window allows you to either assign a new version name, if a new version is to be created, or confirm that the modified version is overwritten (that is, updated in place, option `YES`):

```
EDIT-LMS::D:$ASF.ASF.LMSLIB(SAMTEST)S/§ ----------------------- Columns 001 072
 COMMAND===> save                                                SCROLL===> CSR
****** ****************************** top of data ******************************
000001 ******************************************************************
000002 *  PROGRAM TO OPERATE SAM-FILE (OPEN, PUT, CLOSE)                 *
000003 *  JUST TO GET FAMILIAR WITH FILE-OPERATING                       *
000 +-----------------------------------------------------------------------+
000 !                                                                       !
000 ! Version     §                        of type  S LMS library element   !
000 ! named                                                                 !
000 !    SAMTEST                                                            !
000 ! already exists.                                                       !
000 !                                                                       !
000 ! Enter new version name   ________________________ ,                   !
000 !                                                                       !
000 ! or press the PF3 key to abort SAVE operation                          !
000 ! or enter YES to overwrite existing version    ___ .                   !
000 !                                                                       !
000 +-----------------------------------------------------------------------+
000017          OPEN  (R6),EXTEND
000018 *
000019 *  PUT RECORD
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right :s
```

3. If the LMS Override option is set to `NO` in your BS2000 Defaults profile, a similar prompt window to the one in the above example is opened when you wish to save an existing element. In this window, however, you are not offered the option of overriding the old version.
4. Start an edit session with the current version, for example, by selecting it from a list of LMS elements with the `E` line command. Modify the data and issue the `RENAME` command without any parameters. A window opens in which you can specify a version name to be assigned to the previous version. When you store the modified version (with the `SAVE` command, or `END` with `AUTOSAVE=ON`), the new version will have the same version name as the previous version before it was modified. This method is recommended if you always wish to carry over the same name to the current version. This is particularly useful to retain version name `§` (or the character that corresponds to hexadecimal 7C), which always denotes the highest version with respect to the LMS internal sort sequence.

As an alternative to the prompt window, you can use the full `RENAME` function command syntax in the command line of the edit session with the LMS element. Issue either:

```
RENAME,VERSION=vname
```

or:

```
RENAME,  /vname
```

where *vname* is the version name to be assigned to the previous version.

Notes:

1. When specifying version names, you should ensure that the version names are created in ascending order, otherwise the ordering in a list of element versions will not correspond to the "history" of the element.
2. Updating of LMS elements stored in Delta mode is currently not supported by Natural ISPF.

BS2000 Job Information

The job information facility allows you to display information about BS2000 jobs (tasks). You can also use this facility to handle files related to a specific job.

To enter the job information facility

- Select the BS2000 jobs option from the Natural ISPF Main Menu.

  The BS2000 Jobs Entry Panel appears:

```
-------------------------- BS2000 JOBS ENTRY PANEL ----------------------------
COMMAND ===>


   Job Name      ===>
   TSN           ===>
   Type          ===>                  ( B,D,S,T,W,H,K or combination,
                                          or a combination of digits 1..8   )
   Originator    ===>                 ( Related TSN,   for Print Jobs only )
   Out-Device    ===>                 ( Output device, for Print Jobs only )
   Node          ===> 31





Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right :s
```

You can specify the job you wish to handle in the input fields and enter a function command in the command line.

The meaning of the input fields is explained in the following table:

Field	Meaning
`Job Name`	Job name as specified on the `/LOGON` or `/PRINT` command. Leave blank or use strings and wildcards (* and _) to generate a list of jobs matching the job name pattern. See the subsection ***Selection Windows and Wildcards*** in the section *Command Logic*.
`TSN`	Task sequence number assigned to a specific job. This can be numeric or alphanumeric, depending on the version of BS2000 installed and the settings of its system parameters.

`Type`	Type (category) of job. Possible options:
	
	`B` - Active Batch jobs
	
	`D` - Active Dialog tasks
	
	`H` - Tasks in status `HOLD`
	
	`K` - Tasks in status `KEEP`
	
	`S` - Active System tasks
	
	`T` - Active TP tasks
	
	`W` - Tasks in status `WAITING`
	
	`1` - Tasks waiting in job pool: `WT`, `HO` or `DO`(rmant)
	
	`2` - Active tasks

Field	Meaning
	3 - (active) Dialog tasks 4 - Spoolout jobs waiting to be processed 5 - Active spoolout jobs 6 - RBP spoolout jobs (Remote Batch Processing) 7 - RSO spoolout jobs (Remote Spool Output) 8 - Active RBP tasks (Remote Batch Processing) Combinations of types are possible, for example ST selects Types S and T, or 12 selects Types 1 and 2. Combinations of digits and letters (for example, D1) are not accepted.
Originator	For spoolout jobs, TSN of the task that started the job.
Out-Device	For spoolout jobs, name of associated output device.
Node	Node ID of machine on which the job runs. Enter a question mark (?) and press ENTER to list available nodes. Leave blank to select the default node.

Function Commands

The available function commands are:

Command	Object Parameter Syntax
CC	*tsn* NODE=*id*
FOLLOW	*tsn* NODE=*id*
INFORMATION	*tsn* NODE=*id*
LIST	*_* TYPE=*t* ORIG=*o* DEV=*d* USER=*u* NODE=*id*
OUTPUT	*tsn* NODE=*id*
STATUS	*tsn* NODE=*id*

A full description of these commands is contained in the section *Command Reference*. The object parameters correspond to the input fields on the BS2000 Jobs Entry Panel, except the USER parameter on the LIST command. This parameter is relevant only if you are working with administrator authority (TSOS).

Notes:

1. If you issue any of the above function commands from outside the BS2000 jobs facility, you must specify the appropriate object-type parameter (BJ) before the object parameters.
2. If you issue the CC, FOLLOW, STATUS or OUTPUT commands without parameters, the job last submitted during the current Natural ISPF session is assumed.

3. In environments with alphanumeric TSNs, the first object parameter in the above commands can be either a job name or TSN. For the commands `FOLLOW`, `INFORMATION` and `STATUS`, the first object parameter is first checked as TSN, then as job name. Alternatively, you can specify job name with the `JNAME=` parameter, or enclose the parameter in parentheses, indicating that it is a TSN.

Listing Jobs

Lists of jobs currently executing under your user ID can be generated using the `LIST` command and selection criteria in the parameter input fields of the BS2000 Jobs Entry Panel. Alternatively, you can issue the `LIST` command with appropriate parameters from any system screen.

Example: LIST BJ (1)

The following can be listed:

1. Jobs according to prefix or pattern using the wildcards * and _. For example, the following command lists all jobs whose names start with `BS`:

   ```
   LIST BJ BS*
   ```

2. Dialog tasks (TIAM), started tasks, standard batch jobs, spoolout tasks using the `TYPE` parameter. The `TYPE` parameter can be a string of type characters or type digits. You can further restrict the list by specifying a job name pattern. For example, the following command lists all jobs whose names start with `BS` that are in `WAIT` state:

   ```
   LIST BJ BS* TYPE=W
   ```

3. Spoolout tasks associated with a specific output device:

   ```
   LIST BJ BS* DEVICE=ddd
   ```

4. Spoolout tasks originating from a specific job:

   ```
   LIST BJ * ORIGINATOR=tsn
   ```

   An example list of jobs is shown below.

Example: LIST BJ (2)

The following figure shows an example of a list of jobs generated using the command:

```
LIST BJ *
```

```
LIST-BJ:* ---------------------------------------- Row 0 of 3 - Columns 035 076
 COMMAND===>                                                  SCROLL===> CSR
   JOBNAME    TSN TYP CATEGORY  PRIO CPU-MM:SS.CC DEVICE   OR-TSN   SIZE COPIES
** ********************************* top of list ********************************
   EDRGBPA    PEA  2  BATCH      255      0:00.89
   EDRGBPB    PD9  2  BATCH      255      0:00.88
   MZC        W3Y  3  DIALOG     240      3:12.78
** ****************************** bottom of list ******************************
```

The list generated shows all jobs executing under your user ID. The list appears in Editor format. This means you can use Editor `BROWSE` commands (`UP`, `DOWN`, `TOP`, `BOTTOM`, `LEFT`, `RIGHT`, `FIND`, `LOCATE`), as well as `SORT` and `LAYOUT`.

The list is refreshed every time you press ENTER without a command in the command line.

Use the `RIGHT` command to display more information:

```
LIST-BJ:* ---------------------------------------- Row 0 of 3 - Columns 066 107
 COMMAND===>                                                  SCROLL===> CSR
   JOBNAME    TSN TYP CATEGORY  PRIO SIZE COPIES USER-ID    CPU-USED    CPU-MAX
** ********************************* top of list ********************************
   EDRGBPA    PEA  2  BATCH      255            NATISPF        0.89   32767.00
   EDRGBPB    PD9  2  BATCH      255            NATISPF        0.88   32767.00
   MZC        W3Y  3  DIALOG     240            NATISPF      192.78   32767.00
** ****************************** bottom of list ******************************
```

Meaning of column headings (some columns may be empty, depending on the job type and status):

Column	Meaning
JOB-NAME	Job name as defined in the LOGON statement.
TSN	Task sequence number assigned to the job.
TYP	Numeric job type identifier. For example, 2=Batch, 3=Dialog.
CATEGORY	Information on job type and status.
PRIO	Run priority of job/task.
CPU-MM:SS.CC	Amount of CPU time consumed by the job, specified in minutes, seconds and hundredths of a second.
DEVICE	Output device assigned (spoolout jobs only).
OR-TSN	Task sequence number of the job that generated the spoolout (in other cases, blank).
SIZE	Size of spoolout file (spoolout jobs only).
COPIES	Total number of copies to be printed (spoolout jobs only).
USER-ID	User ID under which the job is executing (usually your own user ID).
CPU-USED	Amount of CPU time consumed by the job, in seconds.
CPU-MAX	Maximum amount of CPU time allowed for the job.

Note: You can change the layout of this list according to your needs. For detailed information, see the subsection ***LAYOUT Command for Lists*** in the section *Useful Features*.

Line Commands

You select a job from a list by typing in a line command in the input field preceding the job name and pressing ENTER. Each line command is an abbreviation of a function command:

Line Command	Function	Meaning
CC	CC	Display resulting information kept in job variables (“condition codes”).
FL	FOLLOW	Report job progress in the message line.
IN	INFORMATION	Report current status of job on a full screen.
L	LIST	List related spoolout jobs (if any).
OT	OUTPUT	List output files created by the job and associated by means of a file name pattern. Note that site-specific association rules may have been established by your system administrator. If no such rules apply, the files containing the TSN as part of their name are listed.
ST	STATUS	Show current status of job in the message line.

Line commands can also be used as valid abbreviations of function commands entered in the command line of any system screen.

Local Commands

If you display a list of BS2000 jobs, you can issue the following local commands from the Editor command line in addition to Editor commands: `ALL`, `LAYOUT`, `RELIST` and `SORT`. For detailed information, see the corresponding subsections in the section ***Useful Features***.

Condition Codes of BS2000 Jobs

The function command `CC` provides an overview of job variables related to a specific job. It lists the variable names and the first 64 bytes of their values. The criteria for selecting the variables are as follows:

- job variables containing the TSN of the job as part of their names, enclosed in periods (.);
- job variables containing the TSN at any position in their value;
- if the specified job was the last job created by the Natural ISPF `SUBMIT` function, and if a monitoring job variable was specified when submitting, that monitoring job variable is included;
- only job variables of the current BS2000 user ID are selected.

Notes:

1. The `CC` function command is available only if SNI's software product JV has been installed as part of your BS2000 system environment.
2. Selecting job variables by their values (contents) could cause poor performance for this function. For this reason, your system administrator may have disabled this part of the selection logic, and the job variables will only be selected by name.

BS2000 Job Variables

The BS2000 job variables facility provides functions that include `LIST`, `EDIT`, `RENAME` and `DELETE` job variables. You can also display information on a specific job variable, including its attributes and (if you are authorized to read it) its value. You can create a new job variable with standard attributes simply by editing and saving it, and you can modify the attributes of a job variable using the function command `CHANGE`.

All functionality described in this section is available only if SNI's software product JV has been installed as part of your BS2000 system environment.

To enter the BS2000 job variable maintenance facility

- Select the `JOB-VARS` option from the Natural ISPF Main Menu.

  The BS2000 Job Variables Entry Panel appears:

```
-------------------- BS2000 JOB VARIABLES - ENTRY PANEL -----------------------
COMMAND ===>


   Job variable  ===> *

   Password      ===>                          ( If password protected )
   Node          ===> 31
```

You can specify the name of the intended job variable in the input fields and enter a function command in the command line.

Meaning of the input fields:

Field	Meaning
`Job variable`	Displays the job variable name last used. Select any other job variable by overtyping this name. Leave blank or use strings and wildcards (* and _) to generate a more selective list of variables. See the subsection ***Selection Windows and Wildcards*** in the section *Command Logic*.
`Password`	System password if job variable is protected. Must be specified irrespective of read or write protection.
`Node`	Select Entire System Server node. Enter a question mark (?) and press ENTER to open a window in which all node numbers are scrolled with an `ACTIVE` or `INACTIVE` status report. If you do not specify a node, the default node on the main menu is assumed.

Function Commands

The available function commands are:

Command	Parameter Syntax
`CHANGE`	*jvar-name* `PASSWORD=`*p* `NODE=`*id*, `SHARE=YES/NO RETPD=`*nnnn* `ACCESS=READ/WRITE READPSWD=`*rpwd* `WRITEPSWD=`*wpwd*
`DELETE`	*jvar-name* `PASSWORD=`*p* `NODE=`*id*
`EDIT`	*jvar-name* `PASSWORD=`*p* `NODE=`*id*
`INFORM`	*jvar-name* `PASSWORD=`*p* `NODE=`*id*
`LIST`	`*_*` `NODE=`*id*
`RENAME`	*jvar-name* `PASSWORD=`*p* `NODE=`*id*

Note: If you issue any of the above function commands from outside the job variables facility, you must specify the object-type parameter `JV` before the object parameters.

A full description of these commands is contained in the section ***Command Reference***. The object parameters correspond to the input fields on the BS2000 Job Variables Entry Panel.

Listing Job Variables Selected by a Path Name Pattern

The specification can optionally contain a catalog ID and a different user ID. If so, only shared job variables are listed. If catalog ID and user ID are omitted, or if the specified user ID is identical to the current one, the list contains all job variables owned by the current user ID that fit into the path name pattern.

Notes:

1. If a catalog ID is specified, it must be embedded between colons (:).
2. If a user ID is specified, it must be preceded by the dollar sign ($).

Example: LIST JV (1)

1. `LIST JV ASM*`

   lists all job variables with prefix ASM and owned by the current user ID.
2. `LIST JV $DEMO.*ASM*`

   lists all public job variables owned by the user ID `DEMO` and containing the string `ASM` in their names.
3. `LIST JV :K:$DEMO.*`

   lists all public job variables owned by the user ID `DEMO` found in catalog `K`.
4. `LIST JV $DEMO.* NODE=38`

   lists all public job variables owned by the user ID `DEMO`, in a multi-node environment.

An example of a list of BS2000 job variables follows.

Example: LIST JV (2)

The following figure shows an example of a list of BS2000 files generated using the command:

```
LIST JV $DC1.*
```

```
LIST-JV:$DC1.* ---------------------------------- Row 0 of 8 - Columns 056 064
 COMMAND===>                                                  SCROLL===> CSR
   PATH-NAME OF JOB VARIABLE
** ************************* top of list ************************
   :D:$DC1.ASF.TEST.RD-PW
   :D:$DC1.ASF.TEST.WR-PW
   :D:$DC1.INST.JV.NPR
   :D:$DC1.JV.ASF.NINU
   :D:$DC1.JV.SUBMIT.TEST
   :D:$DC1.JV.TEST.NEU
   :D:$DC1.MRS.TEST2
   :D:$DC1.TEMP
** ************************ bottom of list ************************
```

The list appears in Natural ISPF Editor format in BROWSE mode. This means you can use all available BROWSE commands (UP, DOWN, BOTTOM, TOP, LEFT, RIGHT, FIND, LOCATE), and in addition the command SORT.

The list is static: it is filled once and is not updated automatically, for example, after a new job variable has been created.

Lists of job variables can also be created as a result of a CC command, showing the monitoring job variables ("condition codes") of a specific BS2000 job (see the subsection *Condition Codes of BS2000 Jobs*); in this case, the list also contains the first characters of the values stored in the job variables listed.

Meaning of column headings:

Column	Meaning
PATH-NAME	Full path name including catalog ID and user ID specification
CONTENT	First 64 bytes of job variable value (CC-type lists only)

Line Commands

Select a job variable from a list by typing in a line command in the input field preceding the path name and pressing ENTER. Each line command is an abbreviation of a function command. Line commands can also be used as valid abbreviations of function commands entered in the command line.

Available line commands are:

Line Command	Function	Meaning
CH	CHANGE	Change attributes of job variable, for example, define a password.
D	DELETE	Erase job variable from system.
E	EDIT	Start editing job variable, in order to modify its contents or to use cross-copy operations.
I	INFORM	Show attributes and full contents of job variable.
R	RENAME	Rename the job variable.

Local Commands

In Edit Mode:

If you display a job variable in Editor format in EDIT mode, the following local command is available in addition to Editor commands:

Command	Meaning
PASSWORD *password*	If the job variable is password-protected, use this command to enter the valid password in order to update the variable. If you enter the PASSWORD command without parameter, a window prompts you for the password. Password input in the window is invisible.

In List Mode:

If you display lists of job variables in Editor format, you can issue the following local commands in addition to Editor scroll commands: ALL, LAYOUT, RELIST and SORT. For detailed information, see the corresponding subsections of the section ***Useful Features***.

Information about a BS2000 Job Variable

An information screen similar to the following appears as a result of the command:

```
INFORMATION JV $DC1.TEMP
```

```
------------------------- JOB VARIABLE INFORMATION ----------------------------
COMMAND ===>


  JOB VARIABLE NAME: $DC1.TEMP

  HISTORY                                        SECURITY
    DATE OF CREATION / UPDATE :  1994-03-02        SHARE          :  YES
    DATE OF EXPIRATION        :  1994-03-02        ACCESS         :  WRITE
                                                   READ-PASSWORD  :  NONE
  CURRENT LENGTH IN BYTES     :  100               WRITE-PASSWORD :  YES

  COLUMNS  CURRENT CONTENTS
           ----+----#----+----#----+----#----+----#----+----#----+----#----
  001-064: P.ADALOD,FNR=140,LIST=L.ADALOD.NOM140,NAME=NOM133-FUSER,INPUT=NO
  065-128: M133.FUSER,DEV=T9G,VOL=NOM133,FSEQ=3
```

This screen displays current attributes of job variable `$DC1.TEMP`. Its current value is also displayed, if you are authorized to read it; otherwise the subsection titled `CURRENT CONTENTS` is omitted without notice. The data displayed are self-explanatory.

Local Commands for Job Variable Information

There are two local commands available from this screen:

Local Command	Meaning
`PASSWORD` *`password`*	Supplies the correct password for read access, in order to display the subsection `CURRENT CONTENTS`. If you enter the `PASSWORD` command without parameter, a window prompts you for the password. Password input in the window is invisible.
`REFRESH`	Reevaluates data to display latest attributes.

Editing a Job Variable

When a job variable is edited, its contents are read into a single line of an edit session. A screen appears similar to the following:

```
EDIT-JV:$DC1.TEMP ---------------------------------------------- Columns 001 072
 COMMAND===>                                                    SCROLL===> CSR
 =cols ----+----1----+----2----+----3----+----4----+----5----+----6----+----7--
****** ****************************** top of data ******************************
000001 ADALOD,FNR=140,LIST=L.ADALOD.NOM140,NAME=NOM133-FUSER,INPUT=NOM133.FUS
****** ***************************** bottom of data ****************************
```

Use the Editor commands `RIGHT` and `LEFT` (usually assigned to PF11 and PF10 respectively) to scroll horizontally (the length of the line is restricted to 256 characters).

You can create more than one line of edited data temporarily (this can be useful during complex editing operations), but the `SAVE` command is rejected if the session contains more than one line of data.

Changing the Attributes of a Job Variable

The `CHANGE` function command enables you to modify attributes of a job variable. You can specify one or more new attribute values as function parameters, using keyword parameters (see Example 3).

If you issue the `CHANGE` command with or without object parameters, but without any function parameters or if you issue the line command `CH` from a list of job variables, a window opens that displays the current attribute values, except for passwords. To modify one or more attributes, overtype the values in the input fields and press ENTER.

Example 1:

The following window opens if you enter the command:

```
CHANGE JV MY.STANDARD.JVAR
```

```
-------------------- BS2000 JOB VARIABLES - ENTRY PANEL -----------------------
COMMAND ===> CHANGE JV MY.STANDARD.JVAR


   Job variable  ===> *

   Password      ===>                          ( If password protected )
   Nod +-----------CHANGE-JOBVAR: $NATISPF.MY.STANDARD.JVAR------------+
       !                                                                !
       ! Modify attributes as desired:                                 !
       !                                                                !
       !  SHARED JOB-VARIABLE :  YES       (YES/NO)                    !
       !  ACCESS-TYPE         :  WRITE     (READ/WRITE)                !
       !  RETENTION-PERIOD    :  _____     (in days)                   !
       !  READ-PASSWORD       :                                        !
       !  WRITE-PASSWORD      :                                        !
       !                                                                !
       +----------------------------------------------------------------+




Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Meaning of the input fields:

Field	Keyword	Meaning
SHARED JOB-VARIABLE	SHARE	Enter Y or YES to make the job variable accessible to other users. Enter N or NO to restrict access to the owner user ID.
ACCESS-TYPE	ACCESS	Enter W or WRITE to allow write access to the specified job variable, which also implicitly includes authorization for read access. Enter R or READ to allow read access only.
RETENTION-PERIOD	RETPD	Enter a number representing a retention period (in days) that is to be assigned to the specified job variable: This means it can neither be modified nor deleted within the specified period of time.
READ-PASSWORD	READPSWD	Enter a password to protect the job variable against unauthorized read accesses. Enter *NONE to reset an existing password, provided that you specified that password together with the current CHANGE function.
WRITE-PASSWORD	WRITEPSWD	Enter a password to protect the job variable against unauthorized modification or deletion. Enter *NONE to reset an existing password, provided that you specified that password together with the current function.

Example 2:

If the job variable to be changed is password-protected, you must also correctly specify the password. For example, the following window opens if you issue the command:

```
CHANGE JV JV.MIDI.2PTG PASSW=BOBO
```

```
 -------------------- BS2000 JOB VARIABLES - ENTRY PANEL -----------------------
 COMMAND ===> CHANGE JV JV.MIDI.2PTG PASSW=BOBO


    Job variable  ===> *

    Password      ===>                          ( If password protected )
    Nod +-------------CHANGE-JOBVAR: $NATISPF.JV.MIDI.2PTG--------------+
        !                                                                !
        ! Modify attributes as desired:                                 !
        !                                                                !
        !  SHARED JOB-VARIABLE :  YES       (YES/NO)                    !
        !  ACCESS-TYPE         :  WRITE     (READ/WRITE)                !
        !  RETENTION-PERIOD    :  _____     (in days)                   !
        !  READ-PASSWORD       :                                        !
        !  WRITE-PASSWORD      :            (Enter *NONE to reset)      !
        !                                                                !
        +----------------------------------------------------------------+




 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
       Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Enter `*NONE` to reset the password.

To avoid visible input of the old password on your screen, issue the `CHANGE` command from the BS2000 Files Entry Panel and specify the password in the corresponding input field of the `CHANGE` window.

Example 3:

To avoid being prompted for the new attribute values (for example, during execution of a command script), specify them as function parameters using the keywords indicated above. For example, the command:

```
CH JV PARMJV, SHARE=Y WRITEPSWD=BOHR
```

makes the job variable `PARMJV` shared, that is accessible by other users, and protects it against unauthorized modification.

Console Display

A console display is available for BS2000 environments. For a description, see the subsection ***Console Display*** in the section *z/OS Objects*.

13 Command Reference

- Note about Command Syntax Symbols 358
- Session Commands - Description 358
- Function Commands - Description 381
- Local Commands - Description 405

This chapter lists, explains and provides examples of all Natural ISPF session, function and local commands.

Before consulting any command described here, you should read section *Command Logic*.

Note: Bear in mind that you can issue session and function commands from any system screen. You can issue local commands only from Editor screens (lists of object names, objects in `EDIT` or `BROWSE` mode).

Editor commands are described in section *Editor*.

Note about Command Syntax Symbols

The following symbols are used in the description of command syntax:

Symbol	Name	Meaning
[ ]	Square brackets	Enclosed elements are optional.
{ }	Braces	Only one of the enclosed elements can be specified.
[{ }]		Enclosed elements are optional, but if you do make a selection, only one element may be specified.
;	Semicolon	Assumed command delimiter when entering multiple commands (see your user profile).

Note: Do not type in brackets or braces as part of the command input.

Session Commands - Description

With Natural ISPF session commands, you can control your Natural ISPF session(s) and navigate within the system. Session commands can be issued from any system screen.

This subsection describes session commands in alphabetical order.

ACTIVITY

Opens a window with a list of all Natural ISPF sessions you have started. The current session appears highlighted.

The command format is:

```
ACTIVITY
```

Example:

You can issue any of the following line commands for any session in the list:

Command	Function
P	Makes the selected session the current session (see the POP session command).
S	Moves the selected session to the other half of the screen in split-screen mode.
-	Terminates the selected session(s); if the session is an edit session, it is ended without saving modifications.

When you close the activity display with ENTER or PF3, the session selected with the P line command will be the current (active) session.

You can give a session a short name by typing the short name in the appropriate field in the column headed Name in the activity window. This serves for easy identification of the session in a subsequent POP command. Alternatively, you can type a PF key name in the Name field. Pressing this PF key calls the associated session to display. This PF key is valid only until you change it or log off from Natural ISPF, and overrides the PF definition in your user profile.

The following is an example of an ACTIVITY window:

```
S*>>>EDIT-NAT:NSPFHELP(ALL)-Text->Report-Free-42K ------------- Columns 001 072
 COMMAND===> activity                                            SCROLL===> CSR
****** ****************************** top of data ******************************
 +------------------------N-ISPF ACTIVITY TABLE-------------------------+
 !                                                                      !
 !  Cmd Name Session                                                    !
 !  --- ---- ---------------------------------------------------------- !
 !      PF6  WORKPOOL - ENTRY PANEL                                     ! e  to
 !      PF7  LIST-PDS:BRY.COMN.SOURCE(*)                                ! list.
 !      PF5  NATURAL VIEW - ENTRY PANEL                                 ! uting
 !      F1   EDIT-NAT:NSPFHELP(ALL)-Text                                !
 !                                                                      !
 !                                                                      !
 !                                                                      !
 +----------------------------------------------------------------------+
000130 &L
000140   ALL ED
000150   The command edit is executed for  all  members  of  the  list.  If  an
000160  #end§command usually assigned to #PF3§ is  entered,  the  current  edit
000170   session is terminated and an edit session for the next§member from the
000180   list is opened.
000190 &L
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

The window shows four Natural ISPF sessions, the Natural `EDIT` session being the current session. You can call this session to display from any screen using the command:

```
POP F1
```

If you press PF6 from any screen, the Workpool Entry Panel becomes the current session.

APPLICATION

Activates the Natural application specified in the command parameters.

The command format is:

```
APPLICATION library[startup-program[parameters]]
```

where:

Parameter	Meaning
library	The Natural library to be invoked.
startup-program	The program to be started.
parameters	First parameter(s) of the startup program

Examples:

Command	Function
APP SYSERR	Displays the Natural SYSERR Utility Menu.
APP SYSDDM MENU	Displays the Natural View Maintenance Menu.
APP SYSMAIN MENU C	Selects the COPY function on the SYSMAIN menu.

The APPLICATION command is especially useful for the system administrator in menu definition, allowing applications written in Natural to be integrated into the Natural ISPF menu structure (see also the section *Menu Maintenance* in the *Natural ISPF Administration Guide*).

BPSTAT

Invokes the Natural Edit Buffer Pool Utility (see the section *Buffer Pool and Recovery Files* in the *Natural ISPF Administration Guide*).

The command format is:

```
BPSTAT
```

BREAK

Modifies the break interval defined in your user profile. The value specified is the number of trace windows to be displayed before you are prompted to interrupt processing: BREAK PROCESS (Y/N). For example, after the command BREAK 3, you are given the opportunity to interrupt processing every 3 trace windows.

The command format is:

```
BREAK {n }
      {OFF}
```

where *n* stands for the number of trace windows to be displayed before the break in processing is allowed, and OFF disables the break feature.

For more information, see the subsection ***Break in Processing*** in the section *Useful Features*.

CALCULATOR

Invokes the calculator on your screen that allows you to perform computation in numeric or hexadecimal mode.

The command format is:

```
CALCULATOR
```

For more information, see the subsection ***Using the Calculator*** in the section *Useful Features*.

CHARPROF

Invokes the Magic-Character definition screen in your user profile. You can define magic characters for any number of Natural ISPF commands (see the subsection ***Magic-Character Definition*** in the section *Profile Maintenance*).

The command format is:

```
CHARPROF
```

COLPROF

Defines the colors to be used by the Natural ISPF browser, which is responsible for all `LIST`, `BROWSE` and `EDIT` sessions. The colors used in other screens cannot be modified with this profile option.

The command format is:

```
COLPROF
```

For more information, see the subsection ***Editor Color Definition*** in the section *Profile Maintenance*.

CONFIRM

Specifies whether confirmation windows are used when performing functions such as `DELETE`, `COMPRESS`, `CATALOG` and `UNCATALOG`. You can use this command to override the setting in your user profile.

The command format is:

```
CONFIRM [ON]
        [OFF]
        [LONG]
        [SHORT]
```

Meaning of the parameters:

Parameter	Meaning
ON (default)	Used after a CONFIRM OFF command issued from the same system screen. Reactivates the confirmation feature according to the setting in your user profile (LONG or SHORT).
OFF	Deactivates the confirmation feature for the current screen. Useful when deleting multiple items from a list in a single input operation with multiple D line commands. The message CONFIRM OFF is displayed in the message line. As soon as another system screen is invoked from the current one, an implicit CONFIRM ON is performed.
LONG	Confirmation windows are used. Confirm by entering the object name in the window.
SHORT	Confirmation windows are used. Confirm by entering Y in the window.

If you issue the CONFIRM command without parameters, the default is ON, taking LONG or SHORT from your user profile.

CONTINUE

The command format is:

```
CONTINUE
```

The CONTINUE command can be used in command scripts to gain more flexible control in error situations. If no CONTINUE statement is in the command script, the script is set to PAUSE mode after an error.

If a CONTINUE statement (which can be compared to a label) is in the script and an error occurs, the following actions are taken:

- RECORD ON is set internally if not activated by the user.
- The command causing the error and the message is recorded.
- All lines of the script until the next CONTINUE command are deleted and execution of these lines is skipped.
- Processing continues with the next CONTINUE statement. All following statements are executed.
- Termination resets RECORD to its previous value and informs the user if an error has occurred.

Example:

```
KEYS 3 PAUSE
HELP VERIFY
MESSAGE 7480
TECH
........
CONTINUE
REMARK PROCESSING WILL CONTINUE HERE AFTER ERROR
KEYS 3 INITIAL
```

The above script modifies the user profile, and by using the `CONTINUE` command, it makes sure that PF3 is reset to the value from the user profile after execution of the script, even if errors have occurred during execution of the script.

CONTROL

Applies to CA Panvalet and CA Librarian only. `CONTROL OFF` suppresses the automatic control cards passed to CA Panvalet or CA Librarian when a member is saved. Only the edited data (which may include user-defined control cards) are sent to CA Panvalet or CA Librarian.

The command format is:

```
CONTROL [ON]
        [OFF]
```

Meaning of the parameters:

Parameter	Meaning
`ON` (default)	Re-activates the automatic control cards
`OFF` cards in the member	Suppresses the automatic control cards. You can define your own control

You must issue the `CONTROL` command before opening the new Editor session.

DEFPROF

Displays your User Defaults definition screen in your user profile. You can modify any default (see the subsection ***User Defaults*** in the section *Profile Maintenance*).

The command format is:

```
DEFPROF
```

DEFBS2PROF

Displays the general defaults specific to BS2000 sites in your user profile. You can modify any default as required (see the subsection ***General Defaults*** in the section *Profile Maintenance*).

The command format is:

```
DEFBS2PROF
```

Note that this command is available only if the BS2000 subsystem is enabled at your site.

DEFSUBPROF

Displays the defaults specific to submitting BS2000 jobs in your user profile. You can modify any default as required (see the subsection ***Submit Defaults*** in the section *Profile Maintenance*).

The command format is:

```
DEFSUBPROF
```

Note that this command is available only if the BS2000 subsystem is enabled at your site.

EDITPROF

Invokes the Editor Profile definition screen in your user profile. You can modify your Editor profile (see the subsection ***Editor Profile*** in the section *Profile Maintenance*).

The command format is:

```
EDITPROF
```

Note: Any modifications made to your Editor profile using this command only affect subsequently opened edit sessions, not already existing ones.

END

Returns you to the previous screen. Issued from the Main Menu, `END` terminates the Natural ISPF session. Issued from the last session, this command returns you to Natural.

This command is usually assigned to PF3.

The command format is:

```
END
```

FIN

Terminates all Natural ISPF sessions and Natural immediately. The command corresponds to `LOGOFF IMM;FIN`.

The command format is:

```
FIN
```

FLIP

Switches between PF key display (from PF1-PF12 to PF13-PF24 or vice versa). See also the `KEYS` session command.

The command format is:

```
FLIP
```

GENNCP

Starts the generation of a command processor for Natural ISPF. For more information, see the subsection *NCP Concept* in the section *System Configuration* of the *Natural ISPF Administration Guide*.

The command format is:

```
GENNCP
```

HELP

Issued without parameter, displays a screen-related help text. The `HELP` command is usually assigned to PF1.

The help text is displayed according to the following hierarchy:

- Menu help text defined by your system administrator;
- Help text from the online help facility.

You can also issue the `HELP` command with a parameter indicating the topic on which help is required from any system screen in the format

```
HELP [INDEX]
     [object]
     [:C]
```

where:

Parameter	Meaning
INDEX	Lists all objects for which there is a help text. You specify any object in a HELP command to display the related help text.
object	The object for which a help text is required. This can be any object listed by the HELP INDEX command. Enter the asterisk wildcard (*) and press ENTER to generate selection lists of objects with the same prefix (see examples below)
:C	Is substituted by the string marked by the cursor (see the subsection ***Cursor-Sensitive String Selection*** in the section *Useful Features*). If you use the :C directive from a help screen, you need not type in the HELP command keyword.

Note: In addition to the HELP command, most menus offer the command NHLP which takes you to the Natural help system directly.

Examples:

Command	Function
HELP LIST	Displays the help text for the command LIST.
HELP L*	Displays a selection list of all items and commands starting with L for which there is a help text. You can select any item from the list to display the associated help text.
HELP FEATURES	Displays help text for special Natural ISPF features.
HELP *nnnn*	Displays a help text for error message number nnnn.
HELP :C	With cursor on string EDIT, executes the command HELP EDIT.

Note: Items in the body of a help text for which there is a separate help appear in reverse video (on color terminals: yellow).

KEYS

Used without parameter, displays current PF key assignments in your user profile. You can modify the assignments and specify whether and which PF keys are displayed on system screens (see the subsection ***PF-Key Definition*** in the section *Profile Maintenance*).

The command format is:

```
KEYS   [ON]
       [OFF] [string]
       [FIRST] [INITIAL]
       [LAST]
       [n]
```

The string and INITIAL parameters are only valid in conjunction with *n* (see below).

The parameters have the following meaning:

Parameter	Meaning
OFF	Eliminates the PF key line and associated command line from display.
ON (default)	Displays the PF key line and associated command line.
FIRST	Displays PF1 - PF12.
LAST	Displays PF13 - PF24.
n string	Assigns a command string to PF key *n*. If the string consists of more than one command, you must separate the commands with two command delimiters. If you use only one command delimiter here, Natural ISPF executes the KEYS command up to the first delimiter, and then executes each command in the string in turn (see also the subsection ***PF Key Assignments*** in the section *Command Logic*).
n INITIAL	Resets PF key *n* to its initial value as defined in the user profile. This is especially useful for resetting a PF key in a command script to its original value after temporary modification.

Examples:

Command	Function
KEYS 24 SEP;;SPLIT	Assigns the command sequence SEPARATE;SPLIT to PF24 (note the use of the double command delimiter).
KEYS 3 PAUSE	Assigns the command PAUSE to PF3
... CONTINUE KEYS 3 INITIAL	... and resets it later to its initial value.

LAST

Redisplays the last ten (10) commands entered via the keyboard in this session (but see note below). You can select any command for reexecution, or modify any command by overtyping it before reexecution. Select the required command from the list by placing the cursor on it and pressing ENTER.

If a command is executed again, it is always put on top of the last command buffer. This keeps the commands used most in the buffer. Additionally, you can also delete commands from the last buffer. Commands not used again can be deleted; this avoids automatic deletion of commands likely to be reused.

The command format is:

```
LAST
```

Note: To be stored for redisplay, the command must consist of at least two words. Commands entered by PF key, magic character or selected from a selection window are not stored. This also applies to commands entered using menu options.

LOGOFF

The `LOGOFF` session command allows you to terminate Natural ISPF even if you are working with several sessions.

The command format is:

```
LOGOFF [IMM] [;Natural-command]
```

- If your Editor profile has `AUTOSAVE=OFF` and you issue the `LOGOFF` command without parameters, logoff processing is interrupted if an Editor session with modified data is detected.
- If your Editor profile has `AUTOSAVE=ON` and you issue the `LOGOFF` command without parameters, all modifications are saved during logoff processing.

If you wish to terminate Natural ISPF immediately, that is, regardless of any modifications, you can use the `IMM` (`IMMEDIATE`) parameter. All sessions will be closed without saving and Natural ISPF will be terminated.

You can concatenate the `LOGOFF` command with any valid Natural command.

If Natural ISPF is the only user interface installed, all sessions will be closed and you return to the Natural ISPF main menu.

Examples:

Command	Function
`LOGOFF;FIN`	Terminates Natural ISPF as described above and terminates the Natural session.
`LOGOFF IMM;FIN`	Terminates Natural ISPF immediately and terminates the Natural session.
`LOGOFF IMM;SYSPROF`	Terminates Natural ISPF immediately and logs on to library `SYSPROF`.

LOGON

Logs on to the specified Natural library as a default library from any system screen.

The command format is:

```
LOGON  library-name
```

The new library is addressed as the default library by function commands for Natural objects and as the library in the session command Natural.

Note: Use this command especially if Natural ISPF is installed as the only user interface, because in this case, the Natural `LOGON` command (`NAT LOGON`) does not apply.

MACPARM

The `MACPARM` command is used in command scripts to stack data which is read by a macro using an input statement later in the command script.

This avoids prompting by the macro for parameters, when using macros in a command script. The `MACPARM` command must be the only command in a source line.

The command format is:

```
MACPARM  p1
```

where:

Parameter	Meaning
p1	Maximum length of this parameter is 50 bytes and it can contain blanks.

Examples:

Command	Function
`MACPARM LS PDS JW(A*) PLAY MAC MAC1`	Passes command `LS PDS JW(*)` to macro `MAC1`.

Another useful example can be found in member `VERIFY` in our example library.

MACRO

Specifies whether macros are expanded in Natural programs and other sources that include inline macros or the `INCLUDE-MACRO` statement. For details, see the section *Macro Facility* in the *Natural ISPF Programmer's Guide*.

The command format is:

```
MACRO [ON]
      [OFF]
```

where `ON` specifies macro expansion and `OFF` specifies non-expansion. Default is `ON`.

MENU

Activates a defined menu. For example, the command `MENU MAIN` displays the Natural ISPF Main Menu. The name of the menu must be defined in the menu maintenance facility, see the section *Menu Maintenance* in the *Natural ISPF Administration Guide*.

The command format is:

```
MENU name
```

where name specifies the menu name as defined in the menu maintenance facility.

MESSAGE

The `MESSAGE` command can be used in command scripts to display a text during execution of a script on the screen and to interrupt the active command script. The `MESSAGE` command must be the only command in a source line.

The command format is:

```
MESSAGE  nnnn[p1,p2 ... pm]
```

where:

Parameter	Meaning
nnnn	Must be a 4-digit error message number. First, the user library SYSISPFU is searched for the message text. If it does not exist, it is taken from the system library `SYSISPS1`.
p1, ... *pm*	Optional parameters which are used to replace variable parameters (`:1: ..... :m:`) in the text. Parameters must be separated with your parameter delimiter, usually a comma , and can contain blanks.

Examples:

Command	Function
`MESSAGE 6812,MYPROG`	Results in the following message if no text for this number is available in the user library SYSISPFU: `Member MYPROG not found.`
`MESSAGE 6809,Please enter some text`	Results in the following message if no text for this number is available in the user library `SYSISPFU`: `Please enter some text`

Another useful example can be found in member `VERIFY` in our example library.

NATDEF

Invokes the Natural Defaults definition screen in your user profile which allows you to set several defaults that affect your working environment in Natural ISPF. For more information, see the subsection ***Natural Defaults*** in the section *Profile Maintenance*.

The command format is:

```
NATDEF
```

NATP-LOG

When you enter this command, a window prompts you for your user ID, password and Entire System Server node ID. Performs a logon to Entire System Server on the specified node.

The command format is:

```
NATP-LOG
```

NATURAL

Suspends your Natural ISPF session and invokes the Natural environment. You can return to your Natural ISPF session at any time by issuing the command `SPF`.

The command format is:

```
NATURAL [Natural-command and parameters]
```

If the Natural command is issued with parameters, the parameters are executed by Natural and you are automatically returned to Natural ISPF.

NEWS

Displays a summary of changes in the current release of Natural ISPF (new features, enhancements, etc.), selectable by topic after the `NEWS` command has been issued.

The command format is:

```
NEWS
```

NKEY

From Version 2.1.1 onward, this command is no longer needed and has no effect. However, it is still accepted for compatibility reasons.

The command format is:

```
NKEY [ON]
     [OFF]
```

NODE

Selects the Entire System Server node ID specified with the command keyword. This command changes the default node and has no effect on existing sessions.

The command format is:

```
NODE id
```

NODES

Invokes the Entire System Server node table, which offers active help for the `NODE` field in several Natural ISPF screens. For more information, see the subsection *Entire System Server Node Table* in the section *System Configuration* of the *Natural ISPF Administration Guide*.

The command format is:

```
NODES
```

NSPROF

Displays the NSPF Parameters definition screen. You can modify any parameter (see the subsection *Natural ISPF Parameters* in the section *System Configuration* in the *Natural ISPF Administration Guide*).

The command format is:

```
NSPROF
```

OPERATOR

Signals Natural ISPF that the command entered in the same input operation is an operator command. For easier operator command input, you are advised to assign this command to a magic character (see the subsection ***Magic-Character Definition*** in the section *Profile Maintenance*).

The command format is:

```
OPERATOR operator-command
```

Note: If the command line is too short for an operator command, type a plus sign + at the end of the command line. This invokes a prompt window in which you can enter an operator command of up to 80 bytes long. Also, you are advised to use the plus sign if the operator command string contains special characters that have a special function in Natural ISPF (magic character, delimiter), as the window does not interpret special characters.

PAUSE

The `PAUSE` session command can be written in a command script that is executed with the `PLAY` function command. It must be either the only or the last command in a source line of the script. When the script is executed, it is interrupted at the place of the `PAUSE` command.

To continue the script, you can issue the `PAUSE` command from the Natural ISPF command line. You can also use the `PAUSE` command if a command script is interrupted due to an invalid command: the command appears in the command line. You can correct the command, press ENTER to reexecute it, and then issue the `PAUSE` command to continue the script.

Note: The `PAUSE` command must always be the last command or the only command in a script line, otherwise it is ignored.

The command format is:

```
PAUSE
```

See also the subsection ***Executing Command Scripts*** in the section *Useful Features*.

PLAY OFF

If the execution of a Natural ISPF command script is interrupted due to the `PAUSE` command, you can cancel the script by issuing the command `PLAY OFF`. The script entry is also deleted from the workpool.

The command format is:

```
PLAY OFF
```

See also the subsection ***Executing Command Scripts*** in the section *Useful Features*.

POP

Selects the specified Natural ISPF session for work.

The command format is:

```
POP [string]
```

where *`string`* identifies the required session by the short name assigned in the `ACTIVITY` window or by any string from the session's header line. For example, the command

```
POP PDS
```

selects the Natural ISPF session that might have `EDIT-PDS:NSP.V100.JCL(JOB1)` as its header. If you use the `POP` command without a parameter, you can select a session by placing the cursor on it (usually on its header line). If the cursor remains in the current session, a window with selectable sessions appears. The current session is highlighted, the other sessions are numbered. Press ENTER to select the current session, or select any other session by entering its number in the input field.

PROFILES

Invokes the user Profile Menu. You can select any option (see the section ***Profile Maintenance***).

The command format is:

```
PROFILES
```

RECORD

Activates the recording of Natural ISPF commands issued and the resulting error messages. The recorded messages are held in the User Workpool in the member `RECORD`. This member can be played (the commands are executed, see the `PLAY` function command).

The command format is:

```
RECORD [ON]
       [OFF]
```

`RECORD` or `RECORD ON` activates recording of commands. `RECORD OFF` ends recording. If a command causes an error, the error message is also recorded, but prefixed by two asterisks (**). The `PLAY` command ignores these lines.

RECOVER

If you lose files for any reason (for example, after an abnormal termination), Natural ISPF notifies you with a message the next time you log on. The `RECOVER` command displays a list of the lost files which you can select with a line command for further editing and saving, browsing or deleting.

The command format is:

```
RECOVER
```

See also the subsection ***Recovery*** in the section *Useful Features*.

REFRESH

Refreshes screen display.

The command format is:

```
REFRESH
```

REMARK

The `REMARK` command is used in command scripts for documentation purposes and must be the only command in a source line.

The command format is:

```
REMARK text
```

Example:

```
REMARK The following command extracts all members
REMARK including the string ADABASLIS
```

```
T PDS JW(*) SC=ADABAS
```

RETURN

Returns directly to the Natural ISPF Main Menu of the current session. If you issue this command from an edit session, Natural ISPF prompts you to `SAVE` or `CANCEL` any modifications.

The command format is:

```
RETURN
```

SEND

Allows you to send a message to up to five TSO, TIAM or Com-plete users anywhere within the network. A window opens prompting you for user ID(s), destination node number and message text. See also the subsection ***Message Switching*** in the section *Useful Features*.

The command format is:

```
SEND
```

SEPARATE

Splits the current screen from its session to create two separate and distinct sessions, for example, after selecting a member from a list for an `EDIT` operation: the edit screen and the list become two separate sessions.

The command format is:

```
SEPARATE
```

Useful in conjunction with the `SPLIT` session command to display two screens from the same session, for example, an edit session and the list of objects from which it was started. Use the command sequence:

```
SEP;SPLIT
```

See the subsection ***Multi-Session Operations*** in the section *Useful Features* for a more detailed example.

SHORTLIB

Displays the Short Libraries definition screen for library names in your user profile. You can define a two-character alias for any library (see the subsection ***Library Definition*** in the section *Profile Maintenance*).

The command format is:

```
SHORTLIB
```

SPLIT

Switches your session to split-screen mode. Your screen is divided horizontally into two at the cursor position or at the line number specified as command parameter. An existing session is redisplayed in the lower part of your screen. If there are no other existing sessions, a new one is started.

The command format is:

```
SPLIT [n]
```

where *n* is the line number at which the screen is to be split (n must be smaller than the available number of lines for the screen). The `SPLIT` command is usually assigned to PF2.

SUSPEND

Suspends the current Natural ISPF session and starts a new one.

The command format is:

```
SUSPEND
```

The `SUSPEND` command is usually assigned to PF4.

SWAP

When in split-screen mode, makes the other session active. The cursor moves to the first input field of the activated session. When issued from a session in full-screen mode, recalls a suspended session in wrap-around fashion.

The command format is:

```
SWAP
```

The `SWAP` command is usually assigned to `PF9`.

TECH

Displays technical information on Natural ISPF.

The command format is:

```
TECH
```

TRACE

Modifies the trace interval in seconds defined in your user profile. When you issue a command, a message appears in a window if the function is not yet complete after the specified interval. The message informs you of Natural ISPF activity, for example, `PROCESSING 44 MEMBERS`. The message is updated at the specified interval.

The command format is:

```
TRACE [n]
      [OFF]
```

where *n* stands for the interval in seconds, and `OFF` disables the trace feature.

Note: If you specify `TRACE 0`, the trace function is also disabled.

UINFO

Displays site-specific information maintained by your system administrator. For more details, see the subsection ***Online Technical and Site-Specific Information*** in the section *Useful Features* (see also the section *Site-specific Online Information* in the *Natural ISPF Administration Guide*).

The command format is:

```
UINFO
```

UNZOOM

Reverses the effect of a previous `ZOOM` command (see the `ZOOM` command below).

The command format is:

```
UNZOOM
```

VERSIONS

Specifies versioning for PDS, z/VSE and Natural members. This command overrides the setting of the `VERSIONS` parameter in your user profile (see the subsection ***User Defaults*** in the section *Profile Maintenance*).

The command format is:

```
VERSIONS [ON]
         [OFF]
```

where `OFF` deactivates versioning and `ON` (re-)activates versioning. `ON` is the default.

ZOOM

Eliminates the header lines of suspended sessions from your screen. When in split-screen mode, a second `ZOOM` command displays the current session in full-screen mode. You can return to split-screen mode using the `UNZOOM` command.

The command format is:

```
ZOOM
```

Function Commands - Description

Function commands perform functions on Natural ISPF objects. You can issue a function command in three different ways:

- Enter a command in the command line and parameters in the parameter fields of the object Entry Panel as appropriate;
- Use a line command to select an object from a list;
- Enter command syntax in the command line of any screen in the format

```
COMMAND object-type object-parameters,function-parameters
```

A description of the command parameters follows.

Object Types

Abbreviations of object types used in command syntax are:

In all environments	Explanation
BPF	Buffer pool files
BPR	Recovery files
CON	Console
CNF	Configuration members
CTN	Incore database container
DOC	Con-nect documents
ERR	Natural error messages
MAC	Macro objects
MNU	Menus
MV	Previous versions of all types
N	Natural objects
NLI	Natural libraries
NV	Natural versions
O	Output in workpool
PRD	Predict descriptions
R	Recovery files
SET	Predict cross-reference set
USR	Users
V	Views

Additional objects under z/OS	Explanation
A	Active jobs
D	Datasets
CST	Module CSECTs
ENQ	System enqueues
J	Jobs
LMO	Loaded modules
LIB	CA Librarian members
LV	CA Librarian versions
LOG	System log
P	PDS members
PV	PDS versions
SYS	Job SYSOUT files
VOL	Volumes
PAN	CA Panvalet members
UNI	System units

Additional objects under z/VSE	Explanation
DA	Active jobs
DJ	Jobs
DV	Volumes
FIL	Files
LIB	CA Librarian members
LV	CA Librarian versions
LOG	System log
MEM	z/VSE members
SUB	Sublibraries
PAN	CA Panvalet members
UNI	System units
VV	Member versions

Additional objects under BS2000	Explanation
BF	BS2000 files
BJ	BS2000 jobs
BV	BS2000 job variables
LMS	LMS elements
LMV	LMS element versions

Object-Parameters

There are two types of object parameters.

- **Positional parameters:**
  These correspond to the name parameter fields on the object Entry Panel and identify the required object, for example `library(member)`;
- **Keyword parameters:**
  Specify a further characteristic of the object that corresponds to a parameter field on the object Entry Panel other than the name parameters. Keyword parameters take the format `KEYWORD=value`, for example `TYPE=t`.

  Keyword parameters can be abbreviated. The abbreviation must be long enough to identify the keyword, for example, `LIST DS * V=MYDISK`. In this case, `V` is a valid abbreviation for the keyword `VOLSER`.

  A list of possible keywords and their valid synonyms can be found in the section *Keyword Parameters* at the end of this documentation.

Function Parameters - for CHANGE, COPY, DOWNLOAD, EXPORT, HOLD, PRINT, RELEASE, RENAME, SUBMIT, UPLOAD

Command	Function Parameters	Meaning (Member: `KEY-FUNC`)
`CHANGE`	`NEWCLASS`	Changes class of job (valid in z/OS and z/VSE environments).
	`DISP`	Changes disposition of job in R queue (z/VSE only).
	`PRIORITY`	Changes job priority (z/VSE only).
	`COPIES`	Changes the number of output copies (z/VSE only).
	`NEWDEST`	Changes output destination (valid in z/OS and z/VSE environments).
	`SHARE`	Makes file/job variable accessible to other users (for BS2000 files/job variables)
	`ACCESS`	Changes protection against modification (for BS2000 files and job variables).
	`RETPD`	Changes retention period (for BS2000 files and job variables).
	`READPSWD`	Changes read protection (for BS2000 files and job variables).
	`WRITEPSWD`	Changes write protection (for BS2000 files and job variables).
	`EXECPSWD`	Changes execution protection (for BS2000 files only).
	`BACKUP-TYPE`	Changes backup class (for BS2000 files only).
`COPY`	*`target-parms`*	Object parameters of the source to which the object is to be copied. Replaces the target object with the same name. If this parameter is omitted, no replacement occurs.
	`REPLACE`	
`DOWNLOAD`	`SOURCE`	Downloads program source (Natural views and members only).
	`BINARY`	Downloads a Natural view or member in binary format.
	`OBJECT`	Downloads a cataloged Natural object.

Command	Function Parameters	Meaning (Member: KEY-FUNC)
	DESTINATION	Destination of downloaded Natural object (for example, PC file name).
EXPORT	PC	Target environment is a PC.
	CNT	Target environment is Con-nect.
	CABINET	Name of Con-nect cabinet, where object is to be stored as a document.
	PASSWORD	If required, Con-nect password to access the cabinet.
	DESTINATION	Document name under which the object is to be stored. Not required when using the SEND function.
	SEND	Name of recipient.
	TARGET	PC or Con-nect.
HOLD	LEVEL	Holds a version and sets the version number to *n*.
PRINT	*name*	Name of the printer. This printer overrides the printer specified in your user profile.
	ASIS	Valid from an Editor session only: prints the whole Editor session, including header, PF key line, etc.
	CONTROL	Honors any ASA or machine code control characters. No additional headers will be printed.
	NOCONTROL	Deactivates automatic carriage control when printing Natural objects or job SYSOUT files.
	DRIVER	Specifies the name of a printer control characters table as defined in the Natural NTCC macro, or under Com-plete the name of a logical output driver routine, which can perform additional output formatting during printing.
	FORM	Specifies a printout form.
	NAME	Specifies a list name for the printout.
	DISP	Disposition of the printout.
	COPIES	Specifies the number of additional printouts.
	WORKPOOL	Writes the output to the user workpool.
	PS	Specifies the number of lines per page for the printout.
	SUPPRESS	Suppresses header information and generation of form feeds.
	PRINTER	Name of the printer. This printer overrides the printer specified in your user profile.
	NOM	Uses the extended interface between Natural ISPF and Entire Output Management (NOM).
RELEASE	NEWCLASS	Releases held output of a job and assigns new class (z/OS only).
RENAME	NEWNAME	New name of the object to be renamed.
	VERSION	New version name to be assigned. This keyword is accepted only when renaming an LMS element version.
SUBMIT	TARGET=*id*	Where *id* is the target node on which the job is to be submitted, if different from the current node.

Command	Function Parameters	Meaning (Member: KEY-FUNC)
	TYPE=IDCAMS	The object is not treated as a job, but the command sequence is passed to the IDCAMS utility.
	TYPE=TSO	The object is passed to the TSO Batch interface and should contain valid TSO commands.
UPLOAD	SOURCE	Uploads program source (Natural views and members only).
	BINARY	Uploads a Natural view or member in binary format.
	OBJECT	Uploads a catalogued Natural.
	REPLACE	Overwrites an existing object.
	FROM	Location of Natural object to be uploaded (for example, PC file name).

Which Commands for which Object Type?

Full parameter syntax is described for each object type in the relevant section of this documentation. No special distinction of required and optional parameters is made here, as Natural ISPF supports function command input with windows that prompt you for any required parameter you omit.

This subsection describes all function commands in alphabetical order and gives some examples. Each function command appears together with its valid abbreviation, which can be used as a line command from a list of object names.

Not all function commands apply to each object type. To find out which commands are available for a specific object type, simply type the object type code in the Natural ISPF command line and press ENTER. A window opens with a list of the valid functions.

Example:

To list available functions for Natural objects, type the object code N in the command line and press ENTER. The list of valid functions is displayed:

```
-------------------------- NATURAL ISPF MAIN MENU -----------------------------
OPTION  === +-------------------------------------------------+
            ! ENTER FUNCTION:                                  ! id   BRY
            !   1 L  LIST              17 DS DESCRIPTION       !      11:04:01
   0    PRO !   2 B  BROWSE            18 FR FORMAT            ! inal DAEFTC55
   1    NAT !   3 E  EDIT              19 DW DOWNLOAD          ! ary  BRY
   2    VIE !   4 D  DELETE            20 UP UPLOAD            !      148
   3    ERR !   5 R  RENAME            21 CR COMPARE           !
   4    PRE !   6 SB SUBMIT            Select ==> __           !
   5    WOR !   7 PL PLAY                                      !
            !   8 PR PRINT                                     !
   6    JOB !   9 CP COPY                                      !
   7    PDS !  10 CT CATALOG                                   !
   8    DAT !  11 U  UNCATALOG                                 !
   9    JOB !  12 I  INFORMATION                               !
   10   MEM !  13 HL HOLD                                      !
   11   VSE !  14 RU RUN                                       !
   12   BS2 !  15 XE EXECUTE                                   !
   13   LMS !  16 EX EXPORT                                    !
   14   BS2 +-------------------------------------------------+
   15   JOB-VAR'S  - Maintain job variables (BS2000)

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

ALLOCATE / AL

Allocates a z/OS dataset, a z/VSE file or a BS2000 file. The `ALLOCATE` command is usually issued for an existing dataset or file. The allocation information for the existing item is displayed, and you can overtype the name and modify the specifications for the new item.

Examples:

Command	Function
`AL D MYFILE`	Assuming you have a cataloged dataset named `MYFILE`, this command displays the Allocate Dataset screen with information for the `MYFILE`. Modify the display for the dataset to be allocated.
`AL D NEWFILE VOL=COM811`	If you wish to allocate a new, uncataloged dataset without a model, this command displays the blank Allocate Dataset screen for file `NEWFILE` on Volume `COM811`.

Under z/OS, this command is invoked implicitly if the target dataset of a copy operation does not exist.

BROWSE / BR

Displays the specified object in Editor format. This means you can use all Editor scrolling commands, including FIND and LOCATE, as well as appropriate local commands described in the subsection *Local Commands - Description*.

Examples:

Command	Function
BR N NATLIB	Browses the Natural object MYPROG in the (MYPROG) library NATLIB.
BR P MYLIB(EX*)	Displays a list with members starting with EX in PDS library MYLIB. You can enter another prefix in the window to modify the list, or mark a member with any character to browse it.
BR CON	Displays the console; you can issue any operator command if you precede it with the Natural ISPF command OPERATOR.
BR LOG	Displays the system log.

CATALOG / CT

Catalogs the specified Natural program or dataset.

Example:

The command:

```
CAT N NATLIB(MYPROG)
```

catalogs the Natural program MYPROG in the library NATLIB.

CC

Displays the condition codes of the job specified in the command parameters. Condition codes appear in Editor format in browse mode.

Example:

The command:

```
CC COM444
```

displays the condition codes for job COM444. If there are several copies of the same job, use the job number parameter, otherwise you are notified with a message and the code for the copy last submitted is selected.

If you issue the CC command without parameters, the job last submitted from Natural ISPF is selected.

CHANGE / CH

Changes one or more attributes of the specified object; valid for z/OS jobs and SYSOUT file(s), z/VSE jobs, BS2000 files and job variables. The new attribute values can be specified by means of function parameters (see below). If none of the available function parameters are present, the new attribute values are prompted.

Available function parameters and restrictions:

Parameter	Restriction
NEWCLASS=*c*	(valid in z/OS and z/VSE environments)
DEST=*destin* or NEWDEST=*destin*	(valid in z/OS and z/VSE environments).
DISP=*disp*	(z/VSE only)
PRIORITY=*p*	(z/VSE only)
COPIES=*nnn*	(z/VSE only)
USER=*user*	(z/VSE only, synonym for DEST)
SHARE=YES/NO	(for BS2000 files and job variables)
ACCESS=READ/WRITE	(for BS2000 files and job variables)
RETPD=*nnnn*	(for BS2000 files and job variables)
READPSWD=*password*	(for BS2000 files and job variables)
WRITEPSWD=*password*	(for BS2000 files and job variables)
EXECPSWD=*password*	(for BS2000 files only)
BACKUP=*t*	(for BS2000 files only)

Combinations of the above parameters can be specified in one command.

Note: Job classes, output classes and output destinations are installation-dependent.

Example Job Class (z/VSE):

The command:

```
CH J MYJOB CLASS=C
```

changes the class of job `MYJOB` in an z/OS environment from `C` to a new class which is prompted. Additionally a new output destination can also be specified in the prompt window. If there is more than one copy of the job, use the job number parameter. If JES2.4.1 or a lower version of JES2 is installed at your site, the specified job queue entry must not be in the HOLD queue.

Example Job Class (z/OS):

The command:

```
CH J MYJOB CLASS=C
```

changes the class of job `MYJOB` in an z/OS environment from `C` to a new class which is prompted. Additionally a new output destination can also be specified in the prompt window. If there is more than one copy of the job, use the job number parameter. If JES2.4.1 or a lower version of JES2 is installed at your site, the specified job queue entry must not be in the HOLD queue.

Example Job Variable (BS2000):

The command:

```
CH JV PARMJV,SHARE=Y WRITEPSWD=BOHR
```

makes job variable `PARMJV` shared, that is, accessible to other users; modification of the job variable is password protected.

Example SYSOUT File:

The command:

```
CH SYS 3982 SI=SO FILE=2
```

prompts you for a new job class to which file `SO-2` of job number `3982` is to be redirected. The specified job file must be in the HOLD queue.

COMPARE / CR

Compares Natural sources stored in the Natural system file. For further information, see the example in the corresponding subsection of the section *Common Objects*.

COMPRESS / CM

Compresses the specified dataset.

Example:

The command:

```
COM D L99022.EDITOR.LOAD
```

compresses the dataset `L99022.EDITOR.LOAD` after confirmation. You are notified of completion by a message.

COPY / CP

Copies the specified object as another object of the same or different object type. You can enter target parameters as part of the command syntax.

Under z/OS, if the target dataset of a `COPY D dataset-name` command does not exist, you are prompted for a file allocation.

Examples:

Command	Function
`CP N NATLIB(MYPROG), P MY.ONLY.SOURCE(PROG01) REP`	Copies Natural program `MYPROG` in library `NATLIB` to PDS member `PROG01` in dataset `MY.ONLY.SOURCE`. If member `PROG01` already exists in the target library, it is replaced.
`CP P MYLIB(ISP*),P YOURLIB`	Copies all members in PDS library `MYLIB` that start with `ISP` to the PDS library `YOURLIB`.
`CP LMS,NAT`	Prompts you first for the LMS element to be copied, then for the Natural object as destination.

See also the subsection *Copying Objects* in the section *Useful Features*, as well as the `COPY` local command.

DEFINITION / DF

Displays the specified Natural DDM.

Examples:

Command	Function
`DF V PERSONNEL`	Displays the field definitions in the `PERSONNEL` view.
`DF V P*`	Displays a list of views starting with `P`. You can mark a view with any character to display the field definition.

DELETE / D

Deletes the specified object from the system file or system environment after confirmation.

Example:

The command:

```
CONFIRM OFF;D N NATLIB(NATMEM)
```

deletes, without the confirmation prompt, the Natural member `NATMEM` from the library `NATLIB`.

DESCRIPTION / DS

Opens an edit session with the Predict long description of the selected Natural object, Natural view or any other Predict object type in Editor format. You can modify the description as required.

Example:

The command:

```
DS N NATLIB(MYPROG)
```

starts an edit session with the Predict description of Natural program `MYPROG` in library `NATLIB`.

Note: A Predict entry must exist for the selected object.

DIFFERENCE / DI

Note: Not available for LMS versions.

The `DIFFERENCE` function is available for any previous version of a versioned object. It displays the current version together with any changes made during the period between the selected version and current version. Changes are indicated by highlighting and a corresponding remark in the prefix area.

For example, the command:

```
DI NV MBE(MYPROG)
```

opens a window prompting you for the date and time of a previous version for Natural member `MYPROG` in library `MBE`. It is more common, however, to issue the `DIFFERENCE` function as a line command from a list of previous versions for the member.

See the subsection *Versioning* in the section *Useful Features* for an example.

DOWNLOAD

In addition to the `EXPORT` command, which usually handles text only, the `DOWNLOAD` command also downloads binary data. Currently it is available for the following Natural ISPF objects:

Object	Explanation
`N` Natural	Objects and sources as well as data areas and maps are processed.
`V` Views	Text and binary download possible.
`P` PDS members	Load modules are handled.

The command format is:

```
DOWNLOAD object-type object-id, SOURCE BINARY OBJECT DESTINATION=file.ext
```

The keywords `SOURCE BINARY OBJECT` are evaluated for views and Natural members only. They have no meaning when downloading PDS members. Any combination of these keywords can be entered in one command, allowing Natural source and object to be downloaded in a single command. If multiple download types are entered, the `DESTINATION` parameter cannot be entered.

The `DESTINATION` parameter can be used to enter the PC file name.

Examples:

Command	Function
`DOWNLOAD PDS ML(PROG1), DEST=PROG1.NCD`	Downloads the load module `PROG1` to `PROG1.NCD` in your working directory, if `ML` is an abbreviation of an z/OS load library.
`DOWNLOAD NAT MYPROG, SOURCE OBJEC`	Downloads source and object of Natural program `MYPROG`. Entire Connection prompts for PC file names.
`DOWNLOAD V EMPLOYEES, BINARY`	Downloads view (DDM) `EMPLOYEES` in binary format. Entire Connection prompts for PC file name.

All PC files created with a Natural ISPF `DOWNLOAD` command can be processed by the `UPLOAD` command.

Note: You can transfer data to a PC only if you are using a PC to emulate a mainframe terminal with Software AG's Entire Connection.

EDIT / E

Starts an edit session with the specified object. If the object does not exist, it is created.

Examples:

Command	Function
`E N NATLIB(MYPROG)`	Starts an edit session with Natural member `MYPROG` in library `NATLIB`.
`E P MYLIB(EX*)`	Opens a window with a list of all members starting with `EX` in the PDS library `MYLIB`. You can type another prefix in the window to modify the list, or mark a member with any character to start the edit session.
`E MYPROG`	Assuming `NAT` is set as default object type and `NATLIB` as default library name in your user profile, this command entered from the Natural ISPF Main Menu is sufficient to start an edit session with member `MYPROG` in the default library.

ENTRY / EN

The `ENTRY` command displays the Entry Panel for the specified object type, for example, the command:

```
EN NAT
```

displays the Natural Objects Entry Panel. Usually, the system administrator uses this command in menu definition (see the section *Menu Maintenance* in the *Natural ISPF Administration Guide*). When working with Natural ISPF, you will normally display an Entry Panel by selecting an option from the Main Menu.

If you use this command as a line command (`EN`), you can select a specific object from a list. This invokes the Entry Panel for the object type, with the parameter fields filled with the selected object's parameter values. This makes it easy to start sessions with objects with similar names.

See the subsection ***ENTRY as Line Command*** in the section *Useful Features* for an example.

EXECUTE / XE

Executes the specified Natural object. The object must be cataloged (`CATALOG` or `STOW` command). If the object is a macro object, the generated output is written to the user workpool (see also the section *Macro Facility* in the *Natural ISPF Programmer's Guide*).

Example:

The command:

```
XE N NATLIB(MYPROG)
```

executes the Natural object `MYPROG` in library `NATLIB`.

EXPORT / EX

Exports the specified object to the specified target environment (PC or Software AG's automated office system Con-nect). For more details, including command syntax and available keywords, see the subsection ***Natural Interface to External Environments*** in the section *Useful Features*.

Example:

The command:

```
EX P MY.ONLY.SOURCE(MYMEM),PC
```

opens a window in which you can specify the PC file name under which the PDS member `MYMEM` in library `MY.ONLY.SOURCE` is to be downloaded.

When you use the `EXPORT` command from an Editor session, the shortest possible abbreviation is `EXP`.

Note: You can transfer data to a PC only if you are using a PC to emulate a mainframe terminal with Software AG's Entire Connection.

EXTENTS / ET

The `EXTENT` command displays the extents for a specific dataset, giving the disk address and cylinder size of each one.

Example:

The command:

```
ET D MBE.COMN.SOURCE
```

displays extent information for dataset `MBE.COMN.SOURCE`. For an example of the `EXTENTS` command, see the section ***z/OS Objects*** in this documentation.

EXTERNS / XT

Displays all external references to a load module or CSECT (z/OS only).

Example:

The command:

```
XT P MBE.COMN.LOAD(NATPARM)
```

displays a list of external references to the load module `NATPARM` in the load library `MBE.COMN.LOAD`. For an example, see the subsection ***Load Modules and CSECTs*** in the section *z/OS Objects*.

FOLLOW / FL

Instructs Natural ISPF to report progress of the specified job. The status message remains in the message line until the job is in the output queue (z/OS and z/VSE) or until job execution has been completed (BS2000). You can discontinue the status reports by using the command `FOLLOW OFF`.

Example:

The command:

```
FL J ISPINT
```

reports the status of job `ISPINST` in an z/OS environment every time you invoke another system screen. If there is more than one copy of the same job, you are notified with a message, and the copy submitted last is selected.

If you use the `FOLLOW` command without parameters, the job submitted last from Natural ISPF is the default.

FORMAT

This function command applies only to Natural objects of the type map. The map layout is displayed in a Natural ISPF Editor session. Modifiable fields (`AD=A` and `AD=M`) are replaced by a special filler character which is the underscore (_). Variable output fields (`AD=O`) are replaced by a period (.) .

Example:

The command:

```
FR ISPN---1
```

displays the map layout in an Editor session.

HOLD / HL

Issued for a job, this command puts the SYSOUT of the specified job on hold.

Example:

The command:

```
HL J ISPINST
```

puts the SYSOUT of job `ISPINT` on hold.

The `HOLD` command is also available for the current version of an object (not applicable to LMS elements). Placing a member in `HOLD` status means that it is not counted as an existing version and will not be automatically deleted as further versions of the member are created. For more information and an example, see the subsection ***Versioning*** in the section *Useful Features*.

INFORMATION / I

Displays information about the specified object.

Example:

The command:

```
I N NATLIB(NATPROG)
```

displays information about Natural object `NATPROG` in library `NATLIB`.

LIST / L

Displays a list of specified objects. You can select an object from a list for further handling using appropriate line commands (abbreviations of function commands).

Note: The command `LIST DS *` gives a list of short names for libraries at system and at user level (see the subsection ***Library Definition*** in the section *Profile Maintenance*.

Examples:

Command	Function
`L N NSPFWORK(ISPF*)`	Lists all Natural objects with prefix `ISPF` in the library `NSPFWORK`.
`L N SYSTEM(A*) TYPE=PM`	Lists all Natural programs and maps whose names start with `A` in library `SYSTEM`.
`L N NATLIB(*) MACRO=MODEL`	Lists all Natural objects in library `NATLIB` that used macro object `MODEL` as edit macro.
`L P MB(*INPL)`	Lists all PDS members in library with short name `MB` with suffix `INPL`.
`L A OP* TYPE=J`	Lists all active standard-type jobs with prefix `OP` (z/OS).
`L N C <`	Lists all Natural members which start with a value less than `C` in the current library.

OUTPUT / OT

- Displays the output of CA Panvalet or CA Librarian protocol after a member is saved and refers to the member saved most recently.
- Displays a list of files created by a specific BS2000 job. If the job is not explicitly specified, the job submitted most recently from your Natural ISPF session is selected.

Examples:

Command	Function
`OT PAN`	Displays the protocol of the saved member.
`OT BJ 1422`	Displays a list of files created by the BS2000 job identified by TSN `1422`.

PLAY / PL

Executes the specified member as a Natural ISPF command script.

Example:

The command:

```
PLAY N MYLIB(SCRIPT)
```

executes the Natural member `SCRIPT` in the library `MYLIB` as Natural ISPF commands. For more information and examples of command scripts, see the subsection ***Executing Command Scripts*** in the section *Useful Features*.

The commands described below can be entered interactively but they are meant to be used in Natural ISPF command scripts.

Command	Function
`CONTINUE`	Defines labels where processing is to continue after errors during execution of a command script.
`MACPARM P112273`	Used to enter input parameters in a script for a macro.
`MESSAGE`	Displays a message in the top right corner and enters `PAUSE` mode.
`REMARK`	Used to document a command script.

PRINT / PR

Prints the specified object at the hard copy device selected as follows:

- Printer specified in the function parameters;
- Printer specified by name in your user profile;
- Printer specified by name in your user group profile (for example, if your user ID is `ABC`, profile `A*`);
- Printer specified in prompt window that appears if an asterisk (*) is specified as printer in your user profile;
- If no printer specification is made in your user profile, the printer defined in your TP environment;

- If no printer is defined in your TP environment, the printer specified by the Natural profile parameter PRINTER2.

Examples:

Command	Function
PR N NATLIB (MYPROG),PRINTER1	Prints Natural member MYPROG from library NATLIB at printer PRINTER1. Automatic carriage control is performed.
PR P MY.ONLY. SOURCE(MYMEM),CC	Prints PDS member MYMEM in library MY.ONLY.SOURCE at the selected printer, honoring any ASA or machine code characters in the member. No other headings are printed.
PR N NSPF120(ISPEX1),WORKPOOL	Prints Natural member ISPEX1 in library NSPF120 to the user workpool.
PR J COM444,NO	Prints the SYSOUT of z/OS job COM444. Automatic carriage control is deactivated.
PR N MYPROG, PRINTER= PRINTER7 COPIES=2	Prints two copies of Natural member MYPROG from the current library at PRINTER7.

Note: The functional parameter NO is available for Natural and job SYSOUT only: without this parameter, automatic carriage control is performed

Printing from an Editor Session

1. Issue the PRINT command from an Editor session with the object in EDIT or BROWSE mode to print the member;

   Note: When issuing the PRINT command from an Editor session, the current boundary settings (BNDS Editor command) are respected: only the data within the set boundaries are printed.

2. Use the Editor line command P to print the selected line, or mark the first and the last lines of a block of text with the Editor PP line command to print the block from the member;

When issuing a PRINT command from an Editor session, you can use the special parameter ASIS as follows:

```
PRINT,ASIS
```

- If a printer is defined in your Natural ISPF profile, the whole Editor session is printed, including header, prefix, etc.
- If an asterisk (*) is defined as printer in your Natural ISPF profile, you are prompted with the following window:

```
+-----------Print parameters------------+
!                                       !
! Lines per page    : 60               !
! Take linesize from                   !
!   edit session    : X                !
!   or screensize   :                  !
! Print prefix      : X                !
! Enter printer     : _________        !
! Listname          : ________         !
! Form              : ____             !
! Disp              : L___   (D/H/K/L) !
! Copies            :          (0-255) !
! Log.-Driver       : ________         !
! Print via NOM     :          (Y/N)   !
+---------------------------------------+
```

Meaning of the input fields:

Input Field	Meaning	
`Lines per page`	Number of lines to be printed on a page before a page break.	
`Take linesize from edit session`	Mark this field if the line length of the printout is to correspond to the length of the edit session (this might be more than 80).	
`or screensize`	Mark this field if the line length of the printout is to correspond to the line length of your screen.	
`Print prefix`	Mark this field if you want the prefix area (line numbers, labels, etc.) printed.	
`Enter printer`	Enter printer name (this can also be Workpool).	
`Listname`	Specify a name of the printout.	
`Form`	Specify a printout form.	
`Disp`	Specify the disposition of the printout:	
	`D` or `DEL`	Delete after printing (default).
	`H` or `HOLD`	Hold printout, do not print.
	`K` or `KEEP`	Keep after processing.
	`L` or `LEAV`	Leave in spool queue after printing.
`Copies`	Specify number of copies (maximum is 255).	
`Log.-Driver`	Specify the name of a logical output driver routine to perform additional output formatting during printing. **Note:** If you want to use the logical output drivers under Software AG's TP monitor Com-plete ask your administrator to activate the special USPOOL interface with `APPLYMOD 22`.	
`Print via NOM`	Enter `Y` to use the extended interface between Natural ISPF and Entire Output Management.	

PURGE / PG

Purges the SYSOUT of a job from the job entry queue.

Example:

The command:

```
PG DJ ISPINST
```

purges the SYSOUT of job `ISPINST` from a z/VSE job entry queue.

RELEASE / RL (Member: RELEASE)

Releases a job, its output or a specific SYSOUT file (object-type `S`, z/OS only) from `HOLD` status (when job has been held explicitly) or from Hold Queue (when job output is associated with a held output class).

Available function parameter (optional):

Parameter	Meaning
`NEWCLASS=c`	Assigns new output class. Valid for z/OS objects only.

Examples:

Examples for use with function command syntax:

Command	Function
`RL J ISPINT RL DJ ISPINT`	Releases job `ISPINT` from Hold status or from Hold queue, whichever applicable. If the job name is not unique, specify job number.
`RL J 3687 QUEUE=H CLASS=O`	Releases held output of job 3687 from class `O` and prompts for optional specification of a new class.
`RL J 3687 CL=O,NEWCL=Y`	Releases held output of job 3687 from class `O` and assigns new output class `Y` (without prompting).
`RL S ISPINT`	Prompts for the name and number of the required SYSOUT file to be released, and for optional specification of a new class.

Note: Only `SM` and `SO` files can be subject to `RELEASE` operations.

Similar considerations apply for use as line command (`RL`).

RENAME / RN

Renames the specified object to the new name specified in the function parameters. If you enter the `RENAME` command without the function parameter, a window is opened in which you can specify the new name.

Examples:

Command	Function
`RN P MYLIB(MYMEM),NEWMEM`	Renames member `MYMEM` to `NEWMEM` in PDS library `MYLIB`.
`RN LMV MYLIB(MYELEM/§), VERSION=V001`	Renames version § of element `MYELEM` in LMS library `MYLIB` to `V001`.

RUN / RU

Compiles and executes the specified Natural program. If the program uses the macro facility or contains the `INCLUDE-MACRO` statement, the macros are executed and the output is written to the user workpool.

STATUS / ST

Displays a status message of the specified job.

Example:

The command:

```
ST J ISPINT
```

reports the status of job `ISPINST` in an z/OS environment. If there is more than one copy of the same job, you are notified with a message, and the copy submitted last is selected.

If you use the `STATUS` command without parameters, the job submitted last from Natural ISPF is the default.

SUBMIT / SB

Submits the specified object to the operating system. You can follow the submitted job with the `FOLLOW` and `STATUS` commands. If the job control contains macros, expansion is performed before submission, provided that the macro expansion has been activated in your current User Defaults profile (see the subsection ***User Defaults*** in the section *Profile Maintenance*). In this case, you can access the submitted object in the `WORKPOOL` facility.

The `SUBMIT` command can be issued with the following function parameters:

Parameter	Meaning
`TARGET`	Allows you to execute a JCL member on a node different from its physical location.
`TYPE`	Possible options: `IDCAMS` - The submitted object is treated as a command sequence and passed to the `IDCAMS` utility. `TSO` - The object is passed to the TSO Batch interface and should contain valid TSO commands.

Examples:

Command	Function
`SB N NATLIB(MYPROG)`	Submits Natural program `MYPROG`.
`SB P MYLIB(MYPROG),TARGET=69`	Submits PDS member `MYPROG` located on the default node (usually `148`) on Node `69`.

UNCATALOG / U

Uncatalogs the specified Natural object or dataset.

Example:

The command:

```
U N NATLIB(MYPROG)
```

uncatalogs Natural program `MYPROG`.

UPLOAD

In addition to the IMPORT command, which is a local command in the Editor and handles text only, the UPLOAD command also uploads binary data. Currently it is available for the following Natural ISPF objects:

Object	Explanation
N Natural	Objects and sources as well as data areas and maps are processed.
V Views	Text and binary upload possible.
P PDS members	Load modules are handled.

The command format is:

```
UPLOAD object-type object-id, SOURCE BINARY OBJECT FROM=file.ext REPLACE=YES
```

The keywords SOURCE BINARY OBJECT are evaluated for Views and Natural members only. They have no meaning when uploading PDS members. Any combination of these keywords can be entered in one command, allowing Natural source and object to be uploaded in a single command. If multiple upload types are entered, the FROM parameter cannot be entered.

The FROM parameter can be used to enter the PC file name. The REPLACE parameter can be used to overwrite an existing object.

Upload of a load module with REPLACE=YES should not be interrupted, because the old module is deleted before starting the upload from PC.

Examples:

Command	Function
UPLOAD PDS ML(PROG1), FROM=PROG1.NCD	Uploads the load module PROG1 is from PROG1.NCD from your working directory, if ML is an abbreviation of an z/OS load library. Upload of load modules works only if the PC files have been created with Natural ISPF DOWNLOAD command. In addition, the BLOCKSIZE of the source and target load library must be identical.
UPLOAD NAT MYPROG, SOURCE OBJECT	Uploads source and object of the Natural program MYPROG. Entire Connection prompts for PC file names.
UPLOAD V EMPLOYEES, BINARY REP=YES	Uploads view (DDM) EMPLOYEES in binary format. Entire Connection prompts for PC file name. If view EMPLOYEES already exists, it is replaced.

Note: You can transfer data from a PC only if you are using a PC to emulate a mainframe terminal with Software AG's Entire Connection.

ZAPS / ZP

Displays a list of Zaps applied to the specified load module or CSECT (z/OS only).

Examples:

Command	Function
`ZP P JWO.COMN.LOAD(NATPARM)`	Displays a list of Zaps applied to the load module `NATPARM` in the specified load library. The list shows CSECT name, date and IDR-DATA.
`ZP CST`	Opens a window in which you can specify the CSECT for which to list Zaps.

Local Commands - Description

Local commands are object-specific and can be issued only when displaying an object in Editor format in `LIST`, `EDIT` or `BROWSE` mode. If you enter an asterisk (*) in the command line, a window opens that lists all valid local commands and you can select a command for execution. Some local commands are available only in `LIST` mode, others only in `EDIT` or `BROWSE` mode.

This subsection describes all local commands in alphabetical order according to object type. Valid abbreviations of local commands are indicated in the command title.

All Objects that can be Edited

COPY

Copies a specified object into the current edit session. The place at which the copied object is to be inserted must be marked by appropriate Editor line commands (`A`, `B`, `O` or `OO`). The following object types can be copied:

Object type	Meaning
`BF`	BS2000 file
`D`	Dataset (sequential)
`DJ`	Job (z/VSE)
`FIL`	z/VSE file
`J`	Job (z/OS)
`LIB`	CA Librarian member
`LMS`	LMS library element
`LMV`	LMS element version
`LV`	CA Librarian member version

Object type	Meaning
MAC	Macro object
MEM	z/VSE member
N	Natural object
O	Output file in workpool
P	PDS member
PAN	CA Panvalet member
S	Job SYSOUT (z/OS)
USR	Natural ISPF user profile
V	Database view

You can issue the COPY local command with full name parameters, for example:

```
COPY MAC library (name)
```

If you issue the COPY command without parameters, you are prompted for object type and name. For special considerations that apply when copying Natural objects and views, see the subsection ***Natural Objects*** and the subsection ***Natural Views*** in the section *Common Objects*.

IMPORT / IM

Copies the specified PC file or Con-nect document into the edit area. If the edit area already contains data, you must mark the place at which you want the file or document copied with Editor line commands A (after this line) or B (before this line).

The command format is:

```
IMPORT [CNT / CONNECT]
       [PC]
```

Prompt windows open to prompt you for identifiers of the file or document you wish to copy.

For more information, including examples, see the subsection ***Natural Interface to External Environments*** in the section *Useful Features*.

To import Natural objects in binary form, use the UPLOAD command as described in the section *Function Commands*.

SET

Opens a window with your current edit profile. You can modify any parameter by overtyping the value.

The command format is:

```
SET
```

Lists of Objects

ALL

Executes a command or command sequence for all objects shown in the list. For details, see the subsection ***ALL Command for Lists*** in the section *Useful Features*.

The command format is:

```
ALL command-string
```

LAYOUT

Opens a window in which you can define a new layout for the list. For details, see the subsection ***LAYOUT Command for Lists*** in the section *Useful Features*.

The command format is:

```
LAYOUT
```

RELIST

Rebuilds the list to reflect changes from line commands such as `DELETE`, `RENAME` or `EDIT`. For details, see the subsection ***RELIST Command for Lists*** in the section *Useful Features*.

The command format is:

```
RELIST
```

SORT

When displaying lists of objects, you can use the Editor `SORT` command with special parameters:

```
SORT [column-header] [a] [d]
```

Where `a` is ascending order (default), `d` is descending order and *`column-header`* is the string of the column header according to which the items in the list are to be sorted. For details, see the subsection ***SORT Command for Lists*** in the section *Useful Features*.

Views - object type V

GENERATE / GEN

Issued from a view definition displayed using the `DEFINITION` function command, generates data definition statements for a Natural source.

Natural Objects object types N and MAC - in LIST mode

CATALL

Catalogs multiple Natural objects displayed in a list. The command format is:

```
CATALL [name] [type]
```

where *`name`* can be an optional name pattern, and *`type`* an optional Natural object type.

Examples:

Command	Function
`CATALL ISP*`	Catalogs all Natural objects in the list whose names start with `ISP`.
`CATALL ISP* P`	Catalogs all Natural programs in the list whose names start with `ISP`.

Note: If you issue the `CATALL` command for objects that do not use the macro facility, it is recommended that you issue a `MACRO OFF` command first, otherwise resources are wasted as the objects are searched for the macro character. If any object uses the macro facility, you must have `MACRO ON`.

Natural Objects object types N and MAC - in EDIT mode

CATALOG / CAT

Stores the edited Natural member in object form only (the source remains unchanged).

CHECK / C

Checks syntax of current Natural object. If the object is a macro-type program, only a check of processing statements and variables to be substituted is performed. The `CHECK` local command does not check that the lines generated by the macro are valid Natural source. To do this, execute the macro, and copy its generated output from the user workpool as a Natural program into a Natural library (see the subsection ***User Workpool*** in the section *Common Objects*).

OUTPUT

Starts an edit session with the output of the current program in the user workpool (only valid after a `RUN` command issued from the edit session).

REGENERATE / REG

Available for Natural programs written using the Edit macro option. Reexecutes the specified macro object and writes the result in the current edit session. Any defined user code remains in place. For details, see the section *Macro Facility* in the *Natural ISPF Programmer's Guide*.

STOW 'text'

Stores the Natural program in source and object form. The text parameter is relevant when stowing the program with versioning on: you can specify a reason for changing the program. The text must be enclosed in quotation marks.

Note: If the current Natural program includes inline macros and/or `INCLUDE-MACRO` statements, macro expansion is performed before compilation (`MACRO ON` must be set).

STRUCT

Performs structural indentation of Natural source statements and identifies any structural inconsistencies. This command does not apply to macro objects.

SM

Enables/disables structured mode. The command format is:

```
SM [ON]
   [OFF]
```

TYPE

Specifies Natural object type. The command format is:

```
TYPE t
```

where *t* can be any of the following:

C	Copy code
H	Help routine
N	Subprogram
O	Macro program
P	Program
S	Subroutine
T	Text

Natural Error Messages - in BROWSE or EDIT mode

NEXT

In BROWSE mode:	Displays the next existing error message.
In EDIT mode:	The current message is saved automatically if it has been modified. Then, an EDIT session is opened with the next existing error message.

Job SYSOUT - object type S

Job SYSOUT is displayed in BROWSE mode only.

FILE

Selects SYSOUT file for display. The command format is:

```
[FILE] [SI] [n]
       [SO]
       [SM]
       [CC]
       [JL]
```

where *n* stands for file number.

NEXT

Selects next SYSOUT file for display.

PREV

Selects previous SYSOUT file for display.

Library Members and Sequential Files - in EDIT mode

The following commands apply to object types `PDS`, `MEM`, `LMS`, `DS`, `FIL` and `BF`.

PASSWORD

This command is available when editing files that are password-protected for update.

The command format is:

```
PASSWORD password
```

Use this command to enter the valid password in order to update the protected file.

REGENERATE / REG

Available for source members written using the Edit macro option. Reexecutes the specified macro object and writes the result in protected lines in the current edit session. Any defined user code remains in place. For details, see the section *Macro Facility* in the *Natural ISPF Programmer's Guide*.

SUBMIT

If the member contains JCL, you can submit the job to the operating system from the edit session with the `SUBMIT` local command.

Additional Local Commands for z/VSE Members object type MEM - in EDIT mode

When selecting a member for `EDIT` from the VSE Members Entry Panel, or when displaying a member in edit mode, the following local commands are available:

MSHP

Specifies whether MSHP protection is to be bypassed `YES` or `NO`. The command format is:

```
MSHP [YES]
     [NO]
```

DATA

Specifies whether the member contains `SYSIPT` data `YES` or `NO`. The command format is:

```
DATA [YES]
     [NO]
```

Additional Local Commands for LMS Library Elements - in EDIT mode

Applies to object types `LMS` and `LMV`.

NEWNAME

Use this command to specify a new element name and/or a new version name, to be written when saving a member. The command format is:

```
NEWNAME [ename][nversion]
```

If you issue the `NEWNAME` command without parameters, a window opens in which you can specify the new name and/or version name. For more details, see the subsection ***LMS Library Elements*** in the section *BS2000 Objects*.

Online Help Facility and UINFO Texts

UP LEVEL or BACK

When scrolling help or information screens, you can use the `UP` command with the parameter `LEVEL` to move directly to the next higher level in the screen hierarchy.

The command format is:

```
UPLEVEL
BACK
```

Assuming the `UP` command is assigned to PF7, you can type `L` in the command line and press PF7.

14 Keyword Parameters

This section contains a list of all possible keywords and their synonyms that can be used as keyword parameters on Natural ISPF function commands (see the subsection ***Function Commands - Description*** in the section *Command Reference*.

Keyword	Synonym / Abbreviation
ACCESS-TYPE	-
BLOCK	-
CATALOG	CAT
CLASS	-
COMMENT	-
CREATION-DAT	-
CSECT	-
DATA	-
DBID	DB
DESTINATION	TO
DEVICE	OUT-DEVICE
DISP	-
ELEMENT-PSWD	ELPSWD
EXPAND	-
FILE	
FNR	-
FROM	-
FCBTYPE	-
GROUP	-
IN	-
INMEM	-

Keyword	Synonym / Abbreviation
JOBNAME	JNAME
LANGUAGE	LANG
LAST-ACCESS	-
LINES	-
LVL	-
MACRO	MODEL
MSHP	-
NAME	-
NODE	-
ORIGINATOR-T	OR-TSN ORIGIN-TSN R-TSN
PASSWORD	PSWD
PGMR	-
PROFILE	-
PRIORITY	PRIO
QUEUE	-
RECLEN	RECL
RECLENGTH	-
SCAN	SC
SET	-
SHARE	-
SIZE	-
STATE	-
STATUS	ST
STORAGE-MODE	MODE
SUPPORT	-
SYSNAME	SI
SYSNUM	FILE
SYSTEM	-
TIME	-
TO	-
TSN	-
TYPE	-
UNIT	-
USER	OWNER FROM
VERSION	VER
VOLSER	VOL