# Natural ISPF

## Administration Guide

Version 9.2.2

October 2023

This document applies to Natural ISPF Version 9.2.2 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 © 1989-2023 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-ADMIN-922-20231011**

## Table of Contents


Preface ..... v
Naming Conventions ..... vi
1 About this Documentation ..... 1
Document Conventions ..... 2
Online Information and Support ..... 2
Data Protection ..... 3
2 System Configuration ..... 5
System Configuration Overview ..... 6
Further Customization Using Open NSPF ..... 6
A Note about Library Names ..... 7
Editing the Configuration Member CONFIG ..... 7
Editing the CA Panvalet Definition Member ..... 32
Editing the CA Librarian Definition Member ..... 34
Defining Short IDs for Libraries ..... 37
Natural ISPF Parameters ..... 39
NCP Concept - Member: NCPUSAGE ..... 41
Defining Versioned Libraries ..... 43
Maintaining Versioning Data ..... 47
Entire System Server Node Table ..... 52
3 User Definitions ..... 55
Types of User Definition ..... 56
User Group Definitions ..... 57
Maintaining User Definitions ..... 59
Authorization Table ..... 60
Maintaining User Definitions with Function Commands ..... 65
4 Menu Maintenance ..... 69
Starting a Menu Edit Session ..... 70
Syntax of Menu Definition ..... 70
Customizing Menus ..... 74
Maintaining Menus with Function Commands ..... 86
5 Site-Specific Online Information ..... 89
Site-Specific Help ..... 90
Site-Specific Information - UINFO ..... 92
Text Syntax ..... 92
6 User Exits ..... 95
Object Exits ..... 96
Examples of Object Exits ..... 99
CA Panvalet Save Exit ISPT-SVU ..... 101
Logon Exit ISP-LONU ..... 101
Logoff Exit ISP-LOFU ..... 102
Print User Exit ISP-PRTU ..... 103
Import/Export Exits ..... 105
Color Settings Exit ISP-ECLU ..... 106

Resume Exit ISP-RESU ..... 106
Suspend Exit ISP-SUSU ..... 108
Session Exit ISP--S-U ..... 109
Rename Function Exit ISP-RN-U ..... 110
User Library Exit ISP-PRFU ..... 111
User Group Exit ISP-UGRU ..... 111
Node Exit ISP-NODU ..... 113
HSM - Hierarchical Storage Manager Exit ISP-HSMU ..... 114
Editor Profile Exit ISP-ED-U ..... 115
Incore Database Defaults Exit IDB-USRN ..... 116
Container File Access Exit IDBC---U ..... 117
7 Buffer Pool And Recovery Files ..... 119
Buffer Pool Maintenance ..... 120
Optimizing the Buffer Pool ..... 120
Buffer Pool Files ..... 120
Recovery Files ..... 121
Troubleshooting ..... 122
8 Natural ISPF Libraries ..... 127
9 Natural ISPF Special Characters ..... 129
10 Authorization Classes ..... 131
11 Subsystems Supported By Natural ISPF ..... 133

# Preface


- Naming Conventions ..... vi

This documentation is intended for the Natural ISPF system administrator and describes Natural ISPF administration procedures. Separate sections explain how to set up the Natural ISPF environment according to the requirements of your site.

This documentation covers the following topics:

| | |
|---|---|
| **System Configuration** | Describes system configuration procedures. These include subsystem customization, activating logon screen and user exits, as well as setting some parameters for the users' working environment. |
| **User Definitions** | Tells you how to create and maintain user definitions, including authorization profiles at the single user, user group and default level. |
| **Menu Maintenance** | Deals with menu customization. Syntax is described with which you can modify existing menus and create new ones. |
| **Site-Specific Online Information** | Describes how you can write online help texts and provide other online information. |
| **User Exits** | Tells how you can make use of the various user exits supplied by Software AG. |
| **Buffer Pool and Recovery Files** | Describes the Editor buffer pool and buffer pool recovery files, and gives instructions on how the buffer pool efficiency can be optimized. A special subsection on trouble shooting tells you what to do when certain messages are given out. |
| **Natural ISPF Libraries** | Lists the name of the libraries as they appear on the installation medium together with their descriptive names. In all sections except in the *Installation* documentation, libraries are referred to by their descriptive names. |
| **Natural ISPF Special Characters** | Lists all special characters used in Natural ISPF, together with their hexadecimal values for easy translation to other keyboards. |
| **Authorization Classes** | Lists the available authorization classes and the Natural ISPF objects they refer to. Authorization classes are used in menu line definition, in the site control table, and in user definition. |
| **Subsystems Supported by Natural ISPF** | Lists the available subsystems supported by Natural ISPF. Subsystems are activated by inclusion in the configuration member `CONFIG`, and are used in menu line definition. |

Further customization facilities are described in the *Natural ISPF Programmer's Guide*.

## Naming Conventions

The section ***Natural ISPF Libraries*** of this documentation contains a table that lists all Natural ISPF libraries as they appear after loading the installation medium, together with a descriptive name as to the library content. For example, a library named `SYSISPX` could be the Exit Library.

From the section ***System Configuration*** onwards, Natural ISPF libraries are referred to by their descriptive names (for example, Exit Library instead of `SYSISPX`).

# 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:<br><br>Variables for which you must supply values specific to your own situation or environment.<br>New terms the first time they occur in the text.<br>References to other documentation sources. |
| `Monospace font` | Identifies:<br><br>Text you must type in.<br>Messages displayed by the system.<br>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

### Product Documentation

You can find the product documentation on our documentation website at **https://documentation.softwareag.com**.

In addition, you can also access the cloud product documentation via **https://www.softwareag.cloud**. Navigate to the desired product and then, depending on your solution, go to "Developer Center", "User Center" or "Documentation".

### Product Training

You can find helpful product training material on our Learning Portal at **https://knowledge.softwareag.com**.

### Tech Community

You can collaborate with Software AG experts on our Tech Community website at **https://tech-community.softwareag.com**. From here you can, for example:

- Browse through our vast knowledge base.
- Ask questions and find answers in our discussion forums.
- Get the latest Software AG news and announcements.
- Explore our communities.
- Go to our public GitHub and Docker repositories at **https://github.com/softwareag** and **https://hub.docker.com/publishers/softwareag** and discover additional Software AG resources.

### Product Support

Support for Software AG products is provided to licensed customers via our Empower Portal at **https://empower.softwareag.com**. Many services on this portal require that you have an account. If you do not yet have one, you can request it at **https://empower.softwareag.com/register**. Once you have an account, you can, for example:

- Download products, updates and fixes.
- Search the Knowledge Center for technical information and tips.
- Subscribe to early warnings and critical alerts.
- Open and update support incidents.
- Add product feature requests.

## 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 System Configuration


- System Configuration Overview ..... 6
- Further Customization Using Open NSPF ..... 6
- A Note about Library Names ..... 7
- Editing the Configuration Member CONFIG ..... 7
- Editing the CA Panvalet Definition Member ..... 32
- Editing the CA Librarian Definition Member ..... 34
- Defining Short IDs for Libraries ..... 37
- Natural ISPF Parameters ..... 39
- NCP Concept - Member: NCPUSAGE ..... 41
- Defining Versioned Libraries ..... 43
- Maintaining Versioning Data ..... 47
- Entire System Server Node Table ..... 52

## System Configuration Overview

This section tells you how to configure some items in the Natural ISPF system to suit your installation's needs. The following points are described:

- Editing the `CONFIG` member to:
  - customize Natural ISPF to the subsystems installed at your site (see *Defining Installed Subsystems*);
  - activate user exits (see *Activating Natural ISPF User Exits*);
  - activate the logon screen (see *Activating Logon Screen*);
  - set special purpose switches (`APPLYMOD` parameters) (see *Special Purpose Switches*).
- Customizing CA Panvalet definitions (see *Editing the CA Panvalet Definition Member*);
- Customizing CA Librarian definitions (see *Editing the CA Librarian Definition Member*);
- Defining short names for libraries (see *Defining Short IDs for Libraries*);
- Setting Natural ISPF system parameters (see *Natural ISPF Parameters*);
- Generating a Natural command processor which can be used to improve performance (see *NCP Concept*);
- Defining libraries for which versioning is to be enforced (see *Defining Versioned Libraries* ) and maintaining versioning data (see *Maintaining Versioning Data*);
- Defining the Entire System Server node table (for multi-CPU environments or Mainframe Navigation) (see *Entire System Server Node Table*).

Other configuration tasks such as user definitions and menu customization are described in separate sections.

The following subsections describe these functions in the order you are advised to follow when configuring Natural ISPF. All settings can be modified later in any order by an authorized user.

## Further Customization Using Open NSPF

In addition to the administration functions mentioned above, you can further customize Natural ISPF to the requirements of your site using the Open NSPF facility. This allows you to:

- Define site-specific commands (for example, `MAIL` to check your automatic office system for a new item).
- Define site-specific objects and relate them to functions (for example, define `EMPLOYEE` as a new object and relate it to functions `LIST`, `INFORMATION`, `DELETE`).

The Open NSPF facility is described in detail in the *Natural ISPF Programmer's Guide*.

## A Note about Library Names

All Natural libraries accessed during administration functions are referred to in this section by their descriptive names. For a list of library names as they appear on the installation medium, see ***Natural ISPF Libraries***.

## Editing the Configuration Member CONFIG

Select the `ADMIN` option on the Main Menu. You are presented with the Administrator Menu which contains all available administration options with a short description of their meaning, for example:

```
---------------------------- ADMINISTRATOR MENU -------------------------------
OPTION  ===>
                                                                 User ID  FHI
                                                                 Time     15:38:16
  _ 1    CONFIG     - Configuration parameters                   Terminal DAEFTCS3
  _ 2    USER       - User maintenance                           Library  NSPF241
  _ 3    MENU LIST  - Display N-ISPF menu list                   Node     148
  _ 4    MENU       - Add/update N-ISPF menu
  _ 5    EXAMPLE    - Invoke example menu
  _ 6    NODES      - Update N-ISPF nodes table

  _ 7    SHORTLIB   - Edit global shortlib member
  _ 8    PANDEF     - Edit PANVALET definition member
  _ 9    LIBDEF     - Edit LIBRARIAN definition member
  _ 10   PDS VERS.  - Edit PDS versioned libraries list
  _ 11   NAT VERS.  - Edit NAT versioned libraries list
  _ 12   VSE VERS.  - Edit VSE versioned libraries list
  _ 13   VERSIONS   - Maintain versioning data

  _ 14   BPSTAT     - Display editor BufferPool status
  _ 15   BP FILES   - List all BufferPool files
  _ 16   BP RECS    - List all Recovery files
 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 :
```

Select the `CONFIG` option on the Administrator Menu. The Configuration Menu appears:

```
---------------------------- CONFIGURATION MENU -------------------------------
OPTION  ===>
                                                                 User ID  FHI
                                                                 Time     15:43:54
 _ 1    N-ISPF      - N-ISPF parameters                          Terminal DAEFTCS3
                                                                 Library  NSPF241
 _ 2    CONFIG      - Edit Config member                         Node     148
 _ 3    CONTROLU    - Edit Site control table

 _ 4    NCP         - Use NCP command processor









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 :
```

This menu contains all available configuration options with a short description of their meaning.

The User Profile Library can contain the member `CONFIG` which defines the installed subsystems and active user exits. If you wish to activate user exits, other subsystems and/or special purpose switches, you must modify the `CONFIG` member.

Without the `CONFIG` member, there are no active user exits and the subsystems enabled are Natural and your site's operating system (z/OS, z/VSE or BS2000). An example of the `CONFIG` member is contained in the System Profile Library. You can copy this example to the User Profile Library using the `SYSMAIN` utility.

The `CONFIG` option on the Administrator Menu provides direct write access to the `CONFIG` member using the Editor. Any modification made to this member will take effect next time you invoke Natural ISPF. It is therefore recommended that you restart Natural ISPF after modification of the `CONFIG` member.

The following subsubsections describe how to define subsystems, activate user exits and activate the logon screen.

- Defining Installed Subsystems
- Activating Natural ISPF User Exits
- Activating Logon Screen
- Versions File Node Number
- Special Purpose Switches - APPLYMOD Parameters Member: APPLYMOD

- Using Comment Lines

## Defining Installed Subsystems

The currently available subsystems are listed in a table in *Subsystems Supported by Natural ISPF* at the end of this documentation.

To enable a subsystem, enter its abbreviation preceded by a plus sign + starting in Column 1 of any line in the `CONFIG` member. You can only enter one subsystem per line.

### Example

The following example of a `CONFIG` member defines a Natural ISPF system with subsystems Natural, z/OS (MVS), SAT and CA Panvalet:

```
EDIT-NAT:SYSISPFU(CONFIG)-Program->Struct-Free-45K ------------ Columns 001 072
 COMMAND===>                                                   SCROLL===> CSR
****** ***************************** top of data ******************************
000010 * Defined subsystems
000020 *
000030 +N  - NATURAL SUBSYSTEM
000050 +M  - MVS SUBSYSTEM
000060 +S  - SAT      SUBSYSTEM
000070 +P  - PANVALET SUBSYSTEM
****** **************************** bottom of data ****************************
```

**Note:** The Natural ISPF menus distributed on your installation medium display the lines relevant to the subsystem(s) activated here; lines (menu options) which have been defined in these menus but which are not related to one of the activated subsystems will be invisible for all users at your site.

## Activating Natural ISPF User Exits

You can activate a user exit for a Natural ISPF object in the `CONFIG` member by entering a 3-character abbreviation in a line preceded by an opening parenthesis ( . Multiple abbreviations separated by blanks can be entered in one line, and multiple lines are possible, all preceded by an opening parenthesis ( .

You can use the following abbreviations to activate the associated exit:

| Abbreviation | Object (Member: TAB-EXIT) |
|---|---|
| ACT | Active jobs |
| BF | BS2000 files |
| BJ | BS2000 jobs |
| BPF | Buffer pool files |
| BPR | Buffer pool recovery files |
| CNF | Configuration object |
| CON | System console |
| CST | Module CSECT |
| CTN | Incore container file |
| DA | z/VSE Active jobs |
| DJ | z/VSE Job |
| DS | Data set |
| DV | z/VSE volume |
| ERR | Natural error |
| FIL | z/VSE File |
| JOB | Job |
| JV | BS2000 job variable |
| LIB | CA Librarian member |
| LMS | BS2000 LMS elements |
| LMV | BS2000 LMS elements version |
| LOG | System log |
| LV | CA Librarian member version |
| MAC | Macro |
| MEM | z/VSE member |
| MNU | Natural ISPF menu |
| MV | Members versions |
| NAT | Natural object |
| NV | NAT member version |
| OUT | Work output |
| PAN | CA Panvalet member |
| PDS | PDS member |
| PV | PDS member version |
| REC | Recovery file |
| SUB | z/VSE sublibrary |
| SYS | Job SYSOUT |
| USR | Natural ISPF user |

| Abbreviation | Object (Member: TAB-EXIT) |
|---|---|
| VIW | Database view |
| VOL | Volume |
| VV | z/VSE member version |

### Other Exits

| Abbreviation | Area of Application (Member: TAB-EXIT) |
|---|---|
| HSM or HSM-S | z/OS migrated data set recall (HSM, DMS archiving facility or similar product). |
| GROUPS | Locate group profiles that apply for users |
| LOGON | Logon procedure |
| LOGOFF | Logoff procedure |
| NODE | Access a Node |
| PRINT | Print |
| PROFIL | Editor profile name |
| RENAME | Rename |
| RESUME | Return to Natural ISPF |
| SESS | Submit or export from edit session |
| SUSP | Suspend Natural ISPF |

**Note:** See also the subsection *Special Purpose Switches* and the detailed descriptions of the user exits in section *User Exits*.

### Example

The following example of a CONFIG member activates the user exits for Natural objects and PDS members.

```
EDIT-NAT:SYSISPFU(CONFIG)-Program->Struct-Free-45K ------------ Columns 001 072
 COMMAND===>                                                    SCROLL===> CSR
****** ***************************** top of data ****************************
000010 * Defined subsystems
000020 *
000030 +N  - NATURAL SUBSYSTEM
000040 +M  - MVS SUBSYSTEM
000050 +P  - PANVALET
000060 *
000070 * Active user exits
000080 *
000090 (NAT PDS
****** **************************** 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
```

This means that when a user issues a specified command for any of these object types, the related user exit is called before the command is executed. For more information on user exits, see section *User Exits*.

### Activating Logon Screen

You can add a line in the `CONFIG` member which causes a logon screen to be displayed when a user invokes Natural ISPF.

The line must start with a dollar sign ($) in column 1 followed by the word `LOGON`:

```
$LOGON
```

You can enter up to two additional parameters in the same line, separated by blanks. The following are possible:

| Parameter | Meaning |
|---|---|
| PROCESS | Causes automatic logon to Entire System Server. |
| USER=*user-id* | Displays the logon screen only to the specified user. This feature is useful if no security check is performed at your site. The user ID specified here should be the ID passed to Natural by the TP monitor. |

### Versions File Node Number

You can optionally add the following parameter in a line of the CONFIG member:

```
-VERSIONS-BY-NODE
```

It is recommended that you use this parameter only if your site has several Entire System Server nodes that access *different* computers in your environment. The node number is then used as part of the identifier of versioning data.

This means that if you have more than one Entire System Server node on one computer, you are strongly advised not to use the VERSIONS-BY-NODE parameter.

### Special Purpose Switches - APPLYMOD Parameters Member: APPLYMOD

Natural ISPF provides some special purpose switches, which for resource reasons can be controlled by setting a switch with a value in the CONFIG member. The general syntax for setting a switch is:

```
APPLYMOD nn = V         /* Comment
```

where:

| Parameter | Meaning |
|---|---|
| *nn* | is a 1 or 2 byte numeric value identifying the particular APPLYMOD. |
| V | is a value assigned to the APPLYMOD to select a particular option. |

The special purpose switches available are described in detail in the following.

Alphabetical List of APPLYMODs:


- APPLYMOD 10
- APPLYMOD 16
- APPLYMOD 18
- APPLYMOD 22
- APPLYMOD 25
- APPLYMOD 26
- APPLYMOD 42
- APPLYMOD 47
- APPLYMOD 48

- APPLYMOD 53
- APPLYMOD 55
- APPLYMOD 57
- APPLYMOD 58
- APPLYMOD 59
- APPLYMOD 63
- APPLYMOD 65
- APPLYMOD 67
- APPLYMOD 68
- APPLYMOD 71
- APPLYMOD 75
- APPLYMOD 80
- APPLYMOD 87
- APPLYMOD 89
- APPLYMOD 90
- APPLYMOD 91
- APPLYMOD 92
- APPLYMOD 95
- APPLYMOD 96
- APPLYMOD 97
- APPLYMOD 101
- APPLYMOD 103
- APPLYMOD 104
- APPLYMOD 105
- APPLYMOD 108
- Example: CONFIG Member


**APPLYMOD 10**

Activates a new syntax and semantic for specifying the `SCAN` parameter for `LIST` functions, which allows you to specify search strings containing blanks even as first or last character. The new syntax is similar to the syntax for the editor command `FIND`, this means that, if the search string contains blanks, it must be embedded in single (') or double (") quotes. Moreover, if the search string is embedded in single quotes, a single quote occurring *within* the search string must be specified as two single quotes (`''` - two characters).

Example - To search for all occurrences of `Marc's` (embedded in blanks), specify either:

```
SC=' Marc''s '
```

or

```
SC=" Marc's "
```

**Notes:**

1. In some cases, the activation of `APPLYMOD 10` causes different results for `SCAN` operations. For example, if `APPLYMOD 10` is not activated, the command `LIST * SC='STAT'` results in a list of all members of the current library containing the string `'STAT'` (embedded in single quotes). When `APPLYMOD 10` is activated, the same command will result in a list of all members containing the string `STAT`, which is consistent with the semantic of the `FIND` command, where `FIND STAT` and `FIND 'STAT'` provide identical results. To list all members containing `'STAT'`, you would have to specify `LIST * SC='''STAT'''` or `LIST * SC="'STAT'"` in this case.
2. Setting `APPLYMOD 10` may also cause command scripts containing these kinds of `SCAN` operations to provide different results.

Possible values for `APPLYMOD 10` are:

| Parameter | Meaning |
|---|---|
| (blank) | Switch is off; `SCAN` results are identical to previous versions. |
| `X` | Switch is on; new logic is enabled. |

Default is `APPLYMOD 10 =` (blank); i.e. switch is off.

**Usage Notes**

You should set this switch if you want to enable your users to scan for strings with trailing blanks, and if there is no need for scan operations to behave consistently with previous versions of Natural ISPF.

### APPLYMOD 16

When working with Natural members, a maximum line length of 88 bytes is used. If you need a line size greater than 88 (for example, for editing or printing), you can enter the new line size here. Possible values are:

| Parameter | Meaning |
|---|---|
| 0 | Switch is off. |
| *nn* | New line size in range from 88 to 233. |

Default is `APPLYMOD 16 = 0` (switch is off).

**Usage Notes**

Set this switch if your Natural programs contain source lines longer than 88 bytes. Note that increasing this value means that Natural edit sessions will consume more space in the Editor Buffer Pool.

### APPLYMOD 18

z/OS only: deactivates the selection of SYSOUT files for a specific output class. That is, all SYSOUT files for a given job are displayed in a list of SYSOUT files. Possible values are:

| Parameter | Meaning |
|---|---|
| (blank) | Switch is off. Listed SYSOUT files will be selected by job number and the assigned output class. |
| X | Switch is on. In a list of SYSOUT files, all files of a job will be listed. |

Default is `APPLYMOD 18 =` (blank); i.e. switch is off.

**Usage Notes**

It is recommended that you set this switch only when notified by Software AG. In general, a difference between the two settings can only be seen with jobs that create output in more than one output class.

### APPLYMOD 22

Activates the extended Natural/`USPOOL` interface under Com-plete. Using `USPOOL` functionality, for example, a logical printer driver can be invoked. Possible values are:

| Parameter | Meaning |
|---|---|
| (blank) | Switch is off. A `DRIVER` parameter specified with any `PRINT` command issued by a Natural ISPF user is interpreted as the name of a printer control character table (as defined with the `NTCC` macro - for further information, see the Natural documentation). |
| X | Switch is on. When executing under Com-plete, a `DRIVER` parameter specified with any `PRINT` command issued by a Natural ISPF user is interpreted as the name of a Logical Output driver routine for Com-plete. See the subsection *Logical Output Drivers in Section 3: TIBTAB-Terminal Information Block Table* of your *Com-plete System Programmer's Guide*. |

Default is `APPLYMOD 22 =` (blank); i.e. switch is off.

**Usage Notes**

For installation of this feature, read also the related subsection of *Optional Interfaces to Software AG Products* in the *Installation* documentation.

### APPLYMOD 25

Deactivates the compression of LMS member list under BS2000. Possible values are:

| Parameter | Meaning |
|---|---|
| (blank) | Switch is off. |
| X | Switch is on. |

Default is `APPLYMOD 25 =` (blank); i.e. switch is off.

**Usage Notes**

It is recommended that you set this switch only when notified by Software AG.

### APPLYMOD 26

Use this switch to control how Natural ISPF reacts, if the Natural source area is found non-empty at the time when Natural ISPF is started, or if it is first suspended (for example, with the session command Natural) and then re-entered. Possible values are:

| Parameter | Meaning |
|---|---|
| N | Switch is off. This setting ensures that Natural ISPF acts in a way fully compatible with previous versions, this means that the source area contents are ignored in the situations described above. |
| X | Switch is on. Natural ISPF opens an `EDIT` session for the object found in the source area, without prompting. When re-entering the program in this way, Natural ISPF is suspended again as soon as the new edit session is ended (users are prompted whether this is intended). This is the default setting. |
| P | Switch is on, but users will be prompted if it is intended to open an `EDIT` session for the object found in the source area. |
| Y | Switch is on, but suspension is performed without prompting. |
| Z | Switch is on, but no automatic suspension is performed. |

Default is `APPLYMOD 26 = X` (switch is on).

**Usage Notes**

Set this switch to `N`, if Natural ISPF is invoked from applications that use the source area for text generation and if you do not want users to modify the source area lines from Natural ISPF. You can also set the switch to `Y` or `Z` to suppress prompting in the situations indicated above.

### APPLYMOD 42

Activates Con-nect Inbasket checking, when the Software AG office system is installed and the appropriate subsystem is activated in the Natural ISPF configuration member (see the subsection *Defining Installed Subsystems*). Possible values are:

| Parameter | Meaning |
|---|---|
| 0 | Switch is off. |
| *nn* (<=150) | Time interval in minutes after which your Con-nect Inbasket is checked. If the number of new items in the Inbasket has changed since the last check, the user is notified by a message. No error message is issued if the user does not have a personal Con-nect cabinet. |
| *nn* (>150) | This option is similar to the above, that is, Inbasket checking is activated, but (*nn*-150) is used as the time interval in minutes and an error message is issued if the user does not have a personal Con-nect cabinet. |

Default is `APPLYMOD 42 = 0` (switch is off).

#### Usage Notes

Use this switch when Con-nect is installed and you want users to be notified of incoming Con-nect messages.

### APPLYMOD 47

z/OS only: when browsing job output, this switch controls whether or not all SYSOUT data sets are shown as one file. Possible values are:

| Parameter | Meaning |
|---|---|
| (blank) | Switch is off - SYSOUT data sets are shown as separate files. |
| X | Switch is on - SYSOUT data sets are shown as one file. |

Default is `APPLYMOD 47 = 0` (switch is off).

#### Usage Notes

Set this switch if this function is requested at your site.

### APPLYMOD 48

When browsing BS2000 files and/or LMS elements, this switch controls whether the data is held in the Editor buffer pool or if the session is handled as “external”, that is, data is read from disk every time when scrolling or scanning (`FIND` operations) is performed. Possible values:

| Parameter | Meaning |
|---|---|
| `N` | Switch is off - data is held in the Editor buffer pool. |
| `F` | Activates external `BROWSE` mode for Files. |
| `L` | Activates external `BROWSE` mode for LMS elements. |
| `X` | Activates external `BROWSE` mode for Files and LMS elements. |

Default is `APPLYMOD 48 = X` (switch is on, both for files and LMS elements).

**Usage Notes**

It is recommended that you modify this switch only when notified to do so by Software AG. Note that the default value of this switch has changed compared with older versions 1.4.1 and 1.4.2 of Natural ISPF.

### APPLYMOD 53

Reduces Entire System Server (ESY) calls to check whether a data set is a GDG.

| Parameter | Meaning |
|---|---|
| (blank) | Check for GDG. |
| `X` | Do not check. |

Default is `APPLYMOD 53 =` (blank); i.e. check for GDG.

**Usage Notes**

Set this switch if you are not using Natural ISPF together with GDG (Generation data sets).

### APPLYMOD 55

Avoids timeout of Editor session 40, which contains Natural ISPF internal data, by doing `STATUS` calls every 10 minutes. Possible values are:

| Parameter | Meaning |
|---|---|
| (blank) | Status calls not active. |
| X | Status calls active. |

Default is `APPLYMOD 55 =` (blank), i.e. no status calls.

**Usage Notes**

Set this switch if you reduced the Delete file timeout value for the Editor Buffer Pool and some of your user get error messages like `Write to BP failed`.

#### APPLYMOD 57

Bypass for Adabas calls exceeded when printing empty SYSOUT data sets.

| Parameter | Meaning |
|---|---|
| (blank) | Bypass not active |
| X | Bypass active |

Default is `APPLYMOD 57 =` (blank) (bypass not active).

**Usage Notes**

It is recommended that you set this switch only when notified by Software AG.

#### APPLYMOD 58

Activates a general limit for Editor `FIND` commands, when editing or browsing PDS members, LMS elements, sequential data sets or SYSOUT data sets. A `FIND` command will then display a message after scanning 5000 records, if you want to continue the search the command `RFIND` must be entered. The limit can always be modified (see also Editor command `LIMIT`).

| Parameter | Meaning |
|---|---|
| (blank) | No limit for `FIND` command. |
| X | Default limit (5000) is activated. |

Default is `APPLYMOD 58 =` (blank) (no limit).

**Usage Notes**

Set this switch if you are working with large data sets in a TP environment and `FIND` commands consume a lot of CPU and cannot be interrupted.

### APPLYMOD 59

Activates full expiration date checking before writing to a data set. If you are working with expiration date and a data set has not yet expired, a prompt warns you that you are going to write to this data set.

| Parameter | Meaning |
|---|---|
| (blank) | No expiration date checking. |
| X | Full expiration date checking is active. |

Default is `APPLYMOD 59 =` (blank) (no check).

**Usage Notes**

Set this switch if you are using expiration date to protect your data sets and you want to avoid overwriting and operator messages. If you are working without expiration date in most of your data sets, you should not set this `APPLYMOD`.

### APPLYMOD 63

Activates display of `REAL-RECORD-COUNT` (number of records, including control records) in z/VSE job lists.

| Parameter | Meaning |
|---|---|
| (blank) | Use `RECORD-COUNT`. |
| X | Use `REAL-RECORD-COUNT`. |

Default is `APPLYMOD 63 =` (blank).

**Usage Notes**

Set this switch if you want to see the real number of records in your z/VSE job lists.

### APPLYMOD 65

Defines whether any fields in the user defaults cannot be inherited from a group. This means that if the field is not defined in the user profile, Natural ISPF will not read group profiles to find a value for this profile field.

To activate this switch, you should assign a numeric value lower than 64; this value will be interpreted as bit-coded. This means, that for each bit set in the binary representation of that number, Natural ISPF will not search group profiles for a definition of the corresponding profile field, as shown in the table below. Of course, any definition found in the user profile of the individual user will always be honored, regardless of the value assigned to this switch.

| Parameter | Meaning |
|---|---|
| .......1 | INIT-LOGON |
| ......1. | Default PRINTER |
| .....1.. | Default FILE-TYPE |
| ....1... | Default DSNAME |
| ...1.... | Initial COMMAND |

Default is APPLYMOD 65 = (blank) (all fields are inherited).

**Usage Notes**

Set this switch, if the performance of the Natural ISPF initialization phase is dissatisfying, and if none or only some of the above-mentioned fields need to be inherited from group profiles.

**Example**

```
APPLYMOD 65 = 5
```

If a user logs on to Natural ISPF and the profile does not contain a value for the fields INIT-LOGON and FILE-TYPE, Natural ISPF will not search for the group profiles of this user for these fields.

### APPLYMOD 67

You can set this APPLYMOD to prevent concurrent editing of a z/VSE member. It activates a check as to whether the member has been modified somewhere else since start of the edit session. If this is the case, the SAVE command returns an error message.

Default is APPLYMOD 67 = (blank) (SAVE is always executed).

**Usage Notes**

Set this switch to avoid concurrent editing of a z/VSE member.

**Example**

```
APPLYMOD 67 = X
```

### APPLYMOD 68

Defines whether a warning is displayed whenever a user tries to access a data set which has been migrated by HSM, DMS or a similar archiving system. The user can cancel the action to avoid a RECALL or can continue processing.

| Parameter | Meaning |
|---|---|
| X | Users are prompted for confirmation before recalling a migrated data set. |
| (blank) | Same as X. |
| D | Prompting takes place *only* for data sets showing ARCIVE in the VOLSER field of the catalog entry (DMS). |
| H | Prompting takes place *only* for data sets showing MIGRAT in the VOLSER field of the catalog entry (HSM). |
| N | No warning window opens. An error message is issued which informs the user that the data set is not available. |
| C | Users are prompted for confirmation before recalling a migrated data set. The VOLSERs that indicate a migrated data set are read from the CONFIG member. Enter up to four VOLSERs preceeded by an underscore (_) and set APPLYMOD 68 to character C. |

Default is APPLYMOD 68 = (blank) (users will be prompted before recall).

**Usage Notes**

1. For the default setting, it is recommended that the Entire System Server startup parameter RECALL be set to NO for performance reasons.
2. Asynchronous recalling is not part of standard Natural ISPF features but can easily be implemented by coding an appropriate job submission within the HSM user exit of Natural ISPF (see section ***User Exits***). Do not set APPLYMOD 68 to N if you have activated this user exit.
3. Set this switch to N (as in the example below) if you are using HSM or a similar product and you want to deny Natural ISPF users the right to recall and access migrated data sets.
4. If you are not using HSM or a similar product, setting this switch to N will improve performance if your Entire System Server startup parameters do not contain the recommended setting RECALL=NO. This is because if the default setting RECALL=YES is in effect, each data set must be checked for migration before its file attributes can be queried from Entire System Server.

**Example**

```
APPLYMOD 68 = N
```

```
APPLYMOD 68 = C
_MIGRAT
_ARCIVE
```

### APPLYMOD 71

Defines whether windowing is suppressed when Natural ISPF is executed in batch. When executing Natural ISPF in batch, this switch can be used to overcome some Natural problems with windowing. This switch is evaluated in batch only.

Default is `APPLYMOD 71 =` (blank) (windowing active in batch).

**Usage Notes**

It is recommended that you set this switch only when notified to do so by Software AG.

**Example**

```
APPLYMOD 71 = X
```

### APPLYMOD 75

Improves performance with export PC for large data sets. Prompt for PC file name is displayed without delay in minutes.

| Parameter | Meaning |
|---|---|
| (blank) | Prompt after reading to the end of the file to determine the number of records contained. |
| X | Prompt immediately. |

Default is `APPLYMOD 75 =` (blank) (no improvement).

**Usage Notes**

Set this switch to export large data sets or members to your PC.

### APPLYMOD 80

Improves performance with the CC function for BS2000 jobs. This function searches for job variables related to a specific job.

| Parameter | Meaning |
|---|---|
| (blank) | Extended search is performed. The CC function checks the contents of all existing job variables within the current BS2000 user ID and reports their values, if they contain the specified job ID. |
| X | The search is restricted to those job variables that contain the specified job ID as part of their names, and to a monitoring job variable specified when submitting the job, if any. |

Default is APPLYMOD 80 = (blank).

**Usage Notes**

Set this switch if many job variables are defined in your environment, and if performance of the CC function is not satisfactory. Note however, that the CC function will then provide reasonable results only for jobs that were submitted with a specified monitoring job variable, or for jobs creating job variables that contain the job ID (TSN) as part of their names, for example:

**Example**

```
/    DCLJV  JV.ISPUSER.ASMASS.&N..&($SYSJV.TSN),LINK=*CCASS
```

### APPLYMOD 87

Modifies the header line of EDIT/BROWSE and LIST sessions. The node number is displayed in the header if it is different from the default node.

| Parameter | Meaning |
|---|---|
| (blank) | Node will not be displayed in header. |
| X | Node will be displayed. |

Default is APPLYMOD 87 = (blank).

**Usage Notes**

Set this switch if you are in a multi-CPU environment with different Entire System Server nodes and you are accessing different nodes from the same Natural ISPF environment.

### APPLYMOD 89

Makes the handling of data entered in the Natural Objects Entry Panel identical to previous versions of Natural ISPF (although different from the general logic).

| Parameter | Meaning |
|---|---|
| (blank) | Fields of the Natural Objects Entry Panel will be filled with the values contained when the user left this panel the last time. |
| X | Field values from the last use are filled in only if these values do not refer to a library different from the current library, as shown on the Natural ISPF Main Menu (logic compatible with ISP 1.4). |

Default is `APPLYMOD 89 =` (blank).

**Usage Notes**

Set this switch if your users prefer the old way of handling the Natural Objects Entry Panel.

### APPLYMOD 90

Controls the way in which the `BR-CONSOLE` session is to be filled in environments where the views `CONSOLE` and `CONSOLE-LOG` are both supported (i.e. in z/OS and z/VSE environments). Possible values are:

| Parameter | Meaning |
|---|---|
| (blank) | Automatic selection. Natural ISPF will choose the access method which best suits the environment of the node being addressed. |
| L | Session is filled using the view `CONSOLE-LOG`. Not supported for z/OS/JES3 environments. The local command `LINES` is supported. |
| N | Session is filled using the new `CONSOLE` function, supported by Entire System Server version 2.1.1 onwards only for the environments z/OS/ESA SP 5.1.0 onwards or z/VSE SP 06.01 onwards. The local command `LINES` is supported to extend the `CONSOLE` session. |
| O | Session is filled using old `CONSOLE` function, which is restricted to the size of one screen. The local command `LINES` is not supported. |

Default is `APPLYMOD 90 =` (blank) (automatic selection).

**Usage Notes**

It is recommended that you set/modify this switch only when notified to do so by Software AG.

### APPLYMOD 91

Controls whether or not activating the session exit ISP--S-U will also cause this exit to be invoked when an EXPORT function is about to be executed.

| Parameter | Meaning |
|---|---|
| (blank) | Exit will be invoked for SUBMIT functions only, as described in section *Session Exit ISP--S-U* in this documentation. |
| X | Exit will be invoked both for SUBMIT and EXPORT functions, thus making it possible to disallow a specific EXPORT operation, for example, after the size of the file being exported has been checked. |

Default is APPLYMOD 91 = (blank) (exit used for SUBMIT only).

### APPLYMOD 92

Setting this APPLYMOD will enable access to DYNAM-D-controlled data sets.

**Note:** This option is *not* a part of released N-ISPF functionality.

| Parameter | Meaning |
|---|---|
| (blank) | Access to DYNAM-D-controlled data sets is rejected. This is the default. |
| X | READ and WRITE access to DYNAM-D-controlled data sets is enabled. This allows the functions EDIT, BROWSE, PRINT, EXPORT and COPY. Other functions (e.g. RENAME, DELETE) will still be rejected. |

**Caution:** Natural ISPF cannot check in advance whether or not a specific DYNAM-D-controlled data set exists. Therefore, specifying a non-existent dsname with function EDIT or BROWSE may cause an attempt to read that data set, which would generate a console message. We recommend not activating APPLYMOD 92 unless all of your users (at least those who are authorised to access VSE files) and all operators are informed and aware of this fact. If you wish to introduce additonal security measures in this context (e.g. reduce the number of users that are allowed to access DYNAM-D data sets), consider using the ISPF object exit ISDD---U for this purpose.

### APPLYMOD 95

When working with z/VSE job output, a maximum line length of 133 bytes is used for browsing the job output lines, and a line length of 241 bytes is used for printing, exporting or copying job output data. If you need a line size greater than 133 for browsing, and/or greater than 241 for printing etc., you can enter the new line size here.

| Parameter | Meaning |
|---|---|
| 0 | Switch is off - the default values are in effect. |
| *nnn* | New line size in the range from 133 to 253. (If the value is less than 241, it will affect BROWSE sessions only; for other functions, 241 will be used). |

Default is APPLYMOD 95 = 0 (switch is off).

**Usage Notes**

Set this switch if your POWER output data contain lines longer than 133 bytes.

#### APPLYMOD 96

To edit and save CA Librarian members, which contain CA Librarian control cards, starting with a hyphen (-) , the hyphen must be internally replaced with an equal sign (=) in column 1 of any data line. Set this switch to activate this substitution.

Possible values are:

| Parameter | Meaning |
|---|---|
| (blank) | Switch is off - no substitution is performed. |
| X | Switch is on - substitution is activated |

Default is APPLYMOD 96 = (blank) (switch is off).

**Usage Notes**

Set this switch if your CA Librarian members contain data lines starting with the escape character, a hyphen (-), which is usually reserved for CA Librarian control cards.

#### APPLYMOD 97

This switch can cause an extended map to be used when the command NATP-LOGON (LOGON to an Entire System Server Node) is executed, allowing users to specify not only a user ID and a password but also Account Information. This can be useful in z/OS environments (for writing SMF records) or in a BS2000 environment for additional security checks, especially when the new startup parameter setting SECURITY=BS2A is being used.

Possible values are:

| Parameter | Meaning |
|---|---|
| (blank) | Switch is off: NATP-LOGON does not allow ACCOUNT specification. |
| X | Switch is on: NATP-LOGON allows ACCOUNT specification and checking. |

Default is APPLYMOD 97 = (blank) (switch is off).

**Usage Notes**

Set this switch if you have specified SECURITY=BS2A in your Entire System Server startup parameters, and if your Entire System Server version allows it (see note above).

### APPLYMOD 101

By default in Natural ISPF group profiles are searched for with prefix logic. This switch, on the other hand, can be used to search for user group profiles that are derived from user groups defined in Natural Security.

Possible values are:

| Parameter | Meaning |
|---|---|
| (blank) | Switch is off. Group profiles are searched for with prefix logic, as in previous versions of Natural ISPF. |
| S | Switch is on. Group profiles are derived from Natural Security definitions (both privileged and non-privileged groups). |
| P | Switch is on. Group profiles are derived from Natural Security definitions (from privileged groups only). |

Default is APPLYMOD 101 = (blank) (switch is off).

**Usage Notes**

Set this switch if Natural Security is installed, and if you would like profile characteristics that are not defined for a specific user to be inherited from profile definitions made for a user group that contains the specific user. Leave the switch unchanged if you would like profile characteristics defined for a matching user prefix (for example, AB* for user ABEG) to apply for these users.

**Note:** If no matching group profile is found, but a profile definition for the default user * exists, this definition will be inherited, regardless of the setting of the above APPLYMOD parameter.

#### APPLYMOD 103

When displaying Natural map layouts with the function command `FORMAT`, the maximum line length is set to the current value of Natural's system variable `*LINESIZE`. If you need a greater line size for browsing, printing, or performing other functions on formatted maps, you can enter the new line size here.

Possible values are:

| Parameter | Meaning |
|---|---|
| `0` | Switch is off. |
| *nn* | New line size in the range from 80 to 250. |

Default is `APPLYMOD 103 = 0` (switch is off).

**Usage Notes**

Set this switch if your Natural maps have more columns than the displayable line size in your Natural ISPF environment. You will then be able to use the scroll commands `RIGHT` and `LEFT` to display the columns which do not fit on the screen.

#### APPLYMOD 104

As of Natural ISPF Version 2.4.1, the menu structure is cursor-sensitive. To select a menu item you can either mark it with an `X`, or simply place the cursor in front of it. To use menu screens as in earlier versions, without cursor-sensitivity, set this parameter to `X`.

Possible values are:

| Parameter | Meaning |
|---|---|
| (blank) | Menus are cursor-sensitive. |
| `X` | Menus are not cursor-sensitive. |

Default is `APPLYMOD 104 =` (blank) (switch is off).

#### APPLYMOD 105

In some cases, JES does not count all records of a SYSOUT file. This occurs if the compiler or the assembler outputs contain control records. The `BOTTOM` command of Natural ISPF does not point to the end of the SYSOUT file if these outputs report a record count without the control records.

This `APPLYMOD` instructs Natural ISPF not to rely on the record count reported by JES but to count the records themselves. Hence this may take some time, enable this switch if best performance is desired.

Possible values are:

| Parameter | Meaning |
|---|---|
| (blank) | Take record count from JES. |
| X | Count all records. |

Default is APPLYMOD 105 = (blank) (switch is off).

### APPLYMOD 108

Affects the ALLOCATE function command for z/OS data sets. The default value for the RLSE parameter is NO. If this APPLYMOD is used, the RLSE option defaults to YES instead of NO.

Possible values are:

| Parameter | Meaning |
|---|---|
| (blank) | Default value of the RLSE parameter is NO. |
| X | Default value of the RLSE parameter is YES. |

Default is APPLYMOD 108 = (blank) (switch is off).

### Example: CONFIG Member

The following is an example of the CONFIG member with the APPLYMODs entered. Note that in this example, some APPLYMODs are activated, others are commented out (see also the subsection *Using Comment Lines* below).

```
EDIT-CNF:SYSISPFU(CONFIG) ------------------------------------ Columns 001 072
 COMMAND===>                                                  SCROLL===> CSR
000400 * The next line will determine whether to input logon-screen at the
000410 * begining of NSPF.
000420 * PROCESS keyword is optional and means to logon to natural process
000430 * USER=xxx will ask for logon only if user = xxx
000440 * $LOGON PROCESS
000450 * APPLYMOD 16=132
000460 * APPLYMOD 18=X
000470 APPLYMOD 22=X
000480 APPLYMOD 42=5  /* Inbasket every five minutes
****** ***************************** bottom of data ****************************
```

### Using Comment Lines

You can enter comment lines in the CONFIG member to provide information or explanations of entries. Comment lines must start with an asterisk (*) in Column 1 of the line.

To deactivate any subsystem or user exit(s), it may be useful to turn the entry in the CONFIG member into a comment by entering an asterisk (*) in the first column of the corresponding line. This gives you a better overview of active and disabled items, and makes it easy to reactivate any available item.

When defining exits or APPLYMODs, you can enter comments at the end of the line if they are preceded by /* (see also the example above).

## Editing the CA Panvalet Definition Member

The definitions for the Natural ISPF/CA Panvalet interface must be contained in the PANDEF member in the User Profile Library.

You can change some of these definitions. However, before you modify the PANDEF member, copy the example member from the System Profile Library to the User Profile Library. Then:

1. Enable access to CA Panvalet in the CONFIG member (see the subsection *Editing the Configuration Member CONFIG*).
2. Restart Natural ISPF.
3. The Administrator Menu now includes the PANDEF option. Select this option to give you direct write access to the PANDEF member.

### Defining the CA Panvalet Update Method

In the PANDEF member, you can specify how CA Panvalet members are updated. Any of the following options are possible, entered in a line of the PANDEF member starting in Column 1:

| Option | Explanation |
|---|---|
| -UPDATEMODE | Default. An update deck containing only the modified lines of the CA Panvalet member is created and used for updates. |
| -REPLACEMODE | The whole CA Panvalet member is replaced. |
| -MIXMODE | Both modes UPDATE and REPLACE are allowed. When the SAVE command is issued, Natural ISPF will select the method that will cause the fewest lines to be sent to CA Panvalet. |

### Activating the Save Exit ISPT-SVU

If your installation requires special control for all CA Panvalet updates, you can activate the ISPT-SVU user exit. This exit receives control after each successful CA Panvalet update (PAM#1), and can be used to obtain the last output from CA Panvalet.

The ISPT-SVU exit is a Natural subprogram and can access Entire System Server to obtain the CA Panvalet output, analyze it, and write it to a data base, file or SPOOL file.

A sample exit is distributed in the Natural ISPF Exit Library. To activate this exit, copy your ISPT-SVU object to SYSLIB using the utility SYSMAIN and enter:

```
$SAVEEXIT
```

in Column 1 of any line of the PANDEF member. For a list of definable parameters, see the section *User Exits*.

### Defining Mandatory Fields for New CA Panvalet Members

You can define which parameter fields on the CA Panvalet Entry Panel are mandatory when users add a new member to CA Panvalet (for example, the parameters USER, LANG, COMMENT). Users must then enter a valid value in these input fields on the CA Panvalet Entry Panel or specify the corresponding object parameters when using the EDIT function command syntax for a new member.

The following text entered in the PANDEF member starting in Column 1 defines the corresponding parameters as mandatory:

```
/USER
/LANG
/COMMENT
```

### CA Panvalet Language Definition

The PANDEF member contains language and sequence number definitions. You must ensure that the language definitions (sequence numbers and message line) correspond to your CA Panvalet definitions, as Natural ISPF uses them to create update decks (in UPDATE and MIX mode). You can check these definitions by displaying a CA Panvalet member using the Editor in BROWSE mode.

In the example of a PANDEF member that follows, these language definitions follow the fields defined as mandatory when users create new CA Panvalet members:

```
 EDIT-CNF:SYSISPFU(PANDEF) ----------------------------------- Columns 001 072
  COMMAND===>                                                  SCROLL===> CSR
****** ***************************** top of data ****************************
 000010 * PANVALET LANGUAGES TABLE
 000020 -UPDATEMODE   ( OR -REPLACEMODE   -MIXMODE )
 000030 * IF THE NEXT LINE STARTS WITH $ USER EXIT AFTER SAVE IS INVOKED
 000040 $SAVEEXIT
 000050 * THE FOLLOWING LINE CONTAIN FIELDS THAT ARE ESSENTIAL FOR ADD MEMBER
 000060 /USER
 000070 /LANG
 000080 /COMMENT
 000090 *SCREEN LANG
 000100 *     !COMMAND LANG
 000110 *     !         !SEQ START - END
 000120 *     !         !       !GENERATED MESSAGE LINE
 000130 !AUTOC!AUTOCODER!001-005!N
 000140 !ASMB !BAL      !073-077!Y
 000150 !COBOL!COBOL    !001-006!N
 000160 !ANSCB!ANSCOBOL !001-006!N
 000170 !COB72!COBOL72  !001-006!N
 000180 !FORT !FORTRAN  !073-077!N
 000190 !PL/1 !PL/1     !073-077!N
  Enter-PF13--PF14--PF15--PF16--PF17--PF18--PF19--PF20--PF21--PF22--PF23--PF24---
Help  SORT  End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

**Note:** The above example was created using a German-language keyboard. For the separator (`!`), use the character that corresponds to `X'4F'`.

## Editing the CA Librarian Definition Member

The definitions for the Natural ISPF/CA Librarian interface must be contained in the `LIBDEF` member in the User Profile Library.

You can change some of these definitions. However, before you modify the `LIBDEF` member, copy the example member from the System Profile Library to the User Profile Library. Then:

1. Enable access to CA Librarian in the `CONFIG` member (see the subsection ***Editing the Configuration Member CONFIG***).
2. Restart Natural ISPF.
3. The Administrator Menu now includes the `LIBDEF` option. Select this option to give you direct write access to the `LIBDEF` member.

The following topics are covered below:

- Defining Mandatory Parameters for New or Existing CA Librarian Members
- Handling of the PROGRAMMER Attribute during Member Update

- Expansion of -INC Statements
- CA Librarian Language Definition

### Defining Mandatory Parameters for New or Existing CA Librarian Members

You can define which parameter fields on the Librarian Entry Panel are mandatory when users wish to edit a CA Librarian member (for example, the parameters `PROGRAMMER` and `LANGUAGE` for new members). Users must then enter a valid value in these input fields on the Librarian Entry Panel or specify the corresponding object parameters when using the `EDIT` function command syntax for a new member.

Additionally, you can specify other mandatory parameters that do not appear on the Librarian Entry Panel, but the user is prompted by a window when requesting an edit session:

- If you specify the `DESCRIPTION` parameter, the user, when requesting to edit a new member, is prompted by a window in which he can describe the new member. This description appears on a list of members in the appropriate column.
- If you specify the `HISTORY` parameter, the user, when requesting an edit session with an existing member, is prompted by a window in which he must enter the reason for changing the member.

The following text entered in the `LIBDEF` member starting in Column 1 defines the corresponding parameters as mandatory:

| Option | Explanation |
|---|---|
| `/PGMR` | `PROGRAMMER` field on Entry Panel. |
| `/LANG` | `LANGUAGE` field on Entry Panel. |
| `/DESC` | `DESCRIPTION` prompt. |
| `/HIST` | `HISTORY` prompt (reason for change). |

### Handling of the PROGRAMMER Attribute during Member Update

You can control the way in which a member's `PGMR` attribute is handled during update (that is, when saving the contents of an `EDIT` session) by specifying one of the following entries, instead of the entry `/PGMR` mentioned in the subsection above.

**Note:** Each of these entries must start in Column 1.

| Option | Explanation |
|---|---|
| /PGMR=STD | This is identical to a plain /PGMR entry, that is, the PGMR attribute is left unchanged during update (and must be specified when adding a new member). |
| /PGMR=OPTUPDATE | Specifies similar handling for ADD (prompt, if field is empty). However, if for an update, the PGMR field is either explicitly specified by the user or supplied by an active user exit, ISPL---U, it will be passed to CA Librarian for updating. |
| /PGMR=NATUSERID | Specifies that *USER will be used as PGMR attribute both for adding new members and updating existing ones. No prompting will take place. |
| /PGMR=NSCUNAME1 | This is like NATUSERID, but PGMR will be based on the Natural Security definition (first "word" of *USER-NAME). |
| /PGMR=NSCUNAME2 | This is like NSCUNAME1, but the PGMR attribute will be formed from the first character plus last "word" of *USER-NAME separated by a period (.). |

### Expansion of -INC Statements

If you wish to have -INC statements expanded by default when browsing CA Librarian members, add the following entry to the LIBDEF member:

```
/EXPAND
```

### CA Librarian Language Definition

Apart from your own site-specific definitions described above, the LIBDEF member contains language and sequence number definitions which you must not modify.

Below is an example of a LIBDEF member.

```
EDIT-NAT:SYSISPFU(LIBDEF)-Program->Struct-Free-44K -- >>> Versioning is invoked
 COMMAND===>                                                  SCROLL===> CSR
****** ***************************** top of data ****************************
000010 * THE FOLLOWING LINES DEFINE MANDATORY FIELDS FOR NEW MEMBERS:
000020 /LANG
000030 /DESC
000040 * /HIST - THIS LINE IS INACTIVE (NOT STARTING IN COLUMN 1)
000050 *
000060 * THE FOLLOWING LINE SPECIFIES THAT PGMR WILL ALWAYS BE SET TO
000070 * THE CONTENTS OF THE NATURAL SYSTEM VARIABLE *USER:
000080 /PGMR=NATUSERID
000090 *
000100 * LIBRARIAN LANGUAGES TABLE
000110 *CODE LANG
000120 *   !SYNONYM
000130 *   !       !SEQ START - END
000140 *   !       !
000150 !ASM!ASM    !073-080
000160 !BAS!BASIC  !000-000
000170 !CMD!CLIST  !000-000
000180 !JCL!CNTL   !073-080
000190 !COB!COBOL  !001-006
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:**

1. The above example was created using a German-language keyboard. For the separator (`!`), use the character that corresponds to `X'4F'`.
2. In the above example, the `HISTORY` parameter is commented out. This means that users are not prompted to give a reason for modifying a member.

## Defining Short IDs for Libraries

In the User Profile Library, you can create a text member `ULIBID`, in which you can define two-character abbreviations for data set names or file names.

Users can use short IDs when addressing data sets/libraries in the input fields of Entry Panels or in the object parameters of function command syntax.

The short IDs defined here are valid system-wide (global short names). However, users can define their own short IDs in their user profile, for example by using the `SHORTLIB` command. When a short ID is used, the user profile is searched first. For further information see the following subsections in the *Natural ISPF User's Guide*:

- description of the `SHORTLIB` command in section *Command Reference*;

- *Library Definition* in section *Profile Maintenance*.

To define global short library names, select the SHORTLIB option from the Administrator Menu. This gives direct write access to member ULIBID in Editor format.

Example of global short library name definitions:

```
EDIT-NAT:SYSISPFU(ULIBID)-Program->Struct-Free-45K -- >>> Versioning is invoked
 COMMAND===>                                                 SCROLL===> CSR
****** ****************************** top of data ******************************
000010 * table with global shortlibs
000020 SP SYSM.PROCLIB SYSXA1
000030 EL EDITOR.COMN.IV135.LOAD
000040 NL FRZ.NAT215.MVSLOAD
000050 OL OPS.SYSF.PROD.LOAD
000060 OJ OPS.COMN.JCL
000070 OA OPS.AKTUELL.LINKLIB
000080 AL OPS.SYSF.V5.ADALOAD
000090 CL PUB.SYSF.USER.LOAD
****** ***************************** 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
```

Each entry consists of a two-character short name followed by a blank followed by the full library name. If the library is not cataloged, the library name must be followed by a blank and the volume serial number to eliminate the catalog request at access time.

Lines beginning with an asterisk (*) are interpreted as comment lines.

When defining short names for libraries, it is advisable to observe naming conventions for easy identification. The following convention is suggested:

- The first character could describe the system or product, for example, A for Adabas.
- The second could describe the library type, for example, L for LOAD, S for SOURCE, J for JCL.

Using this convention, the library short name AL could easily be identified as the Adabas load library currently in use.

## Natural ISPF Parameters

You can set the default parameters for the Natural ISPF system by selecting the N-ISPF option from the Configuration Menu and then entering the required value in the following parameter table:

```
------------------------------ NSPF PARAMETERS --------------------------------
COMMAND ===>


   NATPROC DBID   148   ( Natural Process dbid)

   MACRO CHAR     §     ( Character for macro program facility)
   MACRO SMODE    S     ( Mode (Struct,Report) for non Natural macro)

   VERSIONS DBID    9   ( Versions file dbid)
   VERSIONS FNR    33   ( Versions file number)
   MAX VERSIONS    21   ( Maximum versions for member in versions file)

   PROCESSOR ID   A     ( NCP command processor id to be used)

   NOM PRINTER
                        ( Printer handled by ENTIRE OUTPUT MANAGEMENT)




 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 Parameters

| Parameter | Member: ISPPARAM |
|---|---|
| NATPROC DBID | Entire System Server data base ID as defined in the NTDB macro in the NATPARM module. This field is for information only and cannot be modified. |
| MACRO CHAR | Special character to be used for macro statements in objects that use the Macro facility. |
| MACRO SMODE | Defines the mode of a non-Natural object that uses the Macro facility (for example, a JCL member stored as PDS member). Possible values:<br><br>S (STRUCT), R (REPORT) |

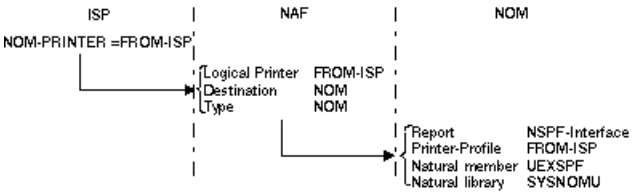
| Parameter | Member: ISPPARAM |
|---|---|
| | This system parameter cannot be modified by a user while editing. |
| VERSIONS DBID* | Physical DBID of versioning data. |
| VERSIONS FNR* | File number of versioning data. |
| MAX VERSIONS | Maximum number of previous versions kept for any one member. At the next SAVE, the oldest version is deleted. |
| PROCESSOR ID | Last byte of the NCP command processor to be used. For details, see the subsection *NCP Concept*. |

* All versioning data (modified lines, known as "update decks", of Natural members, PDS members and z/VSE members) are stored in an Adabas or VSAM file. When a user selects a previous version from a list, the current object version with the appropriate update deck is displayed. You must specify the file by data base ID and file number here. You can use the Natural system file (FNAT) to store the versioning data, but it is strongly recommended that you load a separate file (see also the *Installation* documentation).

The VERSIONS DBID and VERSIONS FNR fields must be set once after the versioning file has been installed to activate versioning.

**Note:** When the NTFILE/LFILE parameter is used, regardless if with or without PASSW, the values for VERSIONS DBID and VERSIONS FNR in the Natural ISPF parameters are ignored, but some values must be set in these fields to activate versioning. It is highly recommended to use the correct numbers to prevent confusion.

| Parameter | Meaning |
|---|---|
| NOM PRINTER | To use the extended interface between Natural ISPF and Entire Output Management (NOM), enter the logical printer profile which is handled by NOM. Prerequisite is the NAF/NOM interface. Assume your NOM PRINTER is FROM-ISP. The following table shows the required definitions. |

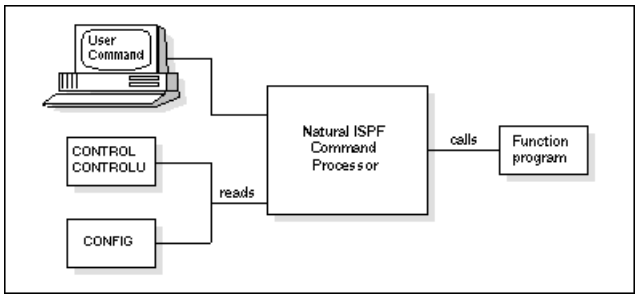



With this definition all reports created by Natural ISPF can be handled by a user routine in Entire Output Management. An example UEXSPF is delivered in the Example Library. For modification and execution it must be copied to the library SYSNOMU. For further details, see the Entire Output Management System *Programmer's* documentation, section *Printer Exits*, *User Separation Routines*, *Separator Pages*.

## NCP Concept - Member: NCPUSAGE

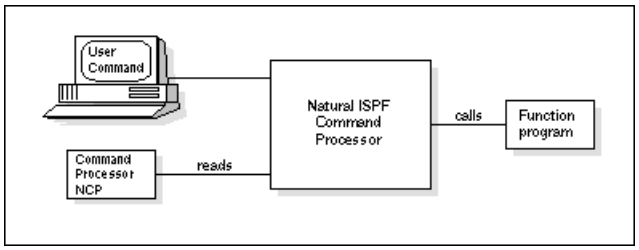
Usually Natural ISPF offers commands/functions and objects for several subsystems such as z/OS, z/VSE or BS2000. Depending on the installation, only a comparatively small subset of all functions is activated (with the subsystem definitions in the `CONFIG` member). The Natural ISPF command processor has to interpret the command entered on the screen by reading the whole `CONTROL` table, which contains commands, functions and objects and by tailoring this table with information from the `CONFIG` member.

The following figure illustrates Natural ISPF command processing:




All tables are stored in a database and are read during processing.

This overhead can be reduced by using a command processor (NCP) as illustrated below:




The command processor replaces `CONFIG` and the `CONTROL` member(s) and is stored in the Natural buffer pool improving performance of Natural ISPF command processing.

When using an NCP, which can be considered as a compiled object of the `CONTROL` and `CONFIG` tables, changes to the Natural ISPF table sources are not effective until the NCP is regenerated. For this reason the NCP should be used in stable environments, where changes to the Natural ISPF tables are infrequent.

The following changes in `CONFIG` affect the NCP:

- Activating/deactivating a subsystem
- Activating/deactivating object exits

In addition, any change to the Site Control Table CONTROLU affects the NCP. *

After installation of a new Natural ISPF release or maintenance level, the NCP must be regenerated to reflect changes in tables delivered by Software AG.

* For a detailed description of the Site Control Table, see the subsection *Defining a User Object* in section *Open NSPF* of the *Natural ISPF Programmer's Guide*.

The following topics are covered below:


- How to Generate an NCP Object for Natural ISPF
- Activating a Specific NCP Object
- Runtime Considerations When Using NCP
- Predefined Command Processors


## How to Generate an NCP Object for Natural ISPF

To generate an NCP (which is an option) enter the command GENNCP or select the NCP option from the Configuration Menu. The following window opens:

```
---------------------------- CONFIGURATION MENU -------------------------------
OPTION  ===> 4
                                                                Userid   BRY
           +-----Generation of NCP processor-----+              Time     14:03:46
   1    N-I !                                    !              Terminal DAEFTC45
           ! Enter processor name    :  A        !              Library  NSPFHELP
   2    CON !                                    !              Node     148
   3    CON ! Select one/more functions           !
           !  Generate and compile    :          !
   4    NCP !  Compile only            :          !
           !  Copy processor to SYSLIB:          !
           !  Generate report         :          !
           !                                     !
           +-------------------------------------+
```

| Field | Meaning |
|---|---|
| Enter processor name | Instead of overwriting an existing NCP, you can create a new one whenever necessary. Enter a 1-byte name (x) and the generated NCP is stored in the library SYSISPFU with the name IS-NCP-x. |
| Generate and compile | With the current contents of the Natural ISPF tables, an NCP source is generated and compiled to create an object in SYSISPFU. |
| Compile only | Mark this option if the generation of a previous execution was successful but the compilation failed. |
| Copy processor to SYSLIB | When a compiled NCP object exists in SYSISPFU, it must be copied to SYSLIB before it can be activated. |

| Field | Meaning |
|---|---|
| Generate report | Mark this option to perform a report, during generation, containing a log of functions. The report will then be written to the Natural ISPF workpool. |

### Activating a Specific NCP Object

A compiled NCP object that has been copied to the library SYSLIB and that follows the naming pattern IS-NCP-x can subsequently be activated by selecting the N-ISPF (parameters) option from the Configuration Menu and assigning the name suffix x to the field PROCESSOR ID.

### Runtime Considerations When Using NCP

If a NAT0888 occurs with NCP, increase the Natural parameter DATSIZE. If the defined NCP cannot be used by Natural ISPF (defined processor deleted or not accessible), an error message is displayed and Natural ISPF automatically invokes its standard command processing logic without NCP. A defined NCP can be deactivated by resetting its name in the NSPF Parameters screen.

### Predefined Command Processors

With Natural ISPF, the following command processors are loaded to SYSLIB and can be used if no user-defined objects and commands are defined and no user exits are to be activated.

| Name | Supported subsystems |
|---|---|
| IS-NCP-N | Natural, Incore database |
| IS-NCP-M | Natural, Incore database, z/OS |
| IS-NCP-P | Natural, Incore database, z/OS, CA Panvalet |
| IS-NCP-L | Natural, Incore database, z/OS, CA Librarian |
| IS-NCP-D | Natural, Incore database, z/VSE |
| IS-NCP-B | Natural, Incore database, BS2000 |
| IS-NCP-A | Natural, Incore database, z/OS, z/VSE, BS2000, CA Panvalet, CA Librarian |

## Defining Versioned Libraries

The parameters for storing previous versions of some edited Natural ISPF objects are described in the subsection ***Natural ISPF Parameters***. The versioning function is activated by the user using the VERSIONS ON setting in the user profile. For further information, see the following subsections in the *Natural ISPF User's Guide*:

- *Versioning* in section *Useful Features*;
- section *Profile Maintenance*.

Additionally, you can enforce or exclude versioning for specific Natural libraries, partitioned organized (PO) data sets and z/VSE libraries or sublibraries. Previous versions of members in these libraries are always stored according to the parameters set in the Natural ISPF system parameter table, irrespective of the `VERSIONS` setting in the user profile.

Versioning for specific libraries is enforced by specifying the library names in a member in the User Profile Library. Together with the library name, you can specify whether Natural ISPF will prompt the user to give a reason for changing a member when issuing the `SAVE` or `STOW` command.

This is done by adding the keyword `REASON` to the library name (see the examples below). The reason for change is then displayed when the user requests a lists of versions for the member (see the subsection *Versioning* in section *Useful Features* in the *Natural ISPF User's Guide*).

With added keyword `NOVERSION` you can exclude versioning for specific Natural libraries, partitioned organized (PO) data sets and z/VSE libraries or sublibraries.

### Natural Versioned Libraries

The names of all Natural libraries for which versioning is to be enforced or excluded at all times are maintained in the member `VERLSTN` in the User Profile Library. The `NAT VERS` option on the Administrator Menu gives direct write access to this member.

The following is an example of the member `VERLSTN`:

```
EDIT-CNF:SYSISPFU(VERLSTN)-Program->Struct-Free-45K ----------- Columns 001 072
   COMMAND===>                                                   SCROLL===> CSR
  ****** ***************************** top of data ****************************
  000010 NSPFvrs REASON
  000020 SYSISP2T NOVERSION
  000030 SYSISPFU
  000040 SYSISP2S
  ****** **************************** 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
```

Versioning is activated or excluded (NOVERSION) for all Natural libraries listed (one library per line). Note that when a user SAVEs or STOWs a member after modification in the library NSPF*vrs* (in the example map above, *vrs* is a placeholder for the current version, release and system maintenance level number), a prompt window allows the user to enter a reason for the change.

### PDS Versioned Libraries

The names of all data sets for which versioning is to be active or inactive (NOVERSION) at all times are maintained in member VERLSTP in the User Profile Library. The PDS VERS option on the Administrator Menu gives direct write access to this member.

The following is an example of the member VERLSTP:

```
EDIT-CNF:SYSISPFU(VERLSTP)-Program->Struct-Free-45K ---------- Columns 001 072
 COMMAND===>                                                   SCROLL===> CSR
****** ***************************** top of data ****************************
000010 EDITOR.SYSF.ZAPS
000020 EDITOR.COMN.SOURCE
000030 EDITOR.COMN.SYSTEM.SRCE REASON
****** **************************** 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
```

Versioning is activated or excluded (NOVERSION) for all data sets listed (one data set per line). Note that when a user SAVEs a member after modification in the library EDITOR.COMN.SYSTEM.SRCE, a prompt window allows the user to enter a reason for the change.

### z/VSE Versioned Libraries

The names of all libraries/sublibraries for which versioning is to be active or inactive (`NOVERSION`) at all times are maintained in member `VERLSTDP` in the User Profile Library. The `z/VSE VERS` option on the Administrator Menu gives direct write access to this member.

The following is an example of the member `VERLSTDP`:

```
EDIT-CNF:SYSISPFU(VERLSTDP) ---------------------------------- Columns 001 072
 COMMAND===>                                                   SCROLL===> CSR
****** ***************************** top of data ****************************
000010 NATPROC.IV131.SYSTEM.LIBRARY
000020 USRLIB.WHE NOVERSION
****** **************************** 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 library notation can be as follows:

- *library-name*

  Versioning is active or inactive for all members in all sublibraries in the library.

- *library.sublibrary*

  Versioning is active or inactive for all members in the specified sublibrary.

Versioning is defined for all libraries/sublibraries listed (one data set per line). For library/sublibrary `USRLIB.WHE`, the versioning is excluded.

## Maintaining Versioning Data

Available maintenance functions on versioning data are:

- List versioned members
- Delete versioned member

Access to versioning data is provided by the `VERSIONS` option on the Administrator Menu. If you select this option, the Versions Objects Entry Panel appears:

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



   Object type   ===>                          ( VSE,NAT,PDS,* )
   DSN / Library ===>
   Sub Library   ===>
   Member        ===>
   Member Type   ===>
   Node          ===>
   DBID          ===>
   FNR           ===>









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 your site does not have z/VSE, the corresponding fields are not displayed on the Entry Panel (Sub Library, Member Type).

You can enter a command in the command line and specify the versioned object you wish to maintain in the parameter fields.

Meaning of the parameter fields:

| Field | Meaning |
|---|---|
| Object type | NAT, PDS, or z/VSE. When using the LIST function, the asterisk wildcard (*) selects all types. The list is then sorted in the sequence PDS, z/VSE, NAT. |
| DSN / Library | Data set or library name of versioned object. When using the LIST function, you can enter a prefix followed by the asterisk wildcard (*). For example, enter NSPF* to list all library names starting with NSPF. |
| Sub Library | For z/VSE only: sublibrary of versioned object. When using the LIST function, you can enter a prefix followed by the asterisk wildcard (*). For example, enter TEST* to list all sublibrary names starting with TEST. |
| Member | Name of versioned member. When using the LIST function, you can enter a prefix followed by the asterisk wildcard (*). For example, enter NAT* to list all member names starting with NAT. |
| Member Type | For z/VSE only: when using the LIST function, specify user-defined type to restrict list to the group of members defined for the type. You can enter a prefix followed by the asterisk wildcard (*). For example, enter N* to list all types starting with N. |
| Node | Specify the Entire System Server node to be searched. If no node is specified, all nodes (from 1 to 255) are searched. |
| DBID / FNR | For NAT objects: when using the LIST function, you can restrict the search to a specific database ID, a file number, or both. If no DBID or FNR is specified, all are searched. |

The following topics are covered below:

- Function Commands
- Line Commands
- Local Commands

## Function Commands

The following maintenance function commands are available for versioned objects:

| Command | Parameter Syntax |
|---|---|
| LIST | *library*/*(*member*/*) TYPE=*t* NODE=*id* DBID=*db* FNR=*n* |
| DELETE | *library*/*(*member*/*) TYPE=*t* NODE=*id* DBID=*db* FNR=*n* |

**Note:** If you issue any of the above function commands from outside the Versions Maintenance facility, you must specify the object-type parameter MV before the object parameters.

The following topics are covered below:

- Example: LIST for TYPE=NAT
- Example: LIST for TYPE=PDS

- Example: LIST for TYPE=VSE

**Example: LIST for TYPE=NAT**

The following example is the result of the command:

```
LIST MV SYSISP*(*) TYPE=NAT
```

It shows all versioned members in all Natural libraries that start with `SYSISP`:

```
LIST-MV:NAT:SYSISP*(*)/DBID=*/FNR=* ------------ Row 0 of 45 - Columns 006 076
 COMMAND===>                                                  SCROLL===> CSR
   TYPE          LIBRARY(MEMBER)                                    NUM SIZE
** ****************************** top of list ***************************
   NAT           <9,80>SYSISPDB(IDBC---N)                           001 0001
   NAT           <9,80>SYSISPDB(IDBI---L)                           001 0001
   NAT           <9,80>SYSISPDB(IDBI---N)                           001 0001
   NAT           <9,80>SYSISPE(IDB-C22N)                            001 0001
   NAT           <9,80>SYSISPE(MAC-MVS4)                            001 0001
   NAT           <9,80>SYSISPFU(CONFIG)                             010 0010
   NAT           <9,80>SYSISPFU(ISP-LONU)                           001 0001
   NAT           <9,80>SYSISPFU(LIBDEF)                             003 0003
   NAT           <9,80>SYSISPFU(ULIBID)                             002 0002
   NAT           <9,80>SYSISPFU(VERLSTN)                            002 0002
   NAT           <9,80>SYSISPFU(VERLSTP)                            002 0002
   NAT           <9,80>SYSISPH1(BJOBMENU)                           001 0001
   NAT           <9,80>SYSISPH1(BS2FMENU)                           004 0008
   NAT           <9,80>SYSISPH1(BS2000)                             001 0001
   NAT           <9,80>SYSISPH1(DOCNAME)                            001 0001
   NAT           <9,80>SYSISPH1(EXAMPLE)                            011 0016
   NAT           <9,80>SYSISPH1(INDEX)                              001 0001
   NAT           <9,80>SYSISPH1(ISUO-9)                             001 0001
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 |
|---|---|
| `TYPE` | Object type. |
| `LIBRARY(MEMBER)` | For Natural members, this field shows DBID, FNR, library name and member name. |
| `NUM` | Number of versions of the member. |
| `SIZE` | Number of records in the versions file. |

**Example: LIST for TYPE=PDS**

The following example is the result of the command:

```
LIST MV MBE*(*) TYPE=PDS
```

It shows all versioned members in all PDS libraries that start with `MBE`. Note that if the `NODE` parameter is not specified, all nodes are searched:

```
LIST-MV:PDS:MBE.COMN.SOURCE(*)/NODE=* ------------ Row 0 of 6 - Columns 006 076
 COMMAND===>                                                 SCROLL===> CSR
   TYPE          LIBRARY(MEMBER)                                   NUM SIZE
** ******************************** top of list ********************************
   PDS           <148>MBE.COMN.SOURCE(ISPUPSUM)                    001 0016
   PDS           <148>MBE.COMN.SOURCE(PRINTER2)                    002 0002
   PDS           <148>MBE.COMN.SOURCE(SCRIPT)                      005 0005
   PDS           <148>MBE.COMN.SOURCE(TEST)                        002 0002
   PDS           <148>MBE.COMN.SOURCE(VERSIONS)                    002 0003
   PDS           <069>MBE.SYSE.SOURCE(VSE)                         020 0021
** ******************************* 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 |
|---|---|
| `TYPE` | Object type. |
| `LIBRARY(MEMBER)` | For PDS members, this field shows the node ID, library name and member name. |
| `NUM` | Number of versions of the member. |
| `SIZE` | Number of records in the versions file. |

**Example: LIST for TYPE=VSE**

The following example is the result of the command:

```
LIST MV NPR214.TEST1(*) TYPE=VSE
```

It shows all versioned members in the library `NATPROC.IV131.SYSTEM.LIBRARY` in sublibrary `TEST1` (note that in the command, the standard label library name is used):

```
LIST-MV:VSE:NATPROC.IV214.SYSTEM.LIBRARY*.TEST1(*. Row 0 of 1 - Columns 006 076
COMMAND===>                                                  SCROLL===> CSR
   TYPE          LIBRARY(MEMBER)                                   NUM SIZE
** ******************************** top of list ********************************
   VSE           <33>NATPROC.IV214.SYSTEM.LIBRARY.TEST1(UEDTB1.A)      003 0003
** ******************************* 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 |
|---|---|
| `TYPE` | Object type. |
| `LIBRARY(MEMBER)` | For z/VSE members, this field shows the DSN name, sublibrary name, member name and type. |
| `NUM` | Number of versions of the member. |
| `SIZE` | Number of records in the versions file. |

### Line Commands

You select a versioned member from a list by entering a line command in the input field preceding the member name and pressing ENTER. Each line command is an abbreviation of a function command:

| Line Command | Corresponding Function Command |
|---|---|
| D | DELETE |

### Local Commands

#### In List Mode

You can use the commands ALL, LAYOUT, RELIST and SORT. For detailed information, see the subsections in section *Useful Features* of the *Natural ISPF User's Guide*.

## Entire System Server Node Table

If you select the NODES option from the Administrator Menu, you can edit the table that is used to offer active help for the field NODE. This field that appears in several Natural ISPF screens is used for easy selection of a node.

**Note:** This table is required if you intend to use Mainframe Navigation functionality under Natural for Windows. In this environment, only nodes defined in this table can be accessed by Mainframe Navigation. For running Mainframe Navigation, it is important that the nodes are activated (see below). When the nodes are active, it is recommended to edit and save this table for getting the complete operating system information.

The following figure illustrates an example node table:

```
-----------------------------UPDATE-NODES-TABLE-------------------------------
COMMAND ===>

    Node Description           Name
    ----  --------------------  ----
    148  F Maschine__________  F-MC
    _68  Alsbach_____________  ALSA  NOT ACTIVE
    _69  E Maschine__________  E-MC
    _70  A Maschine__________  A-MC
    ___  ______________________
    ___  ______________________
    ___  ______________________
    ___  ______________________
    ___  ______________________
    ___  ______________________




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 |
|---|---|
| Node | Node number (Entire System Server DBID). |
| Description | Short description of node. |
| Name | Node name. |

If the node is not active, Entire System Server notifies you with a message. You can add, modify or delete node definitions by overtyping existing node definitions or entering data in empty input fields. To save the node table, issue the END command (usually assigned to PF3).

**Note:** User access to nodes is not restricted to the nodes that appear in this table. If you wish to restrict access to specific nodes, you can do so with user exits.

# 3 User Definitions


- Types of User Definition ..... 56
- User Group Definitions ..... 57
- Maintaining User Definitions ..... 59
- Authorization Table ..... 60
- Maintaining User Definitions with Function Commands ..... 65

A user definition consists of an authorization table in which you can authorize access to functions for classes of Natural ISPF objects, as well as of default settings on user profiles (PF key assignments, short names for libraries, magic characters, Editor profile, user defaults). All characteristics of the user profile are modifiable by the user. Authorization tables are modifiable only by authorized users.

## Types of User Definition

Natural ISPF allows for three different types of user definition:

- **Single users:**
  You can create a separate definition for a specific user ID;
- **User groups:**
  You can create a definition for a group of users. You can choose one of the following methods for associating user IDs with certain groups:
  - Prefix Method
  - Derivation from Natural Security

  These are explained in the subsection *User Group Definitions*.
- **Default definition (`user'*'`):**
  It is highly recommended to create a definition for the asterisk (*). This can be seen as a definition for a null prefix: users are assigned this definition if they log on with a user ID that is not specifically defined and for which there is neither a prefix definition nor a Natural Security group definition.

**Note:** Without a default definition (*), an undefined user ID to which no prefix definition applies is granted full authorization for the system.

In the case of the prefix method, a user is assigned the definition which most closely matches that user's ID. The following table illustrates how some example user IDs are assigned definitions:

| Definition | Assigned to user ID: |
|---|---|
| * | U1 |
| S* | S1 |
| SY* | SY1 |

You can modify the default user definition, and add and modify single user and group (prefix) definitions at any time.

You can enter the user definition facility in any of two ways:

- Select the USER option on the Administrator Menu to display the User Entry Panel. You can specify a function command in the command line and parameters in the input fields (see the following subsection);
- You can access user definitions from any Natural ISPF screen using function command syntax. See the subsection *Maintaining User Definitions with Function Commands*.

## User Group Definitions

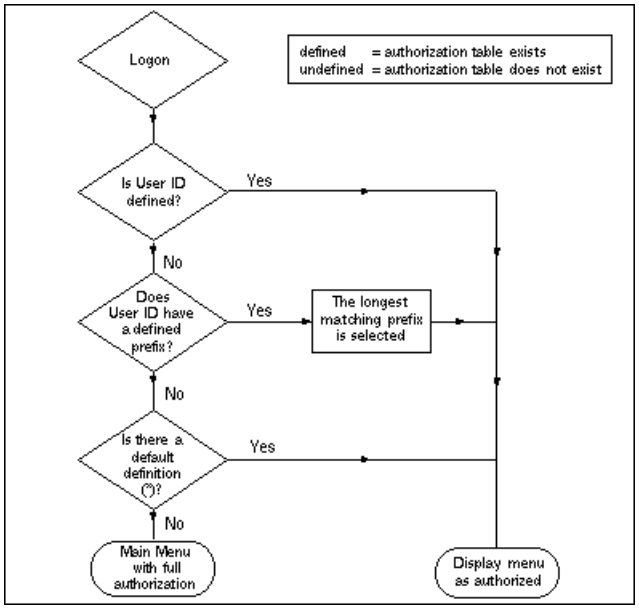
You can create a definition for a group of users. By setting APPLYMOD 101 to an appropriate value you can select one of the following methods for associating user IDs with certain groups:

- Prefix Method
- Derivation from Natural Security - NSC

### Prefix Method

In this case, a definition for a prefix applies for all user IDs matching that prefix, except for those users for which the corresponding profile item has been defined specifically. For example, the definition for the ID SAG* applies to all user IDs that start with SAG and have no unique definition.

The following flow diagram illustrates the internal handling for the prefix method when a user logs on:




**Note:** Without a default definition (*), an undefined user ID to which no prefix definition applies is granted full authorization for the system. When installing Natural ISPF and setting

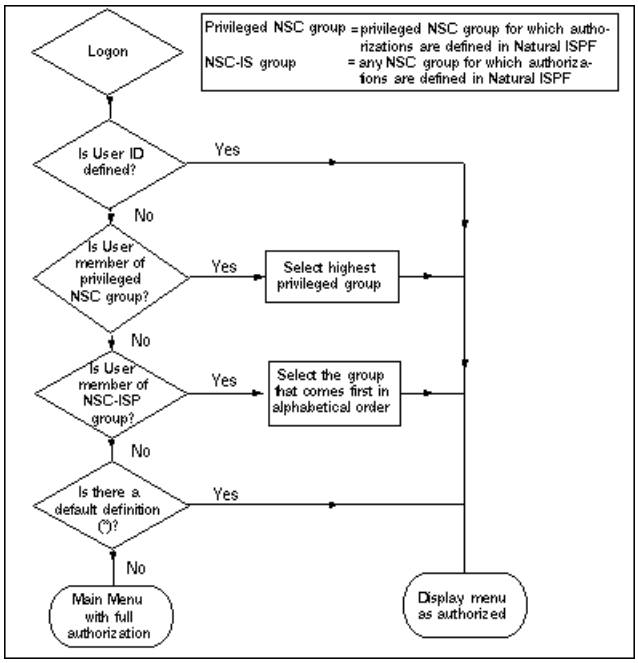
up the system, you must therefore define at least a default definition (*) to control access to the system (see the subsection *Maintaining User Definitions*).

### Derivation from Natural Security - NSC

In this case, a definition made for an ID that has been defined as a user group in Natural Security will be used as a default definition that applies for all members of that group, except for those users for which the corresponding profile item has been explicitly defined.

If a user is a member of several groups, Natural ISPF will first search privileged groups in the specified order and then non-privileged groups in alphabetical order.

The following flow diagram illustrates the internal handling for the Natural Security (NSC) method when a user logs on:




**Note:** If a user is a member of more than 20 groups, only the first 20 will be evaluated in the above context.

## Maintaining User Definitions

If you select the USER option from the Administrator Menu, the User Entry Panel appears:

```
---------------------------- USER - ENTRY PANEL -------------------------------
COMMAND ===>


   User          ===> *
   Profile type  ===>                          ( A,K,L,C,E,D,B,Y,N,O)









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 | |
|---|---|---|
| User | User ID of user definition to be maintained. You can also enter the asterisk wildcard (*) to list all user definitions, or ABC* to list all definitions beginning with ABC. | |
| Profile type | Enter characteristic of definition to be maintained. Possible options: | |
| | A | Authorization table |
| | B | BS2000 defaults |
| | C | Magic characters |
| | D | User defaults |
| | E | Editor profile |
| | K | PF key assignments |
| | L | Library short names |
| | N | Natural defaults |

| Field | Meaning | |
|---|---|---|
| | O | Editor color definition |
| | Y | Layout of object lists (COPY or DELETE only) |
| | * | Full profile (COPY or DELETE only) |

**Note:** The Profile type field is not used as selection criterion for the LIST command. It is used to select sections of the user profile for EDIT, DELETE or COPY operations. The whole profile can be selected for COPY and DELETE operations by entering the asterisk wildcard (*) in this field.

The user authorization table (characteristic A) can only be modified by users authorized to access configuration functions. All other characteristics are modifiable by the user and are described in detail in the section *Profile Maintenance* in the *Natural ISPF User's Guide*. You can access them here to maintain the default settings.

Once you have entered the specified user definition, you can scroll the profile sections using the UP and DOWN commands (usually assigned to PF7 and PF8 respectively).

## Authorization Table

To access the user authorization table for a user profile, specify the profile name (user ID, group ID, prefix followed by the wildcard *, or wildcard * only) in the User field and A in the Profile type field. The authorization table for the specified definition appears, for example:

```
--------------------- EDIT USER BRY , Byrone, Rinaldi -----------------------
 COMMAND ==>


    Authorization Class           Level       Main Menu      ===>
      Natural  programming     ==> 9
      PDS Maintenance          ==> 9
      Data Sets Maintenance    ==> 9       + --- COMMANDS LEVEL REMINDER ---- +
      SYSOUTS                  ==> 9       ! Lvl    Command Abbreviation      !
      System info              ==> 9       ! ---    --------------------      !
      Active jobs              ==> 9       ! 1 - L,B,ZP,XT,I,ET,DI,DF,RU,XE   !
      Operator commands        ==> 9       !     EX,OT,FR,DW,CR,BPSTAT         !
      NSPF Administrator       ==> 9       ! 2 - E,R,SB,PL,PR,CP,A,CT,U,FL     !
      PANVALET                 ==> 9       !     ST,CC,RL,HL,DS,UP             !
      LIBRARIAN                ==> 9       ! 3 - D,PG,CH,NSPR,GENN             !
      USER defined             ==> 9       ! 4 - CM,OPER                       !
                                           !                                   !
                                           !                                   !
                                           !                                   !
                                           +-----------------------------------+


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 above authorization table could be in place for user definitions in a z/OS environment that includes CA Panvalet.

**Explanation of the authorization table:**

- **Header**
  The header line contains the function (`EDIT USER`) and the user ID invoked.
- **Main Menu**
  The field labelled `Main Menu` contains the name of the menu displayed when the user logs on to Natural ISPF. The default menu is the Main Menu, `MAIN` (see the section *Menu Maintenance*).
- **Authorization class**
  The column headed `Authorization Class` contains a list of items that correspond to Natural ISPF objects and certain administration functions. The classes displayed correspond to the subsystem(s) installed at your site. For a list of possible classes, see *Authorization Classes* at the end of this documentation.

  The extent to which the user is authorized for each class of objects is determined by the authorization level.

- **Authorization level**
  The column headed `Level` contains the numerical identifier of the level to which the user is authorized for the corresponding class of objects. An authorization level is a command or group of commands defined in the window headed `Commands Level Reminder`. Typing a level number against a class of objects authorizes the user to issue these commands for the class of objects. The lowest possible level is blank or 0 (zero) and means that the corresponding object option does not appear on the user's Main Menu. The highest possible level is 9 and includes all commands on Levels 1-9.
- **Command Level Reminder**
  This window tells you which commands belong to which level. The abbreviations correspond to the valid abbreviations of the respective commands as follows:

| Level 1 Abbreviation | Function |
|---|---|
| L | LIST |
| B | BROWSE |
| ZP | ZAPS |
| XT | EXTERNS |
| I | INFORMATION |
| ET | EXTENTS |
| DI | DIFFERENCE |
| DF | DEFINITION |
| RU | RUN |
| XE | EXECUTE |
| EX | EXPORT |
| OT | OUTPUT |
| FR | FORMAT |
| DW | DOWNLOAD |
| CR | COMPARE |
| BPSTAT | BPSTAT |
| DEFB | DEFBS2PROF (BS2000 general defaults) |
| DEFS | DEFSUBPROF (BS2000 submit defaults) |

| Level 2 Abbreviation | Function |
|---|---|
| E | EDIT |
| R | RENAME |
| SB | SUBMIT |
| PL | PLAY |
| PR | PRINT |
| CP | COPY |

| Level 2 Abbreviation | Function |
|---|---|
| A | ALLOCATE |
| CT | CATALOG |
| U | UNCATALOG |
| FL | FOLLOW |
| ST | STATUS |
| CC | Condition codes |
| RL | RELEASE |
| HL | HOLD |
| DS | DESCRIPTION |
| UP | UPLOAD |

| Level 3 Abbreviation | Function |
|---|---|
| D | DELETE |
| PG | PURGE |
| CH | CHANGE |
| NSPR | Natural ISPF parameters |
| GENN | Generate command processor |

| Level 4 Abbreviation | Function |
|---|---|
| CM | COMPRESS |
| OPER | Issue operator commands |

You can update an authorization by modifying the Main Menu name and/or modifying the authorization level for one or more classes.

For example, if you type `0` in the authorization level field for the Natural class, the user cannot access Natural objects; this option will not appear on his Main Menu when he logs on, and he cannot use direct commands for Natural objects.

If you type `1` in the authorization level field for the SYSOUT class, the user can perform browse functions on job SYSOUTs, but he cannot perform any other operations. Whether the `JOBS` option appears on that user's Main Menu depends on the system authorization level for the option (see the section ***Menu Maintenance***).

If an option does not appear on the user's Main Menu but the user is authorized for some functions on the object type, he or she can use appropriate direct commands.

To save user authorizations, issue the `END` command (usually assigned to `PF3` ) after having modified any value on the screen.

Below is an example of a default authorization table (*):

```
 -------------------------------- EDIT USER * ----------------------------------
  COMMAND ==>


     Authorization Class            Level       Main Menu      ===> NULL
       Natural  programming     ==>
       PDS Maintenance          ==>
       Data Sets Maintenance    ==>          + --- COMMANDS LEVEL REMINDER ---- +
       SYSOUTS                  ==>          ! Lvl    Command Abbreviation      !
       System info              ==>          ! ---    --------------------      !
       Active jobs              ==>          ! 1 - L,B,ZP,XT,I,ET,DI,DF,RU,XE   !
       Operator commands        ==>          !     EX,OT,FR,DW,CR,BPSTAT        !
       NSPF Administrator       ==>          ! 2 - E,R,SB,PL,PR,CP,A,CT,U,FL    !
       PANVALET                 ==>          !     ST,CC,RL,HL,DS,UP            !
       LIBRARIAN                ==>          ! 3 - D,PG,CH,NSPR,GENN            !
       USER defined             ==>          ! 4 - CM,OPER                      !
                                             !                                  !
                                             !                                  !
                                             !                                  !
                                             +----------------------------------+


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
```

Explanation: Since all authorization levels are blank (zeroes), undefined users who do not belong to a prefix group cannot execute any secured function, and they will be presented with menu `NULL` when they log on to Natural ISPF (see the section *Menu Maintenance* for an example of menu `NULL`).

**Notes:**

1. Functions on the user's profile, the user workpool and recovery files are not secured.
2. If you leave the above screen by issuing an `END` command or by pressing PF3 without having modified any of the values on the screen, the authorization table of the individual user specified will not be updated. From this scenario, you cannot even be sure if the authorization table displayed has been defined for the user explicitly or if it has been inherited from a prefix definition or from a (Natural Security-based) user group definition. To be certain that a user has an individual authorization table, look at the list of Natural ISPF users: all users with an individual authorization table will be listed with `Auth` next to user ID and last access date.

## Maintaining User Definitions with Function Commands

Natural ISPF users are separate objects within Natural ISPF with object type USR. This means that you (and other authorized users) can maintain user definitions with function command syntax entered from any system screen.

The available function commands are:

| Command | Object Parameter Syntax |
|---|---|
| LIST | *user-id* |
| EDIT | *user-id* TYPE=*t* |
| DELETE | *user-id* TYPE=*t* |
| COPY | *user-id* TYPE=*t*,*target-user-id*,REP |

| Parameter | Function | |
|---|---|---|
| *user-id* | Can be a specific user ID, a prefix notation or the default definition (*). | |
| *t* | Identifies the user characteristic to be maintained. Possible options: | |
| | A | User authorization table |
| | B | BS2000 defaults |
| | C | User magic characters |
| | D | User defaults |
| | E | User Editor profile |
| | K | User PF-key assignments |
| | L | User short names for libraries |
| | N | Natural defaults |
| | O | Editor color definition |
| | Y | Layout of object lists (DELETE and COPY only) |
| | * | Whole profile (DELETE and COPY only) |
| *target-user-id* | New user definition to be created or replaced. | |
| REP | Specify to replace target definition, if it already exists. | |

**Notes:**

1. If you issue any of the commands without parameters, Natural ISPF prompts you for valid values.
2. If you issue any of the above commands from outside the user maintenance facility, you must specify the object-type parameter USR after the command keyword.

The following examples are provided below:

- Example: EDIT
- Example: DELETE
- Example: COPY
- Example: LIST

### Example: EDIT

The command:

```
EDIT USR SAG* TYPE=K
```

displays the PF key table assigned to all users with prefix SAG. You can modify this table. The update is performed every time you press the ENTER key, provided the screen contains valid update data. You can leave the screen with the command END (usually assigned to PF3).

### Example: DELETE

The command:

```
DELETE USR *
```

deletes the default definition (*). Note that without a default definition, any undefined user for whom there is no prefix definition receives full authorization at logon.

### Example: COPY

The command:

```
COPY USR MBE TYPE=Y
```

can be used to copy the layout definition of object lists from one user to another. The following window opens:

```
+-------------------------------------------------+
!                                                 !
! Copy User MBE        Section: LAYOUT            !
! to   User                                       !
! Replace   NO                                    !
! Enter to perform , PF3 to exit                  !
+-------------------------------------------------+
```

Enter the user ID of the recipient user in the to User input field to copy the layout definitions from user MBE. All list layouts defined by user MBE are copied. For details on list layout, see the section *LAYOUT Command for Lists* in the section *Useful Features* of the *Natural ISPF User's Guide*.

### Example: LIST

The command:

```
LIST USR  *
```

lists all Natural ISPF users, for example:

```
 LIST-USR:* ------------------------------------ Row 0 of 15 - Columns 010 076
  COMMAND===>                                                 SCROLL===> CSR
    USER              DATE     DEFINED CHARACTERISTICS
 ** ******************************** top of list ******************************
    *        *Edited            Auth,Edit,Default,Char,
    BRY                94/12/13 Auth,Edit,Key,Natural,
    GW                 94/12/08
    HHH                         Edit,
    JWO                94/12/13 Auth,Edit,Default,Char,Key,Color,Natural,Lib,
    JWOAB                       Default,
    MAK                94/11/12 Default,Key,
    MSE                94/12/09
    MZC                94/12/13 Auth,Edit,Default,Char,Key,Layout,Natural,Lib,
    MZCC               94/10/27 Auth,Edit,Default,Char,Key,Lib,
    SML                94/12/09
    UHE                94/10/11 Auth,
    WHE                94/10/17
    WKK                94/12/01
    WOS                94/11/23
 ** ****************************** 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 contains all users who have logged on Natural ISPF, as well as all defined user definitions (authorization tables and profile sections).

Meaning of the column headings:

| Column | Meaning | |
|---|---|---|
| USER | User ID, user prefix or *. Each user appears in the list after first logon | |
| DATE | Date the user logged on to Natural ISPF last. | |
| DEFINED CHARACTERISTICS | Characteristics which are specifically defined. Possible values: | |
| | Auth | User authorization table |
| | BS2 | User BS2000 defaults (general and submit) |
| | Char | User magic characters |
| | Color | User color definitions |

| Column | Meaning | |
|---|---|---|
| | Default | User system defaults |
| | Edit | User Editor profile |
| | Key | User PF-key table |
| | Layout | User layout for lists |
| | Lib | User library short names |
| | Natural | User Natural defaults |
| | For any characteristic not in the list, the user is assigned the values defined for the appropriate prefix definition or, if no prefix matches the user ID, the values defined for the default definition (*). | |

A user characteristic attains *defined* status when you create or modify it for the user ID, or when a user modifies any characteristic in his or her user profile while working with Natural ISPF.

You can select any user or definition from the list with the E (EDIT), CP (COPY) or D (DELETE) line command entered in the input field preceding the user ID. The EDIT option allows you to modify the user's command authorization table as described above, and any profile setting as described in the section *Profile Maintenance* in the *Natural ISPF User's Guide*.

# 4 Menu Maintenance


- Starting a Menu Edit Session ..... 70
- Syntax of Menu Definition ..... 70
- Customizing Menus ..... 74
- Maintaining Menus with Function Commands ..... 86

Menu maintenance means customizing system menus to the requirements of your installation. You can add menus, and modify or delete existing ones.

**Note:** As of Natural ISPF Version 2.4.1, the menu structure has changed. If you wish to use the former menu structure, define the menu `MAIO` as main menu and/or set the parameter `APPLYMOD 104` to `X`. See the subsections *Authorization Table* and *APPLYMOD 104*.

## Starting a Menu Edit Session

Natural ISPF offers three options for starting a menu edit session:

- The `MENU` option on the Administration Menu prompts you for a menu name; enter a menu name to start an edit session with the menu definition (see the subsection *Syntax of Menu Definition*);
- The `MENU LIST` on the Administration Menu prompts you for a prefix to list all menus with that prefix (enter the asterisk wildcard (*) in this window and press ENTER to list all menus); see the example list in the subsection *Maintaining Menus with Function Commands*.
- You can also start an edit session using function commands entered from any system screen, addressing object-type `MNU`. Available function commands are `LIST`, `EDIT`, and `DELETE`. See the subsection *Maintaining Menus with Function Commands*.

## Syntax of Menu Definition

Below is a typical Natural ISPF Main Menu:

```
--------------------- NATURAL / NATURAL ISPF Main Menu ------------------------
OPTION  ===>
                                                                 User ID  FHI
                                                                 Time     15:47:27
 _ 1    NATURAL    - Development Functions                       Terminal DAEFTCS3
 _ 2    NATURAL    - Development Environment Settings            Library  NSPF241
 _ 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-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 :
```

If you select the `MENU` option on the Administrator Menu and enter the menu name `MAIN` in the prompt window, the associated menu definition appears in Editor format in `EDIT` mode:

```
EDIT-MNU:MAIN ------------------------------------------------ Columns 001 072
 COMMAND===>                                                   SCROLL===> CSR
****** ***************************** top of data ****************************
000010 HDR =NATURAL ISPF MAIN MENU
=cols> ----+----1----+----2----+----3----+----4----+----5----+----6----+----7--
000020 HELP=MAIN
000030 LINE=0    MENU PROF       PROFILE    - Profile maintenance
000040 LINE=#    ENTRY NAT   NN1 NATURAL    - Work with NATURAL objects
000050 LINE=#    ENTRY VIW   NN1 VIEWS      - Definition, content of NATURAL Vi
000060 LINE=#    ENTRY ERR   NN1 ERROR      - Work with NATURAL error messages
000070 LINE=#    ENTRY PRD   NN1 PREDICT    - Work with PREDICT descriptions
000080 LINE=#    ENTRY OUT       WORKPOOL   - Listing user output
000090 LINE=#    ENTRY CTN   0   CONTAINER  - Incore database container file
000100 ****=
000110 LINE=#    ENTRY JOB   MJ1 JOBS       - Display JOBS status and data (MVS
000120 LINE=#    ENTRY PDS   MP1 PDS        - Work with PDS members
000130 LINE=#    ENTRY DS    MD1 DATA SETS  - Maintain data sets
------ ---------------------------------------------- 13 line(s) not displayed
000270 LINE=#    MENU SAT    S   SAT        - System Automation Tools
000280 LINE=NEWS NEWS            CHANGES    - New features in NATURAL ISPF
000290 LINE=END                  EXIT       - Exit NATURAL ISPF
000300 LINE=HELP                 HELP       - Display help information
000310 CMD =X            END
000320 CMD =x            END
```

The type of line in the menu is specified by the label starting in Column 1 of each line. The fields in each line must be coded in a specific range of columns (see the columns line in the above figure).

The following table explains the lines in the menu:

| Line | Meaning |
|---|---|
| HDR=*text* (A50) | Defines the menu header to be displayed at the top of the screen. |
| HELP=*member* (A8) | Name of the member containing the help text for the screen. Help texts supplied by Software AG are stored in the System Help Library. You can a define site-specific help texts and store them in the User Help Library. When a user issues the HELP command, the User Help Library is always searched first for a specific help member, then the Software AG library is searched. See also the section ***Site-Specific Online Information***. |
| LINE=# ENTRY NAT NN1 Natural - Work with Natural objects | This line is an example of an option to appear on the menu. The detailed syntax is explained in the table on the following page. |
| ****=*text* | Defines a text line to be displayed on the menu. No text specifies a blank line. |
| CMD=*synonym command* | Where:<br><br>*synonym* is a command synonym (A11).<br><br>*command* can be any Natural ISPF command (A50). |

| Line | Meaning |
|---|---|
| | This line defines a command synonym; this can be an abbreviation of a Natural ISPF command which the user enters in the command line, or the synonym can be used in a menu option line if the available 12 bytes are too short for the original command. The above example menu defines `X` in upper case and lower case as a synonym for the `END` command. |

The number of displayable menu options is limited to 18. This means that the sum of `LINE=` and `****=` lines to be displayed must not exceed 18.

### Detailed Syntax of a Menu Option

```
LINE=# ENTRY NAT NN1 Natural - Work with Natural objects
```

The items must be written in the appropriate column number as shown below:

| Parameter (Format) | Meaning |
|---|---|
| *opt* (A4) (# in example above) (Columns 6 to 9) | Automatic numeration of the option. The special character # assigns the next number following on from the previous option. The first option line can contain a real number (in the above example, the first option is `0` (zero). The default is `1`. Alternatively, you can specify a character string for the option. |
| *command* (A12)(`ENTRY NAT` in example) (Columns 11 to 22) | Valid Natural ISPF command or command synonym. This command is executed when the user selects this option. When defining your own menus or modifying existing ones, the following commands are available:<br>`MENU` *mymenu* — Displays menu `MYMENU`.<br>`NAT` *myprog* — Executes Natural program `MYPROG`.<br>*synonym* — Executes the command assigned to the synonym (see the description of the `CMD` line on the previous page).<br>*Natural-ISPF-command* — Executes the specified Natural ISPF command. |
| *S* (A1) (first `N` in example) (Column 23) | Signifies the subsystem. For example, `N` for Natural. For a list of possible options, see ***Subsystems Supported by Natural ISPF*** at the end of this documentation. The menu line is only displayed if the specified subsystem is activated in the `CONFIG` member. |
| *C* (A1) (second N in example) (Column 24) | Denotes the authorization class as listed in the user authorization table, for example, `N` for Natural. If the menu option is site-defined, use the equal sign (=) here. The authorization codes are also used in the site control table (see the section *Open NSPF* in the *Natural ISPF Programmer's Guide*). For a list of possible values, see ***Authorization Classes*** at the end of this documentation. |
| *L* (N1) (1 in example) (Column 25) | Denotes the authorization level for the class. Possible options are `1-9`. The option line will appear on a user's menu only if the level specified here is equal to or lower than the authorization level for the same class in the user's authorization table. The above example specifies Level 1 for class Natural. |

| Parameter (Format) | Meaning |
|---|---|
| | To bar a user from using the Natural option, the user must have Level 0 in his or her authorization table. If you specify `2` here and the user is authorized for Natural with Level 1, he or she will not see the option on his or her menu, but can execute Level 1 functions using function commands on the corresponding object types. See also the subsection ***Authorization Table*** in the section *User Definitions*. |
| *descriptive-text* (A50) (`Natural -Work with Natural objects` in example) (Columns 27 to 76) | Description of the menu option to appear in the menu line. |

## Customizing Menus

Use the Editor to customize menus. To create a new menu from scratch, or adapt an existing menu for your site's purposes, you are strongly advised to use an existing menu as a model for the new one.

The easiest way of doing this is to work in multi-session mode: start an edit session with a new menu name in one session and an Editor session with an existing menu in another. Use cross-section copying with two Editor line commands `CC` to mark the block of the existing menu to be copied, and an Editor line command A in the edit session with the new menu to mark the place where the menu lines are to be inserted (see also the subsection *Multi-Session Operations* in the section *Useful Features* of the *Natural ISPF User's Guide*).

This subsection provides some typical examples of customized menus:

- Example 1: Menu to Extend Menu Structure
- Example 2: Menu to Invoke Natural Programs/Applications
- Example 3: Menu to Submit Batch Jobs
- Example 4: Menu to Execute Commands
- Example 5: Menu to Provide Example Menu Options

- Example 6: Menu NULL for Unauthorized Users


### Example 1: Menu to Extend Menu Structure

The following menu named `SAG` invokes other menus, using the `MENU` command:

```
------------------------------ Software AG MENU --------------------------------
OPTION  ===>
                                                                 Userid   MBE
                                                                 Time     17:52:12
   1    ISPF       - New features and functions                  Terminal DAEDC617
   2    ISPF       - Error messages and comments                 Library  MBE
   3    BATCH      - Batch utilities menu                        Node
   4    NATURAL    - NATURAL utilities menu

   5    JCL        - for NATURAL ISPF
   S    SDSF       - System display/search facility









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 following is the corresponding menu definition:

```
EDIT-MNU:SAG ------------------------------------------------ Columns 001 072
 COMMAND===>                                                  SCROLL===> CSR
****** ***************************** top of data ****************************
000010 HDR =Software AG MENU
000020 HELP=SAGHELP
=cols> ----+----1----+----2----+----3----+----4----+----5----+----6----+----7--
000030 LINE=#    NEWS            ISPF       - New features and functions
000040 LINE=#    ERRM         A4 ISPF       - Error messages and comments
000050 LINE=#    MENU BTCH    =4 BATCH      - Batch utilities menu
000060 LINE=#    MENU UTIL       NATURAL    - NATURAL utilities menu
000070 ****=
000080 LINE=#    MENU ISPF       JCL        - for NATURAL ISPF
000090 LINE=S    MENU SDSF       SDSF       - System display/search facility
000100 ****=
000110 CMD =ERRM list pds nn(v2*)
****** ************************** 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
```

**Explanation**

The line:

```
HELP=SAGHELP
```

invokes help member `SAGHELP` in the User Help Library when a user issues the `HELP` command from this menu.

The line:

```
LINE=S    MENU SDSF       SDSF       - System display/search facility
```

defines an option `S`, which, when selected, displays the menu named `SDSF`. Since no subsystem and authorization codes are specified in this line, the option is subsystem-independent and will appear on this menu for all users. Only access to the options invoked by the `ERRM` and `MENU BTCH` commands are subject to authorization.

The command `ERRM` is a synonym for a command sequence, defined by the line:

```
CMD =ERRM list pds nn(v2*)
```

If you wish to make this menu available from any menu at your site, you must add the following line to the options on that menu definition:

```
LINE=#    MENU SAG        SAG        - Software AG menu
```

### Example 2: Menu to Invoke Natural Programs/Applications

The following menu invokes Natural programs or applications:

```
  ------------------------ Natural ISPF Utility Menu ---------------------------
  OPTION  ===>
                                                                 Userid   BRY
                                                                 Time     10:48:43
     TP   SYSTP      - NATURAL TP monitor specifics              Terminal DAEFTC10
     PROD SYSPROD    - Installed products                        Library  BRY
     FIL  SYSPROF    - Files information                         Node
     BPM  SYSBPM     - NATURAL Buffer pool
     LOG  DBLOG      - Trace Database calls
     DBA  SYSDBA     - DBA utilities
     CAT  CATALL     - Catalog NATURAL Library
     DIC  PREDICT    - PREDICT Dictionary
     XREF XREF       - Active References

     SEC  SYSSEC     - NATURAL Security
     ERR  SYSERR     - NATURAL Error messages
     CP   SYSMAIN    - NATURAL Copy utility




  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 menu is created by the following definition:

```
EDIT-MNU:UTIL ------------------------------------------------ Columns 001 072
 COMMAND===>                                                   SCROLL===> CSR
****** ***************************** top of data ****************************
000010 HDR =NATURAL ISPF Utility Menu
=cols> ----+----1----+----2----+----3----+----4----+----5----+----6----+----7--
000020 LINE=TP   SYSTP        N1 SYSTP      - NATURAL TP monitor specifics
000030 LINE=PROD NAT SYSPROD  N1 SYSPROD    - Installed products
000040 LINE=FIL  NAT SYSPROF  N1 SYSPROF    - Files information
000050 LINE=BPM  APP SYSBPM   N1 SYSBPM     - NATURAL Buffer pool
000060 LINE=LOG  TDBLOG       N1 DBLOG      - Trace Database calls
000070 LINE=DBA  SYSDBA       N1 SYSDBA     - DBA utilities
000080 LINE=CAT  NAT CATALL   N1 CATALL     - Catalog NATURAL Library
000090 LINE=DIC  APP SYSDIC   N1 PREDICT    - PREDICT Dictionary
000100 LINE=XREF NAT L X      N1 XREF       - Active References
000110 ****=
000120 LINE=SEC  SYSSEC       N3 SYSSEC     - NATURAL Security
000130 LINE=ERR  SYSERR       N1 SYSERR     - NATURAL Error messages
000140 LINE=CP   APP SYSMAIN  N1 SYSMAIN    - NATURAL Copy utility
000150 ****=
000160 CMD =SYSTP   APP SYSTP  MENU
000170 CMD =SYSDBA  APP SYSDBA MENU
000180 CMD =TDBLOG  NAT TEST DBLOG ?
000190 CMD =SYSERR  APP SYSERR MENU
000200 CMD =SYSSEC  APP SYSSEC MENU
```

**Explanation**

The line:

```
LINE=XREF   NAT L X      N1 XREF       - Active References
```

invokes Natural system command `LIST` with parameter `XREF`. This option is subsystem-independent with Authorization Level 1.

If you wish to add a site-specific option that invokes a different application, you must add a menu line similar to the following:

```
LINE=#    APP MYAPPL   =3 MYAPPL     - Natural my application
```

This example assumes that the application `MYAPPL` is terminated with the command `RETURN` on the stack.

This menu option adds a subsystem-independent option `MYAPPL` to the menu. The authorization class is user-defined (=) with Level 3, which means that a user must be authorized with at least Level 3 in his authorization table to see the option on his or her menu.

The line:

```
CMD =SYSERR APP SYSERR MENU
```

defines `SYSERR` as a synonym for the command sequence `APPLICATION SYSERR MENU` and is used in the `SYSERR` menu option to invoke the Natural error message maintenance facility.

If you wish to make the above menu available at your site, you must add the following line to the options on that menu definition:

```
LINE=#    MENU UTIL       UTIL       - Natural ISPF Utilities
```

### Example 3: Menu to Submit Batch Jobs

The selection of an option on the following menu results in the submission of a Batch job:

```
--------------------------- Batch Utilities Menu -------------------------------
OPTION  ===>
                                                                 Userid   MBE
                                                                 Time     17:52:12
   1    IEBCOPY    - Online input for IEBCOPY                    Terminal DAEDC617
   2    ADAREP     - ADABAS Database Report                      Library  MBE
   3    TAPE       - Tape creation                               Node
   4    NATBAT     - NATURAL Batch execution









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 following is the menu definition:

```
EDIT-MNU:BTCH ------------------------------------------------ Columns 001 072
 COMMAND===>                                                   SCROLL===> CSR
****** ***************************** top of data ****************************
000010 HDR =Batch Utilities Menu
000020 HELP=MAIN
=cols> ----+----1----+----2----+----3----+----4----+----5----+----6----+----7--
000030 LINE=#    IEBCOPY         IEBCOPY    - Online input for IEBCOPY
000040 LINE=#    ADAREP          ADAREP     - ADABAS Database Report
000050 LINE=#    TAPE            TAPE       - Tape creation
000060 LINE=#    NATBAT          NATBAT     - NATURAL Batch execution
000070 ****=
000080 CMD =IEBCOPY SUBMIT PDS JWO.COMN.SOURCE(EXSUBMIT)
000090 CMD =ADAREP SUBMIT PDS JWO.COMN.SOURCE(EXADAREP)
000100 CMD =TAPE SUBMIT MACRO EXMAC2
000110 CMD =TAPE2 SUBMIT NAT NSPF1200(EXTAPENA)
000120 CMD =NATBAT SUBMIT PDS JWO.COMN.SOURCE(EXNATBAT)
****** **************************** 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
```

**Explanation**

All options have no subsystem or authorization restrictions.

Please note the use of command synonyms in this example. For example, the line:

```
CMD =ADAREP SUBMIT PDS JWO.COMN.SOURCE(EXADAREP)
```

defines `ADAREP` as synonym for the command string:

```
SUBMIT PDS JWO.COMN.SOURCE(EXADAREP)
```

Selecting the `ADAREP` option on the menu submits JCL member `EXADAREP` in the PDS library `JWO.COMN.SOURCE. JCL` can also be contained in a Natural member (see the `TAPE2` synonym).

The members submitted in this way can also make use of the Natural ISPF macro facility as described in section *Macro Facility* in the *Natural ISPF Programmer's Guide*. For example, submission of JCL could result in a window being displayed in which you enter variable parameters before the job is submitted. Alternatively, for faster processing, you can submit compiled macros by submitting a macro object (see the `TAPE` option and synonym).

If you wish to make the above menu available from any menu at your site, you must add the following line to the options on that menu definition:

```
LINE=#    MENU BTCH       BATCH      - Batch Utilities
```

### Example 4: Menu to Execute Commands

For repetitive use of similar commands with different parameters, it may be useful to define a menu with options that reflect certain command sequences. This is done using the synonym feature. For example, the following menu provides a number of system monitoring functions (it may look familiar to you):

```
---------------------- 'like SDSF' primary option menu -----------------------
   OPTION  ===>
                                                                Userid   MBE
                                                                Time     18:04:16
    LOG  - Display the system log                               Terminal DAEDC617
    DA   - Display active users of the system                   Library  MBE
    I    - Display jobs in JES input queue                      Node
    O    - Display jobs in JES output queue
    H    - Display jobs in JES held output queue

    ST   - Display status of jobs in JES queues
    INIT - Display JES initiators on this system

    END  - Exit 'like SDSF'
    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 Curso
```

This menu is created by the following menu definition:

```
EDIT-MNU:SDSF ------------------------------------------------ Columns 001 072
 COMMAND===>                                                   SCROLL===> CSR
****** ***************************** top of data ****************************
000010 HDR ='like SDSF' primary option menu
000020 HELP=MAIN
=cols> ----+----1----+----2----+----3----+----4----+----5----+----6----+----7--
000030 LINE=LOG  LOG             - Display the system log
000040 LINE=DA   UQ              - Display active users of the system
000050 LINE=I    QQ              - Display jobs in JES input queue
000060 LINE=O    QO              - Display jobs in JES output queue
000070 LINE=H    QH              - Display jobs in JES held output queue
000080 ****=
000090 LINE=ST   QS              - Display status of jobs in JES queues
000100 LINE=INIT QX              - Display JES initiators on this system
000110 ****=
000120 LINE=END                  - Exit 'like SDSF'
000130 LINE=HELP                 - Display help information
000140 CMD =X            END
000150 CMD =x            END
000160 CMD =QQ list job * QUEUE=I
000170 CMD =QO list job * QUEUE=O
000180 CMD =QX list ACT * TYPE=I
000190 CMD =QH list job * QUEUE=H
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
```

**Explanation**

The options have no subsystem or authorization restrictions. Note the use of command synonyms.

If you wish to make this menu available from any menu at your site, you must add the following line to the options on that menu definition:

```
LINE=#    MENU SDSF       SDSF       - SDSF-like options
```

### Example 5: Menu to Provide Example Menu Options

The following example menu illustrates how standard utilities and functions can be integrated into the Natural ISPF environment by defining them as menu options:

```
------------------------ NATURAL ISPF example menu --------------------------
OPTION  ===>

                                            Userid   MBE
                                                              Time     13:02:26
      Integrate menus                                         Terminal DAELC521
   1  NATURAL    - Utilities                                  Library  MBE
   2  SDSF       - SDSF entry panel                           Node     148
      Integrate NATURAL programs/applications
   3  SYSPROD    - Installed products
   4  PREDICT    - PREDICT Dictionary
      Integrate Batch jobs
   5  ADAREP     - ADABAS Database Report
   6  TAPE       - Tape creation
   7  TAPESCAN   - Tapescan for VSE
      Use TSO Batch interface
   8  TSO HELP   - Display Help for TSO commands
   9  TSO MENU   - Use real TSO
      Use internal command
   10 CON-NECT   - List CON-NECT inbasket

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 example menu definition looks as follows:

```
EDIT-MNU:EXAM ------------------------------------------------ Columns 001 072
 COMMAND===>                                                   SCROLL===> CSR
****** ******************************* top of data ******************************
000010 HDR =NATURAL ISPF example menu
000020 HELP=MAIN
000030 ****=Integrate menus
000040 LINE=#    MENU UTIL   NN1 NATURAL    - Utilities
000050 LINE=#    MENU SDSF   M   SDSF       - SDSF entry panel
000060 ****=Integrate NATURAL programs/applications
000070 LINE=#    NAT SYSPROD NN1 SYSPROD    - Installed products
000080 LINE=#    APP SYSDIC  NN1 PREDICT    - PREDICT Dictionary
000090 ****=Integrate Batch jobs
000100 LINE=#    ADAREP      M   ADAREP     - ADABAS Database Report
000110 LINE=#    TAPE        M   TAPE       - Tape creation
000120 LINE=#    DOST        D   TAPESCAN   - Tapescan for VSE
000130 ****=Use TSO Batch interface
000140 LINE=#    TSOH        M31 TSO HELP   - Display Help for TSO commands
000150 LINE=#    MENU TSO    M   TSO MENU   - Use real TSO
000160 ****=Use internal command
000170 LINE=#    POST        C   CON-NECT   - List CON-NECT inbasket
000180 CMD =X            END
000190 CMD =x            END
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
```

Issue the DOWN command (usually assigned to PF8) to display the rest of the menu definition:

```
EDIT-MNU:EXAM ------------------------------------------------ Columns 001 072
 COMMAND===>                                                   SCROLL===> CSR
000200 CMD =POST LS DOC * FO=POSTEINGANG FI=NEU
000210 CMD =ADAREP SUBMIT PDS SAG.ISP141.SRCE(EXADAREP)
000220 CMD =TAPE SUBMIT MACRO SYSISPE(MAC-MVS4)
000230 CMD =DOST SUBMIT MACRO SYSISPE(MAC-VSE2)
000240 CMD =TSOH SUBMIT MACRO SYSISPE(MAC-TSO1),TYPE=TSO
****** ***************************** 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
```

**Explanation**

The menu options speak for themselves. Note the use of command synonyms to submit specific objects using a single simple command. In this way, site-specific functions can be made easy to use, for example, the command POST has been defined to check the Con-nect Inbasket.

If you wish to make this menu available from any menu at your site, you must add the following line to the options on that menu definition:

```
LINE=#   MENU EXAM        A1 EXAMPLE      - Invoke example menu
```

### Example 6: Menu NULL for Unauthorized Users

If you wish to deny access to undefined users for whom there is no prefix authorization, you can define a menu similar to the following to be displayed when the user logs on:

```
------------------------------- NATURAL ISPF --------------------------------
OPTION  ===>
                                                                 Userid   MBE
                                                                 Time     17:52:12
        You are not authorized to use NATURAL ISPF,              Terminal DAEDC617
        please contact your ISPF administrator, who              Library  MBE
        will give you access to the system.                      Node


   1   Terminate 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 menu definition looks like the following:

```
EDIT-MNU:NULL ------------------------------------------------ Columns 001 072
 COMMAND===>                                                   SCROLL===> CSR
****** ***************************** top of data ****************************
000010 HDR =NATURAL ISPF
=cols> ----+----1----+----2----+----3----+----4----+----5----+----6----+----7--
000020 ****=
000030 ****=You are not authorized to use NATURAL ISPF,
000040 ****=please contact your ISPF administrator, who
000050 ****=will give you access to the system.
000060 ****=
000070 ****=
000080 LINE=#    END             Terminate NATURAL ISPF
****** **************************** 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
```

You must specify menu name `NULL` as Main Menu in the authorization table of the default definition (*) which is assigned to undefined users for whom there is no prefix (group) definition (see the section ***User Definitions***).

## Maintaining Menus with Function Commands

Natural ISPF menus are separate objects within Natural ISPF with object type `MNU`. This means that you (and other authorized users) can maintain menu definitions with function command syntax entered from any system screen.

The available function commands are:

| Command | Object Parameter Syntax |
|---|---|
| `LIST` | *`prefix`* |
| `EDIT` | *`menuname`* |
| `DELETE` | *`menuname`* |

where prefix is a prefix notation followed by the asterisk wildcard (*) to select only those menus that start with that prefix. The default is the asterisk wildcard (*) for a list of all menus.

**Note:** If you issue any of the commands without parameters, Natural ISPF prompts you for valid values.

The following examples are provided below:

- Example: EDIT
- Example: LIST

### Example: EDIT

The command:

```
EDIT MNU MAIN
```

displays the Main Menu definition in Editor format. You can modify the menu and save it with the command `SAVE`.

**Note:** If you delete the modified menu, you only back out your changes: you cannot delete a system menu supplied by Software AG.

### Example: LIST

The command:

```
LIST MNU  *
```

lists all defined menus in the same format as when you select the `MENU LIST` option from the Administrator Menu, for example:

```
LIST-MNU:* ------------------------------------ Row 0 of 13 - Columns 006 051
 COMMAND===>                                                  SCROLL===> CSR
   MENU          TYPE HEADER
** ******************** top of list ******************
   ADMN          SYS  CONFIGURATION MENU
   BTCH          USER Batch Utilities Menu
   DOC           USER NATURAL ISPF doctest
   ISPF          USER JCL for ISPF development and ins
   MAIN          USER NATURAL ISPF MAIN MENU
   NULL          USER Sorry you are not authorized to
   MPAN          SYS  NSPF MAIN MENU
   SAG           USER Software AG MENU
   PROF          SYS  PROFILES MENU
   SDSF          USER 'like SDSF' primary option menu
   TEST          USER 'LIKE SDSF' PRIMARY OPTION MENU
   SYS           SYS  SYSTEM FACILITIES
   UTIL          USER NATURAL ISPF Utility Menu
** ****************** 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 |
|---|---|
| MENU | Name of menu. |
| TYPE | Type of menu. Possible values:<br><br>SYS System menu copied from installation medium as delivered by Software AG.<br><br>USER Modified or new menus defined by authorized users.<br><br>When your Natural ISPF system is upgraded by a later version, only SYS-type menus are overwritten by the new menus from the installation medium (see note below). |

You can select any menu from the list with the E (EDIT) or D (DELETE) line command entered in the input field preceding the menu name. The EDIT option allows you to modify the menu definition.

**Note:** When modifying a SYS-type menu, you actually create a new USER-type copy of it. If you delete this menu later, only the USER-type version is deleted, and the old SYS-type menu is displayed again. It is not possible to delete SYS-type menus.

# 5 Site-Specific Online Information


- Site-Specific Help ..... 90
- Site-Specific Information - UINFO ..... 92
- Text Syntax ..... 92

You can make site-specific information available online to Natural ISPF users in either of two ways:

- As menu-related online help texts displayed when the user issues the `HELP` command from a particular menu. See the subsection ***Site-Specific Help***;
- As text presented in screens organized in a tree structure through which the user can navigate. The first (root) screen is displayed when the user issues the `UINFO` command. See the subsection ***Site-Specific Information - UINFO***.

You can write both types of text with the Editor using the same syntax.

The subsection ***Text Syntax*** contains a full description of the definition syntax.

## Site-Specific Help

All help texts are usually stored as Natural members in the `FNAT` system file. Texts supplied by Software AG are stored in the System Help Library. You must not modify these texts. If you wish to create your own help texts, you must use the User Help Library.

The following topics are covered below:


- Overwriting Existing Help
- Defining New Help Texts


### Overwriting Existing Help

Overwriting an existing help text means creating a member in the User Help Library with the same name as a member in the System Help Library. When a user invokes the help text with the `HELP` command, the text is searched first in the User Help Library. Only if the member is not found there is the help text supplied by Software AG in the System Help Library displayed.

The member name of each existing help text is displayed at the end of the command line of the help screen.

For example, if you issue the command:

```
HELP SPLIT
```

the Software AG help screen explaining the `SPLIT` command is displayed:

```
HELP----------- SESSION COMMANDS (working with N-ISPF) SPLIT ----->>> continued
COMMAND ==>                                                          SPLIT

 SPLIT         Splits the screen to start a new Natural ISPF
               session. The screen
               is split at cursor position. The new session is started
               in the lower part of your terminal screen.
               You can change the size of the screen portions devoted
               to each session by moving the cursor to where
               you require the screen division and repeating the SPLIT
               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 Curso
```

The member name in the System Help Library is `SPLIT`, displayed top right in the command line. If you wish to create a different help text for the `SPLIT` command, create the member `SPLIT` in the User Help Library.

### Defining New Help Texts

You can write new help texts as members in the User Help Library. Typically, you will write new help texts for `USER`-type menus. To display the member as the result of the `HELP` command, you must specify the member name of the help text in the `HELP=` line of the menu definition, or reference the member in another help member (see the subsection ***Text Syntax***).

For examples of help text definitions, see any member in the System Help Library.

## Site-Specific Information - UINFO

Site-specific information texts can be any information of interest to the user. The first information screen is displayed when the user issues the command `UINFO`.

You must define the first (root) screen in the member `UINFO` in the User Information Library. The syntax of the text definition is the same as for site-specific help texts (see the subsection *Text Syntax*). You can define information screens in a tree structure.

**Note:** When defining information members, you are advised to use a naming convention. For example, you could name each member according to its level and number within the tree structure using the same prefix (the root member is always `UINFO`). Alternatively, you could use the prefix `ISPU` and use the remaining 4 bytes for a self-explanatory indication of the member contents.

An example of site-specific information is contained in the following subsection.

## Text Syntax

Information and help text members consist of two types of lines: lines of text to be displayed and special lines for blank lines and navigational purposes.

Text lines can contain special characters to start and end text attributes:

| Character | Explanation |
|---|---|
| # | Starts highlighting (number sign, hexadecimal `7B`). |
| + | Activates reverse video: use this attribute for words for which there is a separate help text. |
| @ | Deactivates attributes (commercial at, hexadecimal `7C`). |

**Note:** See *Special Characters* at the end of this documentation for a list of special characters with their hexadecimal values.

The special lines are defined by an ampersand (&) in Column 1 followed by a character and navigational instructions. The following table lists all available options including their format and maximum length:

| | |
|---|---|
| &H *header* (A50) | Where the string *header* is displayed in the header line of the information screen. |
| &L | Creates a blank line in the information screen. |
| &F *member* (A08) | Where *member* is the name of the member to be displayed when the end of the current text is reached and the user presses the ENTER key or issues the command DOWN (PF8). |
| &B *member* (A08) | Where *member* is the name of the member to be displayed when the user issues the command UP (PF7) from the top of the member. |
| &U *member* (A08) | Where *member* is the name of the member to be displayed when the user issues the command UP LEVEL (L + PF7) from the top of the member, or the command BACK. |
| &S *navigation-command member* | Where *navigation-command* invokes the *member* when the user enters the navigation command in the command line of the current screen. Navigation commands can consist of up to 4 characters and provide a menu-like selection mechanism to the user to display information from the next level in the information screen tree structure. |

## Example

Below is an example of a UINFO member definition:

```
EDIT-NAT:SYSISPIU(UINFO)-Program->Struct-Free-45K --- >>> Versioning is invoked
 COMMAND===>                                                  SCROLL===> CSR
****** ****************************** top of data ******************************
000010 &H Demo site info system
000020 &S 0 ISPUNET
000030 &S 1 ISPUPROD
000040 &S 2 ISPUBAT
000050 &L
000060      Please select whatever you like
000070 &L
000080     #0@  Demo site NETWORK
000090     #1@  Installed products
000100     #2@  Batch job limits
****** ***************************** 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
```

Note that the numbers 0, 1 and 2 are defined as navigation commands that invoke the corresponding members when the user enters these numbers in the command line. This means that member ISPUNET contains information on option Demo site NETWORK, member ISPUPROD contains information on option Installed products and member ISPUBAT offers information on option Batch job limits.

When a user issues the command UINFO, the following screen is displayed:

```
 INFO----------------------- Demo site info system -----------------------------
 COMMAND ==>                                                          UINFO

      Please select whatever you like

      0   Demo site NETWORK
      1   Installed products
      2   Batch job limits









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 user can display information on the displayed items by entering Option Number 0, 1 or 2 in the command line.

For examples of site-specific help texts, see the member NEWS and related members in the System Help Library. This member defines the help screen displayed as a result of the command NEWS, which is handled similarly to the command UINFO.

# 6 User Exits


- Object Exits ..... 96
- Examples of Object Exits ..... 99
- CA Panvalet Save Exit ISPT-SVU ..... 101
- Logon Exit ISP-LONU ..... 101
- Logoff Exit ISP-LOFU ..... 102
- Print User Exit ISP-PRTU ..... 103
- Import/Export Exits ..... 105
- Color Settings Exit ISP-ECLU ..... 106
- Resume Exit ISP-RESU ..... 106
- Suspend Exit ISP-SUSU ..... 108
- Session Exit ISP--S-U ..... 109
- Rename Function Exit ISP-RN-U ..... 110
- User Library Exit ISP-PRFU ..... 111
- User Group Exit ISP-UGRU ..... 111
- Node Exit ISP-NODU ..... 113
- HSM - Hierarchical Storage Manager Exit ISP-HSMU ..... 114
- Editor Profile Exit ISP-ED-U ..... 115
- Incore Database Defaults Exit IDB-USRN ..... 116
- Container File Access Exit IDBC---U ..... 117

Natural ISPF provides a number of user exits written in Natural. The sources are delivered in the Exit Library. Before modifying any of these programs, you must copy them to one of your Natural libraries, as any subsequent INPL from the installation medium overwrites the Exit Library. Useful examples for user exits can be found in the Example Library.

After successful modification of an exit, you must copy the module to library SYSLIB, and activate the exit as described in the section *System Configuration*.

The various types of provided exits are described in the following subsections.

## Object Exits

A user exit is provided for each Natural ISPF object type (for example, PDS members, Natural objects, views, jobs, etc.).

Whenever a user issues a specified function command for a certain object, the corresponding user exit is called before the command is executed. A user exit can check whether the function, object type and parameters are valid for the user and can react in any of the following ways:

- Deny access to the function and return an error message;
- Return a warning message;
- Modify invalid function parameters;
- Allow access to the function.

### Data Parameters

The data parameters you can define for object exits are the same for each object type:

<table>
<tr><th>Parameter</th><th>Format</th><th>Type</th><th colspan="4">Meaning (Member: TAB-FUNC)</th></tr>
<tr><td rowspan="11">#FUNCTION</td><td rowspan="11">(A2)</td><td rowspan="11">I</td><td colspan="4">Contains a 2-character abbreviation of the function to be executed. Possible options:</td></tr>
<tr><td>--</td><td>ENTRY</td><td>FR</td><td>FORMAT</td></tr>
<tr><td>-2</td><td>Target of COPY</td><td>HL</td><td>HOLD</td></tr>
<tr><td>AL</td><td>ALLOCATE</td><td>IN</td><td>INFORMATION</td></tr>
<tr><td>BR</td><td>BROWSE</td><td>LS</td><td>LIST</td></tr>
<tr><td>CC</td><td>Condition codes</td><td>OT</td><td>OUTPUT</td></tr>
<tr><td>CH</td><td>CHANGE</td><td>PG</td><td>PURGE</td></tr>
<tr><td>CM</td><td>COMPRESS</td><td>PL</td><td>PLAY</td></tr>
<tr><td>CP</td><td>COPY source</td><td>PR</td><td>PRINT</td></tr>
<tr><td>CR</td><td>COMPARE</td><td>RL</td><td>RELEASE</td></tr>
</table>

| Parameter | Format | Type | Meaning (Member: TAB-FUNC) | | | |
|---|---|---|---|---|---|---|
| | | | CT | CATALOG | RN | RENAME |
| | | | DF | DEFINITION | RU | RUN |
| | | | DI | DIFFERENCE | SB | SUBMIT |
| | | | DL | DELETE | ST | STATUS |
| | | | DO | DOWNLOAD | UN | UNCATALOG |
| | | | DS | DESCRIPTION | UP | UPLOAD |
| | | | ED | EDIT | XE | EXECUTE |
| | | | ET | EXTENTS | XT | EXTERNS |
| | | | EX | EXPORT | ZP | ZAPS |
| | | | FL | FOLLOW | | |
| | | | **Note:** Not all functions are valid for all object types. See the *Natural ISPF User's Guide*. | | | |
| #SES-DATA | (A253) | I/O | Contains object-specific parameters; you must redefine these according to the object type. A parameter data area is delivered in source form for this redefinition. | | | |
| #ERROR-CODE | (N3) | O | Function will be denied if an error code greater than 0 (zero) is received. | | | |
| #ERROR-NR | (N4) | O | Error number for SYSERR, the errors greater than 9000 in the System Profile Library are not used by Software AG and you can therefore define them (currently, 9001 and 9002 are used for example exit ISPN---U, but these can be overwritten). If this field is set and ERROR-CODE is zero, the message will be displayed as a warning, unless more important messages (like FOLLOW) have to be displayed. | | | |
| #ERROR-PARM | (A75) | O | Parameters for the error message can be passed to the Natural ISPF error handler, multiple parameters must be separated by a semicolon (;). They replace :1:, :2:, :3:, etc. parameters in the error text. | | | |
| #OPTIONS | (A20) | I/O | Global data which can be shared by all user exits and by all Open NSPF subprograms. Can also be used to transfer control to another object (also an Open NSPF object). This is useful if a user-written routine handles functions for existing objects. The syntax is:<br><br>`'OBJECT = xx'`<br><br>where: xx is the object code as defined in the CONTROLS table. The field is cleared by Natural ISPF when transferring control to the new object.<br><br>Example: | | | |

| Parameter | Format | Type | Meaning (Member: TAB-FUNC) |
|---|---|---|---|
| | | | MOVE 'OBJECT = -7' TO #OPTIONS |

### Table of Exits and Object Abbreviations

This table lists all object-related user exits provided, as well as the data areas used by them, and an abbreviation of the object valid for the OBJECT parameter.

The data areas are also delivered in source form, the fields used have meaningful names and are documented in the data area source itself.

| Exit Name | Data Area | Object (Member: TAB-EXIT) | Object Abbreviation |
|---|---|---|---|
| ISBD---U | ISBD---L | BS2000 files | BF |
| ISBJ---U | ISBJ---L | BS2000 jobs | BJ |
| ISBL---U | ISBL---L | LMS elements | LMS |
| ISBV---U | ISBV---L | BS2000 job variables | JV |
| ISB6---U | ISBL---L | LMS element versions | LMV |
| ISDA---U | ISDA---L | z/VSE active jobs | DA |
| ISDD---U | ISDD---L | z/VSE files | FIL |
| ISDJ---U | ISDJ---L | z/VSE jobs | DJ |
| ISDL---U | ISDP---L | z/VSE sub-libraries | SUB |
| ISDP---U | ISDP---L | z/VSE members | MEM |
| ISDR---U | ISDR---L | z/VSE member versions | VV |
| ISDZ---U | ISDP---L | z/VSE volumes | DV |
| ISIC---U | ISIC---L | Incore container files | CTN |
| ISPA---U | ISPA---L | z/OS active jobs | A |
| ISPB---U | ISPB---L | Buffer Pool files | BPF |
| ISPC---U | ISPC---L | Console | CON |
| ISPD---U | ISPD---L | z/OS data sets | D |
| ISPE---U | ISPE---L | Recovery files | R |
| ISPF---U | ISPN---L | ISPF configuration | F |
| ISPG---U | ISPJ---L | Syslog | LOG |
| ISPJ---U | ISPJ---L | z/OS Jobs | J |
| ISPK---U | ISPP---L | CSECT | CST |
| ISPL---U | ISPL---L | CA Librarian | LIB |
| ISPM---U | ISPM---L | ISPF Menus | MNU |
| ISPN---U | ISPN---L | Natural | N |
| ISPO---U | ISPO---L | Output (workpool entries) | O |

| Exit Name | Data Area | Object (Member: TAB-EXIT) | Object Abbreviation |
|---|---|---|---|
| ISPP---U | ISPP---L | PDS members | P |
| ISPR---U | ISPR---L | PDS member versions | PV |
| ISPS---U | ISPJ---L | z/OS SYSOUT files | S |
| ISPT---U | ISPT---L | CA Panvalet | PAN |
| ISPU---U | ISPU---L | ISPF users | U |
| ISPV---U | ISPV---L | Views | V |
| ISPX---U | ISPN---L | Macro | MAC |
| ISPY---U | ISPY---L | Natural errors | E |
| ISPZ---U | ISPP---L | z/OS volumes | VOL |
| ISP1---U | ISP1---L | Member versions | MV |
| ISP2---U | ISP2---L | Natural versions | NV |
| ISP4---U | ISPE---L | Buffer Pool Recovery files | BPR |
| ISP5---U | ISPL---L | CA Librarian versions | LV |

## Examples of Object Exits

### Example 1: Exit for Natural

The following program is invoked when a user issues an `EDIT` command for a Natural object. It restricts write access to objects in Natural library `NSPFWORK` to users `JWO`, `GW1` and `MBE`. The program warns these specified users to be careful, and unauthorized users are presented with an error message.

```
* JOB USER EXIT
  *
  * List JOB queue without selection criteria is not allowed.
  * exit modifies the selection criteria and puts first 3 characters of
  * user-id into it.
  *
  DEFINE DATA PARAMETER
  1 #FUNCTION(A2)
  PARAMETER USING ISPJ---L
  PARAMETER
  1 #ERROR-CODE(N3)
  1 #ERROR-NR  (N4)
  1 #ERROR-PARM(A75)
  1 #OPTIONS   (A20)
  LOCAL
  1 #A3     (A3)
  END-DEFINE
  *
```

```
  DECIDE ON FIRST VALUE OF #FUNCTION
    VALUE 'LS'
      IF #JOBNAME   = ' ' OR = '*'
          MOVE *USER TO #A3
          COMPRESS #A3 '*' INTO #JOBNAME LEAVING NO
      END-IF
    NONE IGNORE
  END-DECIDE
  END
```

### Example 2: Exit for JOBS

This exit is invoked when a user issues the `LIST` command on object type `JOBS`. It does not allow a list request without selection criteria and writes the first three characters of the user ID to the job name parameter:

```
* JOB USER EXIT
  *
  * List JOB queue without selection criteria is not allowed.
  * exit modifies the selection criteria and puts first 3 characters of
  * user-id into it.
  *
  DEFINE DATA PARAMETER
  1 #FUNCTION(A2)
  PARAMETER USING ISPJ---L
  PARAMETER
  1 #ERROR-CODE(N3)
  1 #ERROR-NR  (N4)
  1 #ERROR-PARM(A75)
  1 #OPTIONS   (A20)
  LOCAL
  1 #A3     (A3)
  END-DEFINE
  *
  DECIDE ON FIRST VALUE OF #FUNCTION
    VALUE 'LS'
      IF #JOBNAME   = ' ' OR = '*'
          MOVE *USER TO #A3
          COMPRESS #A3 '*' INTO #JOBNAME LEAVING NO
      END-IF
    NONE IGNORE
  END-DECIDE
  END
```

## CA Panvalet Save Exit ISPT-SVU

This exit is called after each successful save of a CA Panvalet member. The exit must be activated by the definitions in the PANDEF member (see the section *System Configuration*).

An example program is delivered in the User Exit Library. It illustrates how to access the output from PAM#1 and displays the output on the screen.

The following table lists all definable parameters:

| Parameter | Format | Type | Meaning |
|---|---|---|---|
| P1 | (A250) | I/O | Exit control block has to be redefined with the following definitions. |
| FILLER-1 | (A12) | | |
| EX-RNUM | (B4) | I | Number of records (saved). |
| EX-RLEN | (B2) | I | Record length. |
| FILLER-2 | (A2) | | |
| EX-ECODE | (B2) | O | Error code. |
| FILLER-3 | (A10) | | |
| EX-ETEXT | (A64) | O | Error text. |
| FILLER-4 | (A4) | | |
| EX-DSNAME | (A44) | I | CA Panvalet data set name. |
| FILLER-5 | (A10) | | |
| EX-MEMBER | (A10) | I | CA Panvalet member name. |
| EX-VOLSER | (A6) | I | Volume serial number. |
| EX-PASSWD | (A8) | I | Password. |
| FILLER-6 | (A1) | | |
| EX-NODE | (B1) | I | Entire System Server node. |

## Logon Exit ISP-LONU

This exit is called when the user logs on to Natural ISPF. It is executed after standard logon handling, and can be used to issue a command to Natural ISPF directly at logon. Any valid Natural ISPF command is possible; for example, this could be a PLAY command.

In addition to the standard **data parameters** described above, you must define the following data in the exit:

| Parameter | Format | Type | Meaning |
|---|---|---|---|
| #USER | (A8) | I | User ID of the user for whom the exit is to be executed. |
| #COMMAND | (A50) | I/O | Natural ISPF command sequence to be executed at logon. |

## Logoff Exit ISP-LOFU

This exit is called when the user logs off from Natural ISPF. It is executed after standard logoff handling, and can be used to issue a command to Natural directly at logoff. Any valid Natural command is possible.

In addition to the standard **data parameters** described above, you must define the following data in the exit:

| Parameter | Format | Type | Meaning |
|---|---|---|---|
| #USER | (A8) | I | User ID of the user for whom the exit is to be executed. |
| #COMMAND | (A50) | I/O | Natural ISPF command sequence to be executed at logon. |

### Example

The following example logoff exit issues the FIN (FINISH) command to Natural after user JWO logs off from Natural ISPF: the user is returned to the TP environment directly, bypassing Natural.

```
* LOGOFF USER EXIT
*
* SKIP NATURAL AT LOGOFF
* DEFINE DATA PARAMETER
1 #USER(A8)
1 #COMMAND(A50)
1 #ERROR-CODE(N3)
1 #ERROR-NR (N4)
1 #OPTIONS(A20)
END-DEFINE
*
IF #USER EQ 'JWO'
  MOVE 'FIN' TO #COMMAND
END-IF
*
END
```

## Print User Exit ISP-PRTU

When this exit is activated in the Natural ISPF configuration member by entering (PRINT, it is called twice whenever a user invokes the Natural ISPF print function.

The exit is invoked before the printer is opened, with FC=O. The following actions can be performed:

- DEFINE PRINTER (for Printer 2): A printer can be opened with the PROFILE parameter to activate specific profiles; #FC must be RESET in this case, no define printer will be executed by the caller;
- Abort print function by returning an error-code, error number and parameters.
- Modify printer and number of lines per page;
- Return an escape sequence to be printed as first line of the printout;
- Define whether the escape sequence has to be printed on printer CCONTROL;

The exit is also invoked before the printer is closed, with FC=C. The following action can be performed:

- DEFINE/CLOSE PRINTER for Printer (2). #FC must be RESET in this case, no close printer will be executed by the caller;
- Return an escape sequence to be printed as last line of the printout;
- Define whether the escape sequence has to be printed on Printer CCONTROL.

The following is an example of a customized user print exit: ISP-PRTU, Lib.NSPFEXAM.

```
* *********************  **    ************************************
DEFINE DATA PARAMETER
* ********************** **    ************************************
1 #FC       (A1)     /* I/O   Function-Code: O=open,C=Close
                     /*       when reset to ' '
                     /*       no Open/Close will be done by caller
1 #PRINTER  (A8)     /* I/O   Printer id
1 #OBJECT   (A2)     /* I     Object type to be printed
1 #SES-DATA (A200)   /* I     Session data for object
1 #RECLEN   (N4)     /* I     Length of records to be printed
1 #CCONTROL (N1)     /* O     CCONTROL available:
                     /*        0 = no
                     /*        1 = yes
                     /*        2 = yes and printer can handle MCC
                     /*            (machine code control chars) for future use
1 #PROFILE  (A8)     /* O     Printer profile for future use
1 #ESC-SEQ  (A80)    /* O     Esc-sequence to be printed:
                     /*       #FC='O' before the first line
                     /*       #FC='C' after  the last  line
1 #NO-LINES (N3)     /* I/O   No of lines per page
```

```
1 #REFRESH-SCREEN(L) /* O     True if screen has to be refreshed
                     /*       Must be set if this exit does any
                     /*       terminal I-O
1 #ERROR-CODE  (N3)  /* O     AS USUAL
1 #ERROR-NUMBER(N4)  /* O     AS USUAL
1 #ERROR-PARM(A75)   /* O     AS USUAL
1 #OPTIONS(A20)      /* I/O   FFU
1 #WORK   (A20)      /* I/O   Internal work area
LOCAL
1 #START-SEQ
  2 #S-B (A17) INIT <' &%21,132;99,999&'>
  2 #S-P (A63) INIT <'!R! SPO L; FONT 23; SCPI 14; SLPI 9; EXIT;'>
1 REDEFINE #START-SEQ
  2 #START (A80)
1 #END-SEQ
  2 #E-B (A17) INIT <'&%21,80;99,999&'>
  2 #E-P (A63) INIT
    <' !R! SPO P; FONT 8; SCPI 12; SLPI 6; UNIT I; SLM 1; EXIT;'>
1 REDEFINE #END-SEQ
  2 #END (A80)
1 #LANDS (A1)
END-DEFINE
*
DEFINE WINDOW WIND1 SIZE 04 * 30
  BASE 04/25
  CONTROL WINDOW
  FRAMED ON
*
*
IF #PRINTER  = 'EDITOR' OR = 'WORKPOOL'
    OR #RECLEN LE 80
  ESCAPE ROUTINE
END-IF
*
*
* If the printer is called 'EXIT' then Open/Close is done
* by this routine (and not by ISPF)
*
IF #PRINTER EQ 'EXIT'
  DECIDE ON FIRST VALUE OF #FC
    VALUE 'O'                          /* open call
      DEFINE PRINTER (2) OUTPUT 'MYPRINT'
      RESET #FC                        /* open done
    VALUE 'C'                          /* close call
      DEFINE PRINTER (2) OUTPUT 'MYPRINT' /* valid for nat21
      RESET #FC                        /* close done
    NONE IGNORE
  END-DECIDE
  ESCAPE ROUTINE
END-IF
*
*
```

```
* This example sets landscape mode on kyocera printer, if
* the records to be printed contain more than 80 bytes.
*
IF #PRINTER EQ 'DAEPRT14' OR         /* kyocera printers
    #PRINTER EQ SCAN 'TEMP'
  DECIDE ON FIRST VALUE OF #FC
    VALUE 'O'                        /* open call
      SET WINDOW 'WIND1'
      INPUT 'Mark for Landscape' #LANDS(AD=MI)
      MOVE TRUE TO #REFRESH-SCREEN
      RESET #WORK
      MOVE #LANDS TO #WORK
      IF #LANDS NE ' '
      MOVE  #START TO #ESC-SEQ       /* set landscape mode
      CALLNAT 'ISP-C22N' #NAT22
      IF #NAT22                      /* if executing under NAT22
        MOVE 1 TO #CCONTROL          /* print on CCONTROL
      END-IF
      END-IF
    VALUE 'C'                        /* close call
      MOVE #WORK TO #LANDS
      IF #LANDS NE ' '
      MOVE  #END   TO #ESC-SEQ       /* set mode back
      END-IF
    NONE IGNORE
  END-DECIDE
END-IF
END
```

## Import/Export Exits

Natural ISPF provides exits that are required when you wish to change the default work file number for the `IMPORT/EXPORT` PC function. The default work file number is 7. These routines contain all `READ/WRITE` operations. You can adapt the source to enforce use of another work file.

After each source change, you must recompile the programs and copy them to the `SYSLIB` library. Note especially, that if you change one program, you must change them all accordingly.

The import/export exits are:

ISP-SEPU
ISP-SE2U
ISP-SE3U
ISP-SE4U
ISP-SIMU
ISP-UPBU
ISP-DLBU

ISP-SECU
ISP-SEWU

## Color Settings Exit ISP-ECLU

This exit is always called whenever Natural ISPF is invoked or suspended. It can be used to modify Natural color settings when leaving or entering Natural ISPF.

The following parameters are passed to the exit:

| Parameter | Format | Type | Meaning |
|---|---|---|---|
| #FC | (A1) | I | I = when entering Natural ISPF. Any other value = when suspending Natural ISPF. |
| #USER-ID | (A8) | I | User ID of the user for whom the exit is to be executed. |

## Resume Exit ISP-RESU

This program can be invoked when returning to Natural ISPF after execution of the command NATURAL or APPLICATION. This program can be used to display messages to the user.

To invoke this resume exit, you must enter (RESUME in the CONFIG member.

In addition to the standard **data parameters** described above, you must define the following data in the exit:

| Parameter | Format | Type | Meaning |
|---|---|---|---|
| #USER | (A8) | I | User ID of the user for whom the exit is to be executed. |
| #COMMAND | (A50) | I/O | Command to be executed. |

### Example

The following example program checks for new messages in the user's Inbasket in Con-nect, Software AG's Office system:

```
* USER EXIT TO BE INVOKED WHEN resuming NSPF
* *********************  **    ************************************
DEFINE DATA PARAMETER
* *********************  **    ************************************
1 #USER(A8)          /* I
1 #COMMAND(A50)      /* I/O   command to be executed
1 #ERROR-CODE(N3)    /* O     GT ZERO  command will not be executed
1 #ERROR-NUMBER(N4)  /* O     AS USUAL
1 #ERROR-PARM(A75)
1 #OPTIONS(A20)      /* I/O   FFU
LOCAL
*
* example to check con-nect inbasket items
*
1 #RETURN-CODE (N2)
1 #CABINET     (A8)
1 #PASSWORD    (A8)
1 #NEW-PHONE   (P8)
1 #NEW-MAIL    (P8)
1 #NEW-INVIT   (P8)
1 #OPEN-MAIL   (P8)
1 #POST-MAIL   (P8)
1 #MAIL-SUM    (P8)
1 #MARK        (A1)
END-DEFINE
*
DEFINE WINDOW WIND1 SIZE 06 * 30
  BASE 09/25
  CONTROL WINDOW
  FRAMED ON
SET WINDOW 'WIND1'
SET CONTROL 'Y45'
INPUT (AD=IM) 'Mark for mail check: ' #MARK
IF #MARK NE ' '
  MOVE #USER TO #CABINET
  CALLNAT 'Z-INBKT'
    #RETURN-CODE #CABINET #PASSWORD
    #NEW-PHONE #NEW-MAIL #NEW-INVIT
    #OPEN-MAIL #POST-MAIL
  IF #RETURN-CODE EQ 0
    COMPUTE #MAIL-SUM = #NEW-PHONE + #NEW-MAIL + #NEW-INVIT
    IF #MAIL-SUM GT 0
      MOVE #MAIL-SUM TO #ERROR-PARM
      MOVE 9004 TO #ERROR-NUMBER
    END-IF
  END-IF
END-IF
*
END
```

## Suspend Exit ISP-SUSU

This program can be invoked when suspending Natural ISPF before execution of the command `NATURAL` or `APPLICATION`. For example, the program can be used to prohibit execution of these commands by setting an error code.

To invoke this suspend exit, you must enter `(SUSP` in the `CONFIG` member.

In addition to the standard **data parameters** described above, you must define the following data in the exit:

| Parameter | Format | Type | Meaning |
|---|---|---|---|
| #USER | (A8) | I | User ID of the user for whom the exit is to be executed. |
| #COMMAND | (A50) | I | Command to be executed (APPLICATION/ NATURAL). |
| #PARM | (A78) | I | Parameters to be passed with the command. |

### Example of Data Parameters

```
* User exit to be invoked when ISPF is suspended
* .i.e before the commands NAT or APPL are executed
* *********************  **    ************************************
DEFINE DATA PARAMETER
* *********************  **    ************************************
1 #USER(A8)           /* I
1 #COMMAND(A50)       /* I
1 #PARM   (A78)       /* I
1 #ERROR-CODE(N3)     /* O     GT ZERO  command will not be executed
1 #ERROR-NUMBER(N4)   /* O     AS USUAL
1 #ERROR-PARM(A75)
1 #OPTIONS(A20)       /* I/O   FFU
END-DEFINE
*
IF *USER EQ 'JWO'
  IF #COMMAND EQ SCAN 'HUGO' OR
      #PARM    EQ SCAN 'HUGO'
    MOVE 1 TO #ERROR-CODE         /* not allowed
    MOVE 9001 TO #ERROR-NUMBER
  END-IF
END-IF
*
END
```

## Session Exit ISP--S-U

The session exit is invoked when a SUBMIT command has been entered in an edit or browse session. This exit can be used to disallow the submit function.

To invoke this exit, you must enter (SESS in the CONFIG member. For using this exit in the context of EXPORT operations, see the description of APPLYMOD 91.

In addition to the standard **data parameters** described above, you must define the following data in the exit:

| Parameter | Format | Type | Meaning |
|---|---|---|---|
| #OBJECT | (A2) | I | Identifies the object type to be submitted (for example, P for PDS member). For a list of possible values, see *Table of Exits and Object Abbreviations*. |
| #SES-DATA | (A200) | I | Parameters for this object. The session data is used differently for every object type. The correct redefinitions can be found in the data areas for the object, see also *Table of Exits and Object Abbreviations*. |
| #OPTIONS | (A20) | O | Current SAG Editor session number in format 'SES=*n*' or 'SES=*nn*' where *n* is a one-digit and *nn* a two-digit session number. |

### Example of Data Parameters

```
* Session user EXIT
*
DEFINE DATA PARAMETER
1 #OBJECT  (A2)
1 #SES-DATA(A200)
1 #FUNCTION-DATA(A64)
1 #FUNCTION(A2)
1 #ERROR-CODE(N3)
1 #ERROR-NR  (N4)
1 #ERROR-PARM(A75)
1 #OPTIONS   (A20)      /* I/O   for future use
```

# Rename Function Exit ISP-RN-U

The rename exit is invoked when a RENAME command has been entered. The exit is invoked by the new name and can check whether the new name is valid.

To invoke this rename exit, you must enter (RENAME in the CONFIG member.

In addition to the standard **data parameters** described above, you must define the following data in the exit:

| Parameter | Format | Type | Meaning |
|---|---|---|---|
| #OBJECT | (A2) | I | Identifies the object type to be renamed (for example, P for PDS member). For a list of possible values, see *Table of Exits and Object Abbreviations*. |
| #SES-DATA | (A200) | I | Parameters for this object. The session data is used differently for every object type. The correct redefinitions can be found in the data areas for the object, see also *Table of Exits and Object Abbreviations*. |
| #FUNCTION-DATA | (A64) | I | Contains the function parameters, in this case the new name of the object to be renamed. |

### Example

The following little example program reports new name ANTON as invalid.

```
* RENAME function user exit
*
 DEFINE DATA PARAMETER
 1 #OBJECT  (A2)
 1 #SES-DATA(A200)
 1 #FUNCTION-DATA(A64)
 1 #ERROR-CODE(N3)
 1 #ERROR-NR  (N4)
 1 #ERROR-PARM(A75)
 1 #OPTIONS   (A20)      /* I/O   for future use
 LOCAL USING ISP-RN-L
 LOCAL
 END-DEFINE
 MOVE #FUNCTION-DATA TO #FUNC-DATA-RN
 SET CONTROL 'WB'
 DISPLAY #OBJECT  #NEWNAME
 IF #NEWNAME EQ 'ANTON'
    MOVE 1 TO #ERROR-CODE
    MOVE 6800 TO #ERROR-NR
END-IF
*
END
```

## User Library Exit ISP-PRFU

User-specific data such as profiles and menus is stored in the User Profile Library, which is called SYSISPFU and resides on the FNAT system file.

With this exit, the names of all user-specific libraries can be changed. If the modified library name does not start with SYS, the data will be stored and subsequently read from the FUSER system file. This exit is always invoked when a user library is accessed.

If you want to change the library name, modify the program accordingly, compile it and copy it to SYSLIB to activate it.

| Parameter | Format | Meaning | |
|---|---|---|---|
| #LIBRARY | (A8) | The name of the library. SYS*xxx* libraries are read from FNAT, others from FUSER. | |
| #TYPE | (A1) | Library type. Possible values: | |
| | | ' ' | Profile library. Default: SYSISPFU |
| | | 'H' | Help text library. Default: SYSISPHU |
| | | 'N' | News text library (usually identical with the help text library). Default: SYSISPHU |
| | | 'U' | Uinfo library. Default: SYSISPIU |

This subroutine can be modified by the user. It can be used to modify the Natural library names, where site-specific Natural ISPF data is stored

If the standard library names (on the FNAT system file) are acceptable, do not modify this program.

## User Group Exit ISP-UGRU

When activated, this exit receives control each time a profile item is to be located within the User Profile Library.

With this exit, the sequence of user names or group names to be checked for existing profile definitions can be modified before accessing the database to search for them. In particular, the exit can erase some of the array entries from the array of group names passed to the exit, thus reducing the number of database calls required to locate a profile definition.

To make use of this option, modify the program accordingly, compile it an copy it to SYSLIB. To activate the user group exit, you must enter (GROUPS in the CONFIG member. In addition to the standard **data parameters** described above, you must define the following data in the exit:

| Parameter | Format | Type | Explanation |
|---|---|---|---|
| #S-OPTION | (A1) | I | Search option in use: Contains T if APPLYMOD 101 is set to S.<br><br>Contains Q if " " is set to P.<br><br>Contains D in all other cases. |
| #USER-ID | (A8) | I | User ID of current session or of the user being edited (if invoked in the context of user maintenance). |
| #PROFCHAIN | (A253) | I/O | To be redefined with the following array: |
| #CHAIN-GROUPID | (A8/1:23) | | Array of user IDs, group IDs or prefixes to be searched for. This array will be filled by Natural ISPF before calling the exit, in the manner specified by APPLYMOD 101. |

## Example

```
DEFINE DATA
PARAMETER
1 #S-OPTION               (A1)    /* IN : search option derived
*                                 /* from APPLYMOD / global flag
*                                   = C: compatibility mode:
*                                          use old prefix logic
*                                   = D: like C, but invoke user exit
*                                   = S: search all NSC groups
*                                        (first priviledged,then others)
*                                   = T: like S, but invoke user exit
*                                   = P: serach priviledged NSC groups
*                                        only
*                                   = Q: like P, but invoke user exit
*                                         ISP-UGRU after building chain
1 #USER-ID                (A8)    /* IN : User-Id (from GDA)
1 #PROFCHAIN              (A253)  /* OUT:chain of profiles to be searched
1 REDEFINE #PROFCHAIN
  2 CHAIN-GROUPID         (A8/1:23) /* WARNING: you are advised NOT to
                                    /*  extend the length of this array !
                                    /* NOTE: adding array entries will
                                    /*   slow down performance, removing
                                    /*   entries will speed up NATURAL ISPF
1 #ERROR-CODE(N3)
1 #ERROR-NR  (N4)
1 #ERROR-PARM(A75)
1 #OPTIONS   (A20)      /* I/O   for future use
*
LOCAL
1 #I (N2)
END-DEFINE
*
* Example: If in your environment, only department ids consisting of
*          2 characters are used for defining group profiles, you can
```

```
*          reduce the number of array entries in the following way:
*           IN: FIAA068,FIAA06*,FIAA0*,FIAA*,FIA*,FI*,F*,*
*           OUT:FIAA068,FI*,*
FOR #I = 1 TO 23
  IF CHAIN-GROUPID(#I) = MASK (XX'*') #USER-ID
    CHAIN-GROUPID(2) := CHAIN-GROUPID(#I)
    CHAIN-GROUPID(3) := '*'
    RESET CHAIN-GROUPID(4:23)
    ESCAPE ROUTINE
  END-IF
END-FOR
*
END
```

## Node Exit ISP-NODU

This exit can be used to check whether access to a specific node is allowed. It is invoked whenever a user enters a Natural ISPF function or command, which needs access to an Entire System Server node.

By setting #ERROR-CODE and #ERROR-NUMBER access to a specific node can be disallowed. If the exit permits access to a specific node, this information is stored in Natural ISPF and the exit is not called any more with the same node ID.

To invoke the node exit, you must enter (NODE in the CONFIG member.

In addition to the standard **data parameters** described above, you must define the following data in the exit:

| Parameter | Format | Type | Meaning |
|---|---|---|---|
| #NODE | (N5) | I | Node ID to be checked. |

### Example

```
DEFINE DATA PARAMETER
1 #NODE       (N5)       /* I     Node ID to be checked. Note: N5 format.
1 #ERROR-CODE (N3)
1 #ERROR-NR   (N4)
1 #ERROR-PARM (A75)
1 #OPTIONS    (A20)      /* I/O   for future use
END-DEFINE
*
* Restrict access to node 148
*
IF #NODE NE 148
```

```
    #ERROR-CODE := 1        /* set return code
    #ERROR-NR   := 7009     /* and message number (e.g. Invalid node)
  END-IF
  END
```

## HSM - Hierarchical Storage Manager Exit ISP-HSMU

This exit is invoked (if activated in CONFIG) whenever a migrated data set has to be recalled before Natural ISPF recalls the data set.

This exit can recall the data set by submitting a batch job and inform the user to try again later. A special DELETE handling for migrated data sets can also be coded within this exit.

To invoke the HSM exit, you must enter either (HSM or (HSM-S in the CONFIG member:

- (HSM
  Setting the option (HSM in the CONFIG member causes the user exit to be invoked after the user has been prompted by Natural ISPF and has confirmed recall of the data set.
- (HSM-S
  Setting the option (HSM-S activates the user exit in "silent mode", that is, Natural ISPF's prompting is suppressed: the exit will be invoked without any prompting, whenever a user-initiated function refers to a migrated data set. This option is useful when you want the user exit to set up its own customized prompting dialog.

In addition to the standard **data parameters** described above, you must define the following data in the exit:

| Parameter | Format | Type | Meaning | |
|---|---|---|---|---|
| ISPD---L | | I | Contains session data including the field #DSNAME. | |
| #RECALL-STATUS | (N1) | 0 | Possible options: | |
| | | | 0 | Exit did not start RECALL. RECALL will be performed by caller. |
| | | | 1 | RECALL terminated successfully. Caller will continue processing. |
| | | | 2 | RECALL started but not yet terminated. Caller will abort processing and inform user to try again later. |
| | | | 3 | DELETE operation has been initiated or completed (valid only if #OPTIONS='FCT=DELETE'). Caller will abort processing and signal deletion with appropriate message. |

| Parameter | Format | Type | Meaning | |
|---|---|---|---|---|
| | | | 9 | RECALL rejected by user or user exit. Makes sense mainly in "silent mode", that is, with (HSM-S in CONFIG member. Caller will abort processing. |
| #OPTIONS | (A20) | I/O | This field usually contains blanks. If it contains FCT=DELETE, this indicates that the exit is being invoked while handling a DELETE-DATASET command issued by the user. The exit can either ignore this information, or else it can set appropriate actions to delete the data set from archive without recalling it. In the latter case, #RECALL-STATUS must be set to 3.<br><br>**Caution:** At the time when the exit is invoked, Natural ISPF has not yet prompted for a confirmation of file deletion. If desired, this prompting has to be performed by the user exit. | |

### Example

```
DEFINE DATA PARAMETER
      USING ISPD---L    /* Contains session data including
PARAMETER               /* field #DSNAME
1 #ERROR-CODE(N3)
1 #ERROR-NR  (N4)
1 #ERROR-PARM(A75)
1 #RECALL-STATUS (N1)
1 #OPTIONS   (A20)      /* I/O
END-DEFINE
*
RESET #RECALL-STATUS
END
↩
```

## Editor Profile Exit ISP-ED-U

When activated, this exit is invoked before an edit session is opened in Natural ISPF. The exit can change the list of profile names requested for this edit session.

To invoke this exit, you must enter (PROFIL in the CONFIG member.

In addition to the standard **data parameters** described above, you must define the following data in the exit:

| Parameter | Format | Type | Meaning |
|---|---|---|---|
| #SES-DATA | (A200) | I | Session data for object. |
| #PROFILE-NAME | (A8/10) | I/O | Profile names. |

### Example

```
DEFINE DATA PARAMETER
* ********************* **    ************************************
1 #OBJECT   (A2)       /* I     Object type for the session
1 #FUNCTION (A2)       /* I     Function entered by user
1 #SES-DATA (A200)     /* I     Session data for object
1 #PROFILE-NAME(A8/10) /* I/O   Profile names
END-DEFINE
*
END
```

## Incore Database Defaults Exit IDB-USRN

This exit is invoked whenever the EDIT/BROWSE function is invoked in the Incore database. It can be used to define application-specific defaults and return language-dependent data.

Before modifying the exit, copy it to the application library which uses the Incore database, then make the changes and recompile it.

| Parameter | Format | Type | Meaning |
|---|---|---|---|
| #LANGUAGE | (A1) | I/O | Requested language * |
| #DEFAULT-SCROLL | (A4) | O | Scroll value MOVE 'CSR' TO #DEFAULT-SCROLL |
| **Language-dependent constants:** | | | |
| #-ROW | (A4) | O | Row MOVE 'Row' TO #-ROW |
| #-OF | (A4) | O | Of MOVE 'of' TO #-OF |
| #-COMMAND | (A10) | O | Command MOVE 'COMMAND==>' TO #-COMMAND |
| #-SCROLL | (A9) | O | Scroll MOVE 'SCROLL==>' TO #-SCROLL |
| **PF keys:** | | | |
| #PF-KEY | (A50/24) | O | Contents ASSIGN #PF-KEY(1)=':I' |
| #PF-NAME | (A5/24) | O | Language dependent name ASSIGN #PF-NAME(1)='*Insrt*' |

* Possible values for #LANGUAGE when used as an output parameter:

- The value 'H' indicates that additional calls to Natural's text translation module NATPM are to be made to support inverted terminal display for Middle Eastern countries during the functions

EDIT and BROWSE. When #LANGUAGE='H' is specified, the NATPM module must be specified as CSTATIC in your Natural parameter module.

- Any character other than H causes the screen to be displayed in the normal mode, without invoking the NATPM module.

## Container File Access Exit IDBC---U

This exit is invoked whenever the Incore database container file is accessed. It can be used to restrict access to the container file

Before modifying the exit, copy it to the application library which uses the Incore database container file. You can then make the changes and recompile the exit.

| Parameter | Format | Type | Meaning | |
|---|---|---|---|---|
| #ACTION | (A8) | I | Access type for incore file. Valid actions: | |
| | | | ' ' | Directory from container. |
| | | | DELETE | Delete file from container. |
| | | | RETRIEVE | Read from container. |
| | | | STORE | Write to container. |
| T#YPE | (A8) | I | Identification of container file consists of the #TYPE, #GROUP and #NAME fields. | |
| #GROUP | (A48) | I | | |
| #NAME | (A32) | I | | |
| #ERROR-CODE | (N3) | O | Access denied if this field is anything but zero. | |
| #ERROR-TEXT | (A75) | O | Optional message text. | |

# 7 Buffer Pool And Recovery Files


- Buffer Pool Maintenance ..... 120
- Optimizing the Buffer Pool ..... 120
- Buffer Pool Files ..... 120
- Recovery Files ..... 121
- Troubleshooting ..... 122

This chapter describes maintenance functions you can perform on the Editor buffer pool and user recovery files. These refer to the BPSTAT, BP FILES and BP RECS options on the Administrator Menu.

The following description is not intended to give full technical details on the buffer pool, but provides a rough outline of its function.

The buffer pool software enables you to allocate space to which users perform read and write operations.

The buffer pool consists of area in the memory and a container file. The area is divided into memory blocks of fixed length. The buffer pool manager reads blocks from the memory and writes blocks to it. If the buffer pool is full, it tries to release other blocks by writing them to the container file. In the normal case, however, most blocks in use will be in the memory.

## Buffer Pool Maintenance

The BPSTAT option on the Administrator Menu (or the BPSTAT session command) invokes the Natural utility SYSEDT with the General Information function (SYSEDT Function Code G).

The General Information screen provides an overview of the current status of the editor buffer pool.

For further information, see the description of the SYSEDT utility in the Natural documentation.

## Optimizing the Buffer Pool

Optimizing the buffer pool means modifying certain parameters to increase buffer pool efficiency. To modify a parameter, use the Natural utility SYSEDT or refer to the section *Installing the Software AG Editor* in the Natural *Installation* documentation.

## Buffer Pool Files

The BP FILES option on the Administrator Menu invokes the Natural utility SYSEDT with the Logical Files function (SYSEDT Function Code F).

The Logical Files screen provides information such as file number, user ID, logical file type, pool blocks, file records, last access.

In addition, the first column contains an input field labeled M for Mark. If you enter a question mark (?) in this field, a window is displayed, which shows you the codes that are valid input for this field.

Enter the code in the Mark field to perform functions such as selecting the logical file of a user, deleting a logical file.

**Note:** It is recommended that you delete only *unused* sessions. If you delete a buffer pool file for an Editor session that is in use, the results of the Editor session are unpredictable. (Typically, the user will receive the message: Session does not exist).

You can invoke the above display directly using a function command from a Natural ISPF session. You must address the object type BPF in the command syntax:

```
LIST BPF
```

For further information, see the description of the SYSEDT utility in the Natural documentation.

## Recovery Files

Recovery files are checkpoints which are written during an edit session and deleted when the edit session is terminated normally. If the edit session is terminated abnormally, the checkpoint files are stored in the Editor work file indefinitely or until the user selects the file for recovery.

The BP RECS option on the Administrator Menu invokes the Natural utility SYSEDT with the Logical Files function (SYSEDT Function Code R).

The Recovery Files screen provides the information such as Natural user ID, library member name, library name, library type, number of recovery records per recovery file, creation date and time of the recovery file.

For each recovery file listed, you can execute a line command, for example, to select the recovery file of this user, or to delete the recovery file. You enter a line command in the C column, next to the user required.

To invoke a list of recovery files using a function command from a Natural ISPF session, address the object type BPR in the command syntax:

```
LIST BPR
```

## Troubleshooting

This subsection lists messages that may appear in the message line of a Natural ISPF screen concerning the Editor buffer pool. Related messages are grouped together, and corrective action is suggested.


- Installation Errors
- Editor Work File, Editor Auxiliary Buffer Pool and Timeout Parameters
- Internal ISPF/Editor Problems
- I/O Problems on Editor Work File or Volume
- Other Inconsistencies


### Installation Errors

**Message:**

```
SSIZE too small
```

**Explanation / Action:**

Modify your `NATPARM` module using `SSIZE=64`.

**Message:**

```
DBID of Natural PROCESS is missing
```

**Explanation / Action:**

The DBID of Entire System Server (formerly Natural Process) is missing in the Natural ISPF configuration. See the section ***System Configuration***.

**Messages:**

```
Checkpoint file not active'
-caution-  profile changed to "recovery off" (from "recovery on") because checkpoint ↩
file not active' (in Editor session)
```

**Explanation / Action:**

The Editor work file was not formatted correctly, no space was reserved for the checkpoint file. Check your job that allocates the work file.

**Message:**

```
NSPF FAILURE IN OPEN BUFFERS
```

**Explanation / Action:**

The allocation of the Editor area failed or was not sufficient (`SSIZE` buffer). Check the `SSIZE` parameter. If the parameter has a value of at least 64, you can either increase the `MSIZE` parameter, or decrease the size value of other buffers you do not need, for example `CSIZE`, `FSIZE` or `TSIZE` (see the Natural *Parameter Reference* documentation). In some environments, you can use the Natural command `SYSBUS` to check the allocations in effect.

**Message:**

```
BP not active
```

**Explanation / Action:**

The buffer pool was not initialized correctly. See the section *Installing the Software AG Editor* in the Natural *Installation* documentation.

### Editor Work File, Editor Auxiliary Buffer Pool and Timeout Parameters

**Messages:**

```
Checkpoint of file failed
No space for data block
No space for index block
Cannot allocate BP file
-caution-  profile changed to "recovery off" (from "recovery on") because checkpoint ↩
file is full (in Editor session)
-caution-  profile changed to "log off" (from "log on") because log file is full ↩
(undo command not active) (in Editor session)
Buffer pool is full
Cannot allocate BP block
Cannot allocate checkpoint block
Suspending of a session failed
```

**Explanation / Action:**

If you get one of these messages your current activity might not be finished correctly. You should leave N-ISPF and finish the Natural session.

These messages indicate one of the following reasons:

- The Software AG Editor work file is too small. You must increase the work file size.

  **Note:** The record length of data sets or PDS members, which will be edited with Natural ISPF, cannot be larger as the record length of this editor work file.

- The Software AG Editor auxiliary buffer pool defined by the profile parameter `EDPSIZE` is too small. You must increase the `EDPSIZE`.
- The timeout parameter values are too large, causing data to be kept in the buffer pool too long. You must decrease the timeout parameter values.

**Message:**

```
Activation of a session failed
```

**Explanation / Action:**

The session has been deleted due to a buffer pool timeout. Restart the session.

### Internal ISPF/Editor Problems

The following messages are not usually displayed. They appear in the rare case of some internal problem.

**Messages:**

```
Some data may be ignored
Session does not exist
Cannot free BP block
Log error
```

**Explanation / Action:**

Contact your Software AG support representative.

### I/O Problems on Editor Work File or Volume

**Messages:**

```
I/O ERROR
Cannot read index block
Write to BP failed
Read from BP failed
Read failed
Error reading log block
Cannot read checkpoint block
Cannot write checkpoint block
Recovery of file failed
```

**Explanation / Action:**

Check the work file. If the error persists, reallocate the file on another volume.

### Other Inconsistencies

**Message:**

```
Error in getting lines
```

**Explanation / Action:**

The Natural source area is corrupted, for example if Natural ISPF is aborted by %% and the source area was modified.

**Message:**

```
Getting text failed
```

**Explanation / Action:**

The Editor message table is inconsistent.

**Message:**

```
Invalid printer reference number
```

**Explanation / Action:**

The requested printer has no corresponding definition in the `NATPARM` module. Note that with `PRINTER=OFF` in the `NATPARM` module, the `WORKPOOL`, `BROWSE-VIEW` and `COMPARE` functionality is disabled.

# 8 Natural ISPF Libraries

This section lists all Natural ISPF libraries together with a description of their contents. In the description of installation and administration functions, libraries are referred to by their description.

| Library Name | Description |
|---|---|
| SYSISPS1 | ISPF system tables / menus / profiles |
| SYSISPFU | ISPF user tables / menus / profiles |
| SYSISPH1 | ISPF system help texts |
| SYSISPDB | ISPF Incore database modules |
| SYSISPX | ISPF exit sources |
| SYSISPXC | ISPF user exits for Com-plete. |
| SYSISPE | ISPF example library |
| SYSISPI | ISPF system interface library |
| SYSISPXU | ISPF user-defined exit sources |
| SYSISPHU | ISPF user-defined help texts |
| SYSISPIU | ISPF user-defined information (UINFO) |

# 9 Natural ISPF Special Characters

Some special characters used in Natural ISPF depend on the keyboard used in your country or at your installation. The table below lists the special characters by their hexadecimal values and the characters to which they correspond on some national keyboards:

| Hexadecimal | English Keyboard | German Keyboard |
|---|---|---|
| 50 | & | & |
| 61 | / | / |
| 4F | \| | ! |
| 4A | ¢ | Ä |
| 5A | ! | Ü |
| 5B | $ | $ |
| 5E | ; | ; |
| 5F | : | ^ |
| 6B | , | , |
| 7B | # | # |
| 7C | @ | § |
| 7E | = | = |

# 10 Authorization Classes

This section lists the available authorization classes and the Natural ISPF objects they refer to.

- Authorization classes as they appear in the `Class` column of the table are assigned authorization levels in user definitions.
- The codes in the Code column of the table are used in menu lines in menu definitions, as well as in the site control table. The site control table is described in the section *Open NSPF* in the *Natural ISPF Programmer's Guide*.

| Code | Class | Natural ISPF Objects |
|---|---|---|
| N | Natural programming | Natural objects, views and error messages |
| P | PDS maintenance | PDS members and previous versions |
| D | Data set maintenance | z/OS data sets and volumes |
| J | SYSOUTS | z/OS jobs and SYSOUT files |
| S | System info | System operations |
| 3 | Active jobs | Active jobs (z/OS and z/VSE) |
| 9 | Operator commands | Use of operator commands |
| A | NSPF Administrator | Configuration operations |
| T | CA Panvalet | CA Panvalet members and previous versions |
| L | CA Librarian | CA Librarian members and versions |
| 1 | z/VSE files | z/VSE files, volumes and members |
| 2 | z/VSE SYSOUT | SYSOUT of z/VSE jobs |
| B | BS2000 files | BS2000 files |
| E | BS2000 LMS elements | LMS library elements and previous versions |
| 4 | BS2000 jobs & job var. | BS2000 jobs and job variables |
| = | USER defined | Site-specific objects and menu options |

# 11 Subsystems Supported By Natural ISPF

This section lists all available subsystems supported by Natural ISPF.

- The subsystem abbreviation preceded by the plus sign (+) is used when activating the subsystem for your site in the `CONFIG` member.
- The subsystem abbreviation shown in the right-hand column of the table is used in menu lines in menu definition, and in the Site Control Table when defining new objects or commands using the Open NSPF facility.

| Subsystem in CONFIG | Abbreviation in CONFIG member | Abbreviation |
|---|---|---|
| Natural | +N | N |
| OS/390 | +M | M |
| z/VSE | +D | D |
| BS2000 | +B | B |
| CA Librarian | +L | L |
| CA Panvalet | +P | P |
| Con-nect * | +C | C |
| SAT * (System Automation Tools) | +S | S |
| Incore Database | +O | O |

* See following note:

**Note:** Con-nect (Software AG's office management system) is a separate product of Software AG. SAT (System Automation Tools) is a technical product delivered together with all of the following products: Entire Operations, Entire Output Management, Entire Event Management and Mainframe Navigation. It is recommended that the subsystems C and S be activated if the related products are installed in your system environment. The functions made available by activating subsystem C are described in the subsection *Con-nect Documents* in section *Common Objects* of the *Natural ISPF User's Guide*. For a description of the functions

related to subsystem S, see the subsection *Define SAT, Natural, Product Parameters* in the *System Automation Tools* documentation.

See also:

- Entire Operations *Installation and Operations* documentation
- Entire Output Management *Installation and Customization* documentation
- Entire Event Management *Installation and Customization on Mainframes* documentation and
- Mainframe Navigation *Installation* documentation.