



# natural

---

Version 4.1.2 for Mainframes | Utilities

This document applies to Natural Version 4.1.2 for Mainframes and to all subsequent releases.

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

© Copyright Software AG 1979 - 2003.  
All rights reserved.

The name Software AG and/or all Software AG product names are either trademarks or registered trademarks of Software AG. Other company and product names mentioned herein may be trademarks of their respective owners.

# Table of Contents

<Untitled>	1
Natural Utilities - Overview	1
Utility Activation	1
Utility List	1
Utilities Grouped by Function	3
<b>ADACALL - Issuing Adabas Direct Calls</b>	4
ADACALL - Issuing Adabas Direct Calls	4
Invoking ADACALL	4
ADACALL Parameters	5
Adabas OP Command	6
ADACALL Commands and PF Keys	6
User Exit ADAEXIT	8
<b>DBLOG Utility - Logging Database Calls</b>	9
DBLOG Utility - Logging Database Calls	9
<b>Executing DBLOG</b>	10
Executing DBLOG	10
Data Processing and Storage	10
Activating and Deactivating DBLOG	10
Using Selective DBLOG	11
<b>DBLOG Menu</b>	12
DBLOG Menu	12
DBLOG Menu - Functions	12
DBLOG Menu - Specifying Restrictions	13
Specifying Buffers	14
<b>DBLOG Trace Screen</b>	15
DBLOG Trace Screen	15
DBLOG Trace Screen - Adabas Commands	15
Invoking DBLOG Trace - Adabas Commands	15
DBLOG Trace Adabas Commands - Fields and Functions	15
Displaying Buffers on the DBLOG Trace Screen	16
DBLOG Trace Screen - DL/I Calls	18
Invoking DBLOG Trace - DL/L Calls	18
DBLOG Trace DL/I Calls - Fields and Functions	19
DBLOG Trace Screen - SQL Statements	19
Invoking DBLOG Trace - SQL Statements	19
DBLOG Trace SQL Statements - Fields and Functions	21
<b>DBLOG - Snapshot Function</b>	23
DBLOG - Snapshot Function	23
Snapshot Function - Adabas Commands	23
Invoking Snapshot - Adabas Commands	23
Displaying Buffers on the Snapshot Report	24
Snapshot Function - DL/I Calls	24
Invoking Snapshot - DL/I Calls	25
Snapshot Report Information - DL/I Calls	25
Snapshot Function - SQL Statements	26
Invoking Snapshot - SQL Statements	26
Snapshot Report Information - SQL Statements	27
<b>DBLOG Direct Commands</b>	28
DBLOG Direct Commands	28
Syntax Diagrams	28
Parameters	28
<b>Natural Debugger - Overview</b>	30
Natural Debugger - Overview	30

<b>Concepts of the Natural Debugger</b>	31
Concepts of the Natural Debugger	31
Debug Entries/Spies	31
Debug Window	32
<b>Start the Natural Debugger</b>	33
Start the Natural Debugger	33
Operation Requirements	33
Batch Processing	33
Invoke the Natural Debugger	33
Default Object	34
<b>Set Test Mode ON/OFF</b>	35
Set Test Mode ON/OFF	35
<b>Debug Environment Maintenance</b>	36
Debug Environment Maintenance	36
Load Debug Environment	36
Save Debug Environment	36
Reset Debug Environment	37
Delete Debug Environment	37
<b>Spy Maintenance</b>	38
Spy Maintenance	38
Activate Spy	38
Deactivate Spy	38
Delete Spy	38
Display Spy	39
Modify Spy	39
<b>Breakpoint Maintenance</b>	40
Breakpoint Maintenance	40
Conditions of Use	40
Activate Breakpoint	41
Deactivate Breakpoint	41
Delete Breakpoint	41
Display Breakpoint	41
Modify Breakpoint	42
Display/Modify Breakpoint Screen	42
Set Breakpoint	43
<b>Watchpoint Maintenance</b>	45
Watchpoint Maintenance	45
Invoke Watchpoint Maintenance	45
Activate Watchpoint	46
Deactivate Watchpoint	46
Delete Watchpoint	46
Display Watchpoint	46
Modify Watchpoint	47
Display/Modify Watchpoint Screen	47
Set Watchpoint	48
Watchpoint Operators	49
<b>Call Statistics</b>	51
Call Statistics	51
Set Call Statistics ON/OFF	51
Display All Objects	51
Display Called Objects	52
Display Non-Called Objects	52
Print Objects	52
Print Options	53
<b>Statement Execution Statistics</b>	54
Statement Execution Statistics	54

Set Statement Execution Statistics ON/OFF/COUNT . . . . .	54
Setup Options . . . . .	54
Activate/deactivate Statistics . . . . .	55
Invoke Statement Execution Statistics . . . . .	55
Delete Statement Execution Statistics . . . . .	56
Display Statement Execution Statistics . . . . .	56
Display All Statement Lines . . . . .	57
Display Executed Statement Lines . . . . .	58
Display Non-Executed Statement Lines . . . . .	58
Print Statements . . . . .	58
Print Options . . . . .	58
<b>Variable Maintenance . . . . .</b>	<b>59</b>
Variable Maintenance . . . . .	59
Display User-defined, Global and DB-related System Variables . . . . .	59
Display Variables - Summary . . . . .	59
Display Variable - Individual . . . . .	59
Display System Variables . . . . .	60
Modify Variables . . . . .	60
<b>List Object Source . . . . .</b>	<b>62</b>
List Object Source . . . . .	62
Maintain Breakpoints . . . . .	62
<b>Execution Control Commands . . . . .</b>	<b>64</b>
Execution Control Commands . . . . .	64
ESCAPE BOTTOM . . . . .	64
ESCAPE ROUTINE . . . . .	64
EXIT . . . . .	64
GO . . . . .	65
NEXT . . . . .	65
RUN . . . . .	65
STEP . . . . .	65
STEP SKIPSUBLEVEL . . . . .	65
STOP . . . . .	65
<b>Navigation and Information Commands . . . . .</b>	<b>66</b>
Navigation and Information Commands . . . . .	66
BREAK . . . . .	66
FLIP . . . . .	66
LAST . . . . .	66
OBJCHAIN . . . . .	66
ON/OFF . . . . .	66
PROFILE . . . . .	67
Edit Profile Screen . . . . .	67
SCAN . . . . .	67
SCREEN . . . . .	67
SET OBJECT . . . . .	68
STACK . . . . .	68
SYSVARS . . . . .	68
TEST ON/OFF . . . . .	68
<b>Command Summary and Syntax . . . . .</b>	<b>69</b>
Command Summary and Syntax . . . . .	69
All Debug Commands . . . . .	69
Syntax Diagrams . . . . .	73
ACTIVATE . . . . .	74
DEACTIVATE . . . . .	74
DELETE . . . . .	74
DISPLAY . . . . .	75
LIST . . . . .	75

LOAD . . . . .	76
MODIFY . . . . .	76
PRINT . . . . .	76
RESET . . . . .	76
SAVE . . . . .	76
SET . . . . .	77
<b>INPL Utility - Overview</b> . . . . .	78
INPL Utility - Overview . . . . .	78
<b>INPL Utility - Introduction</b> . . . . .	79
INPL Utility - Introduction . . . . .	79
Restrictions . . . . .	79
Special Cases . . . . .	79
Invoking INPL . . . . .	79
Options Available . . . . .	80
INPL Report . . . . .	81
<b>INPL - Load Library Only</b> . . . . .	82
INPL - Load Library Only . . . . .	82
<b>INPL - Load DDMs Only</b> . . . . .	83
INPL - Load DDMs Only . . . . .	83
<b>INPL - Load Error Messages Only</b> . . . . .	84
INPL - Load Error Messages Only . . . . .	84
<b>INPL - Load All Objects</b> . . . . .	85
INPL - Load All Objects . . . . .	85
<b>INPL - Scan INPL File</b> . . . . .	86
INPL - Scan INPL File . . . . .	86
<b>INPL - Natural Security Recover</b> . . . . .	87
INPL - Natural Security Recover . . . . .	87
Reset Environment . . . . .	87
Remove Owners . . . . .	87
<b>NATPAGE Utility - Screen Capturing</b> . . . . .	88
NATPAGE Utility - Screen Capturing . . . . .	88
<b>NATRJE - Natural Remote Job Entry</b> . . . . .	89
NATRJE - Natural Remote Job Entry . . . . .	89
NATRJE General Information . . . . .	89
Calling NATRJE from a Natural Program . . . . .	90
Invoking NATRJE . . . . .	90
Example Programs . . . . .	91
NATRJE Return Codes . . . . .	93
Return Codes Common to all Environments . . . . .	93
Additional Return Codes for VSE/ESA . . . . .	93
Additional Return Codes for BS2000/OSD . . . . .	94
NATRJE User Exit . . . . .	94
NATRJE Features Applicable to UTM/TIAM . . . . .	95
Additional Values for the parm3 Parameter . . . . .	95
Name of BS2000/OSD Dataset . . . . .	95
<b>NATUNLD/NATLOAD Utilities - Overview</b> . . . . .	96
NATUNLD/NATLOAD Utilities - Overview . . . . .	96
<b>NATUNLD/NATLOAD - Introduction</b> . . . . .	97
NATUNLD/NATLOAD - Introduction . . . . .	97
General Information . . . . .	97
Invoking NATUNLD or NATLOAD . . . . .	98
Options and Specifications . . . . .	98
Names and Ranges . . . . .	98
Dates and Ranges . . . . .	98
File Assignments . . . . .	99

- NATUNLD Utility** . . . . . 100
  - NATUNLD Utility . . . . . 100
    - Invoking NATUNLD . . . . . 100
    - Programming Objects . . . . . 100
      - Functions for Programming Objects . . . . . 100
      - Parameters for Programming Objects . . . . . 101
    - DDMs . . . . . 103
      - Functions for DDMs . . . . . 104
      - Parameters for DDMs . . . . . 104
    - Error Messages . . . . . 104
      - Functions for Error Messages . . . . . 105
      - Parameters for Error Messages . . . . . 105
    - Delete Instructions for Programming Objects . . . . . 106
      - Functions for Delete Instructions - Programming Objects . . . . . 106
      - Parameters for Delete Instructions - Programming Objects . . . . . 107
    - Delete Instructions for DDMs . . . . . 107
      - Functions for Delete Instructions - DDMs . . . . . 108
      - Parameters for Delete Instructions - DDMs . . . . . 108
    - Delete Instructions for Error Messages . . . . . 108
      - Functions for Delete Instructions - Error Messages . . . . . 108
      - Parameters for Delete Instructions - Error Messages . . . . . 109
    - Executing NATUNLD in Batch Mode . . . . . 109
      - Condition Codes and User Exits in Batch Mode . . . . . 110
    - NATUNLD Report . . . . . 111
- NATLOAD Utility** . . . . . 112
  - NATLOAD Utility . . . . . 112
    - Invoking NATLOAD . . . . . 112
    - Functions . . . . . 112
      - Scan . . . . . 112
      - Load . . . . . 113
    - Parameters . . . . . 113
    - Verifying Subroutines and Classes . . . . . 117
      - Subroutine Names . . . . . 117
      - Class Names and GUIDs . . . . . 117
    - Executing NATLOAD in Batch Mode . . . . . 117
      - Condition Codes and User Exits in Batch Mode . . . . . 118
    - NATLOAD Report . . . . . 118
- NATUNLD/NATLOAD - Direct Commands** . . . . . 120
  - NATUNLD/NATLOAD - Direct Commands . . . . . 120
- NATUNLD Direct Command Syntax** . . . . . 121
  - NATUNLD Direct Command Syntax . . . . . 121
    - General Information . . . . . 121
    - NATUNLD Syntax Diagrams . . . . . 121
      - unload-objects-clause - for Programming Objects . . . . . 122
      - unload-ddms-clause - for DDMs . . . . . 122
      - unload-messages-clause - for Error Messages . . . . . 122
      - object-where-clause - for Programming Objects and Error Messages . . . . . 123
      - object-with-clause - for Programming Objects . . . . . 123
      - ddm-where-clause . . . . . 123
      - ddm-with-clause . . . . . 123
      - message-with-clause . . . . . 123
      - delete-objects-clause - Delete Instructions for Programming Objects . . . . . 124
      - delete-ddms-clause - Delete Instructions for DDMs . . . . . 124
      - delete-messages-clause - Delete Instructions for Error Messages . . . . . 124
    - Examples of Direct Commands for NATUNLD in Batch Mode . . . . . 124

<b>NATLOAD Direct Command Syntax</b>	126
NATLOAD Direct Command Syntax	126
General Information	126
NATLOAD Syntax Diagrams	126
Loading/Scanning the Entire Work File	127
Loading/Scanning Programming Objects only	127
Loading/Scanning DDMs only	127
Loading/Scanning Error Messages only	127
Loading/Scanning Programming Objects, DDMs and Error Messages	127
object-where-clause	128
object-with-clause	128
ddm-where-clause	128
ddm-with-clause	129
message-where-clause	129
message-with-clause	129
Examples of Direct Commands for NATLOAD in Batch Mode	129
<b>NATUNLD/NATLOAD Commands for Reports</b>	131
NATUNLD/NATLOAD Commands for Reports	131
<b>Object Handler - Overview</b>	132
Object Handler - Overview	132
<b>Object Handler - General Information</b>	133
Object Handler - General Information	133
Principles of Object Transfer	133
Invoking the Object Handler	134
Batch or Direct Command Calls	134
Batch Mode	134
Online Mode	135
Issuing Object Handler Commands from a Natural Program	135
Text Member for Reports, Restarts and Traces	135
Natural Security	135
Standard PF Keys	136
<b>Object Handler - Functions - Overview</b>	137
Object Handler - Functions - Overview	137
<b>Object Handler Wizards</b>	138
Object Handler Wizards	138
Step 1 - Start the Procedure	138
Step 2 - Unload/Load/Scan Objects into/from Work Files	139
Step 3 - Set Parameters	141
Step 4 - Select Objects	141
Step 5 - Execute Processing	142
Step 6 - Continue Processing	142
<b>Object Handler - Advanced User</b>	144
Object Handler - Advanced User	144
Activating Advanced User	144
Processing Objects	144
<b>Object Handler - View</b>	145
Object Handler - View	145
Natural Library Objects	145
Natural System Error Messages	145
Natural Command Processor Sources	146
FDTs	146
Natural-related Objects	146
DDMs	147
Columns and Commands	147
<b>Object Handler - Find</b>	148
Object Handler - Find	148

- Object Handler - Administration** . . . . . 149
  - Object Handler - Administration . . . . . 149
    - List the available Workplans in the Workplan Library . . . . . 149
      - Columns and Commands . . . . . 149
    - Create a new Workplan . . . . . 151
    - Change the Workplan Library . . . . . 151
- Object Handler - System File Selection** . . . . . 152
  - Object Handler - System File Selection . . . . . 152
- Object Handler - Library Selection** . . . . . 153
  - Object Handler - Library Selection . . . . . 153
- Natural System Error Message Selection** . . . . . 154
  - Natural System Error Message Selection . . . . . 154
    - Columns and Commands . . . . . 154
- Object Handler - Object Selection** . . . . . 156
  - Object Handler - Object Selection . . . . . 156
    - Columns and Commands . . . . . 156
- Object Handler - Object Specification** . . . . . 158
  - Object Handler - Object Specification . . . . . 158
    - All Objects on the Work File . . . . . 158
    - Natural Library Objects . . . . . 158
      - Natural Library Object Details . . . . . 159
      - Natural Library Object Properties . . . . . 161
      - Natural Library Object Exceptions . . . . . 162
      - Natural Library Object Exception Properties . . . . . 163
    - Natural System Error Messages . . . . . 164
      - Natural System Error Message Details . . . . . 164
      - Natural System Error Message Exceptions . . . . . 165
    - Natural Command Processor Sources . . . . . 165
      - Natural Command Processor Source Exceptions . . . . . 166
    - Natural-related Objects . . . . . 166
      - Natural Profiles . . . . . 166
        - Natural Profile Exceptions . . . . . 167
      - Natural Debug Environments . . . . . 167
        - Natural Debug Environment Exceptions . . . . . 168
      - Natural DL/I Subfiles . . . . . 168
        - Natural DL/I Subfile Exceptions . . . . . 169
    - DDMs . . . . . 169
      - DDM Properties . . . . . 170
      - DDM Exceptions . . . . . 171
    - FDTs . . . . . 172
    - Use Selection or List Workplan . . . . . 173
  - Object Handler - Settings** . . . . . 174
    - Object Handler - Settings . . . . . 174
      - Settings Screen Fields . . . . . 174
      - Set Additional Options . . . . . 175
        - Work File and Report Options . . . . . 176
        - XREF Options . . . . . 177
        - Version Check . . . . . 178
        - Transfer Options . . . . . 178
        - Replace Options . . . . . 179
        - Number to Process . . . . . 180
        - FDIC Settings . . . . . 180
        - FSEC Settings . . . . . 180
      - Set Global Parameters . . . . . 180
        - Rules for New Values . . . . . 183

<b>Object Handler - Workplans</b>	184
Object Handler - Workplans	184
Contents of Workplans	184
Examples of Workplans	185
Referencing Workplans	185
Keyword Explanation	186
<b>Name, Date and Time Specification</b>	187
Name, Date and Time Specification	187
Name	187
Date	188
Time	189
<b>Object Handler - Work Files</b>	190
Object Handler - Work Files	190
Work File Assignment	190
Work File Format	190
Internal Format	190
Transfer Format	191
<b>Object Handler - Commands - Overview</b>	192
Object Handler - Commands - Overview	192
<b>Direct and PROCEDURE Workplan Commands</b>	193
Direct and PROCEDURE Workplan Commands	193
<b>Direct and PROCEDURE Workplan Syntax</b>	195
Direct and PROCEDURE Workplan Syntax	195
<b>Object Handler - Select-Clause</b>	196
Object Handler - Select-Clause	196
SELECTION Workplan	196
Natural Object and DDM Selection	197
Syntax	197
Natural-related Debug Environment Selection	202
Syntax	202
Natural-related Profile Selection	203
Syntax	203
Natural-related DL/I Subfile Selection	204
Syntax	204
Natural System Error Selection	206
Syntax	206
Natural Command Processor Selection	207
Syntax	207
FDT Selection	208
Syntax	208
<b>Object Handler - Object List</b>	210
Object Handler - Object List	210
Syntax of Object-type-and-location	210
Syntax of Object-Name-Description	211
<b>Object Handler - Parameter-Setting</b>	214
Object Handler - Parameter-Setting	214
Syntax of Parameter-Setting	214
PARAMETER-Clause	215
Keyword Explanation - Parameter-Setting	216
<b>Object Handler - Option-Setting</b>	218
Object Handler - Option-Setting	218
Syntax of Option-Setting	218
OPTION-Clause	219
Keyword Explanation - Option-Setting	220
XREF	222

- Object Handler Commands for CUIs** . . . . . 223
  - Object Handler Commands for CUIs . . . . . 223
- Batch Condition Codes and User Exit Routines** . . . . . 226
  - Batch Condition Codes and User Exit Routines . . . . . 226
- Object Handler - Tools** . . . . . 227
  - Object Handler - Tools . . . . . 227
    - Status . . . . . 227
    - Last Result . . . . . 227
    - Traces . . . . . 227
    - Reports . . . . . 227
- SYSOBJH Profile** . . . . . 228
  - SYSOBJH Profile . . . . . 228
    - Text Object PROFILE . . . . . 228
- Natural Recording Utility** . . . . . 230
  - Natural Recording Utility . . . . . 230
    - Purpose of Recording . . . . . 230
    - Data and Functions Recorded . . . . . 230
    - Recording a Session . . . . . 231
      - Specifying Libraries . . . . . 231
      - Activating a Recording . . . . . 231
      - Deactivating a Recording . . . . . 231
    - Playing Back a Recording . . . . . 232
      - Step Mode and Background Mode . . . . . 232
      - Activating a Playback . . . . . 232
      - Interrupting a Playback . . . . . 233
    - Manipulating a Recording . . . . . 233
- SYSBPM Utility - Buffer Pool Management** . . . . . 234
  - SYSBPM Utility - Buffer Pool Management . . . . . 234
- Invoking and Operating SYSBPM** . . . . . 235
  - Invoking and Operating SYSBPM . . . . . 235
    - Invoking SYSBPM . . . . . 235
    - Online Help . . . . . 236
    - SYSBPM Main Menu - Fields and Functions . . . . . 236
      - Fields . . . . . 236
      - Functions . . . . . 237
    - SYSBPM in a Sysplex Environment . . . . . 238
- SYSBPM - List Objects** . . . . . 239
  - SYSBPM - List Objects . . . . . 239
    - Invoking List Objects . . . . . 239
    - Columns and Selection Options . . . . . 240
      - Name and Range Specification . . . . . 244
      - Date Specification . . . . . 245
      - Time Specification . . . . . 246
    - Screen Navigation . . . . . 247
    - Line Commands . . . . . 248
- SYSBPM - Delete Objects** . . . . . 249
  - SYSBPM - Delete Objects . . . . . 249
- SYSBPM - Directory Information** . . . . . 250
  - SYSBPM - Directory Information . . . . . 250
    - Fields for Buffer Pool Objects . . . . . 250
    - Fields for BP Cache Objects . . . . . 251
    - Commands . . . . . 252
- SYSBPM - Hexadecimal Display** . . . . . 254
  - SYSBPM - Hexadecimal Display . . . . . 254
    - Commands . . . . . 254

<b>SYSBPM - Write to Work File</b>	255
SYSBPM - Write to Work File	255
<b>SYSBPM - Display Sorted Extract</b>	256
SYSBPM - Display Sorted Extract	256
<b>SYSBPM - Buffer Pool Statistics</b>	257
SYSBPM - Buffer Pool Statistics	257
General Buffer Pool Statistics	257
Buffer Pool Load/Locate Statistics	259
Buffer Pool Fragmentation	262
Internal Function Usage	263
Buffer Pool Hash Table Statistics	263
<b>SYSBPM - BP Cache Statistics</b>	266
SYSBPM - BP Cache Statistics	266
General BP Cache Statistics	266
BP Cache Call Statistics	268
BP Cache Hash Table Statistics	269
<b>SYSBPM - Select Buffer Pool</b>	271
SYSBPM - Select Buffer Pool	271
Invoking Select Buffer Pool	271
Display Buffer Pools	271
Reset Buffer Pool	272
<b>SYSBPM - Blacklist Maintenance</b>	273
SYSBPM - Blacklist Maintenance	273
Maintain Blacklist	273
Adding Objects	274
Modifying Objects	274
Deleting Objects	274
List Object Sets	275
Edit Object Set	275
Creating Object Sets	276
Modifying Object Sets	276
Add Object Set to Blacklist	277
Delete Object Set from Blacklist	277
Delete Object Set Text Member	278
Blacklist Maintenance in Batch Mode	278
Explanation of Syntax	278
<b>Examples of Input</b>	279
<b>SYSBPM - Preload List Maintenance</b>	281
SYSBPM - Preload List Maintenance	281
List Preload Lists	281
Edit Preload List	282
Creating Preload Lists	282
Modifying Preload Lists	282
Generate Preload List from Buffer Pool	283
Delete Preload List	284
<b>SYSBPM Direct Commands</b>	285
SYSBPM Direct Commands	285
SORT	292
SORT BPC	292
<b>SYSBPM - Batch Processing</b>	293
SYSBPM - Batch Processing	293
Related Topics	293
<b>SYSBPM - Appl. Programming Interfaces</b>	295
SYSBPM - Appl. Programming Interfaces	295
<b>SYSDDM Utility</b>	296
SYSDDM Utility	296

- DDMs . . . . . 296
- SYSDDM and Predict . . . . . 296
- Invoking SYSDDM . . . . . 296
  - Overview of Functions . . . . . 297
- Edit DDM . . . . . 299
  - DDM Editor . . . . . 300
  - Field Attributes . . . . . 301
  - DDM Editor Commands . . . . . 303
  - Extended Field Editing . . . . . 305
- Read DDM . . . . . 306
- Catalog DDM . . . . . 306
- Delete DDM . . . . . 306
- List DDMs . . . . . 306
- List DDMs with Additional Information . . . . . 306
- Show Defined DBIDs and Used FNRs . . . . . 307
  - Database IDs Defined in Natural . . . . . 307
  - File Numbers of Existing DDMs for a Database . . . . . 307
- Copy DDM to Another FDIC File . . . . . 307
- Generate DDM from Adabas FDT . . . . . 307
- SYSEDT Utility - Editor Buffer Pool Services** . . . . . 309
  - SYSEDT Utility - Editor Buffer Pool Services . . . . . 309
  - Defining a Natural Security Library Profile . . . . . 309
  - Invoking and Operating SYSEDT . . . . . 309
    - Invoking the SYSEDT Utility . . . . . 309
    - Invoking a SYSEDT Function . . . . . 310
    - Using Direct Commands Help . . . . . 310
  - General Information . . . . . 310
  - Generation Parameters . . . . . 311
    - Modifying Parameter Values . . . . . 312
  - Users . . . . . 313
  - Logical Files . . . . . 313
  - Recovery Files . . . . . 314
  - System Administration Facilities . . . . . 315
- SYSERR Utility - Overview** . . . . . 316
  - SYSERR Utility - Overview . . . . . 316
- SYSERR - General Information on Messages** . . . . . 317
  - SYSERR - General Information on Messages . . . . . 317
  - Message Types . . . . . 318
  - Message Languages . . . . . 318
  - Issuing Messages . . . . . 318
  - Retrieving Natural System Short Messages . . . . . 319
  - Retrieving User-defined Short Messages . . . . . 319
  - Obtaining Message Information . . . . . 320
- Invoking SYSERR** . . . . . 321
  - Invoking SYSERR . . . . . 321
- SYSERR - Parameters** . . . . . 322
  - SYSERR - Parameters . . . . . 322
  - Message Type . . . . . 322
  - Library . . . . . 322
  - Message Number . . . . . 322
  - Language Codes . . . . . 322
- SYSERR - Functions** . . . . . 323
  - SYSERR - Functions . . . . . 323
  - Adding Messages . . . . . 324
    - Copying a Sample Message . . . . . 326
  - Deleting Messages . . . . . 326

Displaying Messages . . . . .	327
Modifying Messages . . . . .	328
Printing Messages . . . . .	330
Scanning Messages . . . . .	332
Selecting Messages from a List . . . . .	334
Translating Messages into other Languages . . . . .	336
Copying Messages to a System File - SYSMAIN Utility . . . . .	337
<b>SYSErr - Direct Commands . . . . .</b>	<b>338</b>
SYSErr - Direct Commands . . . . .	338
<b>SYSErr - Upper Case - ERRUPPER . . . . .</b>	<b>339</b>
SYSErr - Upper Case - ERRUPPER . . . . .	339
<b>Replacing Characters - ERRCHAR . . . . .</b>	<b>340</b>
Replacing Characters - ERRCHAR . . . . .	340
<b>SYSErr - Unloading Messages - ERRULDUS . . . . .</b>	<b>341</b>
SYSErr - Unloading Messages - ERRULDUS . . . . .	341
<b>SYSErr - Loading Messages - ERRLODUS . . . . .</b>	<b>343</b>
SYSErr - Loading Messages - ERRLODUS . . . . .	343
<b>Appl. Programming Interface USR0020P . . . . .</b>	<b>344</b>
Appl. Programming Interface USR0020P . . . . .	344
<b>SYSMAIN Utility - Overview . . . . .</b>	<b>345</b>
SYSMAIN Utility - Overview . . . . .	345
<b>SYSMAIN General Information . . . . .</b>	<b>346</b>
SYSMAIN General Information . . . . .	346
Objects . . . . .	346
Environments . . . . .	346
Functions . . . . .	347
<b>Invoking SYSMAIN . . . . .</b>	<b>348</b>
Invoking SYSMAIN . . . . .	348
Invoking SYSMAIN Online . . . . .	348
Terminating SYSMAIN Online . . . . .	348
Invoking SYSMAIN in Batch Mode . . . . .	348
Invoking SYSMAIN by an Appl. Programming Interface . . . . .	349
<b>SYSMAIN Functions and Function Processing . . . . .</b>	<b>350</b>
SYSMAIN Functions and Function Processing . . . . .	350
Commands . . . . .	350
Function Processing . . . . .	352
Command Line . . . . .	354
Selection List . . . . .	354
Enhanced Selection Criteria . . . . .	357
<b>SYSMAIN Direct Commands . . . . .</b>	<b>358</b>
SYSMAIN Direct Commands . . . . .	358
General Direct Command Syntax . . . . .	358
where-clause . . . . .	358
with-clause . . . . .	359
Sequence of Syntax Elements . . . . .	359
Effects of Direct Commands . . . . .	359
<b>SYSMAIN Parameters and Keywords . . . . .</b>	<b>360</b>
SYSMAIN Parameters and Keywords . . . . .	360
Description of Parameters and Keywords . . . . .	361
CRITERIA (A1) . . . . .	361
OBJECT TYPE (A15) . . . . .	361
DATE FROM (A10) . . . . .	361
TIME FROM (A5) . . . . .	361
DATE TO (A10) . . . . .	361
TIME TO (A5) . . . . .	362
USER ID (A8) . . . . .	362

TERMINAL ID (A8) . . . . .	362
DDM DBID (N5) . . . . .	362
DDM FNR (N5) . . . . .	362
DDM NAME (A32) . . . . .	362
ENVIRONMENT NAME (A8) . . . . .	362
ERROR NUMBER FROM (N4) . . . . .	363
ERROR NUMBER TO (N4) . . . . .	363
ERROR TYPE (A1) . . . . .	363
FDIC (A27) . . . . .	363
FSEC (A27) . . . . .	363
NEW NAME (A8) . . . . .	364
NEW NUMBER FROM (N4) . . . . .	364
NEW NUMBER TO (N4) . . . . .	364
OBJECT NAME (A9) . . . . .	364
OBJECT TYPE (A15) . . . . .	365
PROFILE NAME (A8) . . . . .	366
PROFILE TYPE (A3) . . . . .	366
REPLACE (A1) . . . . .	366
RULE NAME (A32) . . . . .	366
RULE TYPE (A2) . . . . .	367
SELECTION LIST (A1) . . . . .	367
SET NUMBER (N2) . . . . .	367
SET USER (A8) . . . . .	367
SOURCE CIPHER (A8) . . . . .	368
SOURCE DATABASE (N5) . . . . .	368
SOURCE FILE (N5) . . . . .	368
SOURCE LANGUAGE (A9) . . . . .	368
SOURCE LIBRARY (A8) . . . . .	368
SOURCE NAME (A8) . . . . .	368
SOURCE PASSWORD (A8) . . . . .	369
SUBFILE NAME (A8) . . . . .	369
SUBFILE TYPE (A3) . . . . .	369
TARGET CIPHER (A8) . . . . .	369
TARGET DATABASE (N5) . . . . .	369
TARGET FILE (N5) . . . . .	369
TARGET LANGUAGE (A9) . . . . .	369
TARGET LIBRARY (A8) . . . . .	370
TARGET NAME (A8) . . . . .	370
TARGET PASSWORD (A8) . . . . .	370
XREF (A1) . . . . .	370
Additional Keywords for Direct Commands . . . . .	371
Range Notation for SYSMAIN Parameters . . . . .	372
<b>SYSMAIN Programming Objects . . . . .</b>	<b>373</b>
SYSMAIN Programming Objects . . . . .	373
Programming Object Menus . . . . .	373
Processing Status . . . . .	377
Options H, I and L . . . . .	377
Options B, X and Z . . . . .	377
Options A, C and S . . . . .	377
Direct Commands for Programming Objects . . . . .	379
COPY and MOVE Direct Command Syntax . . . . .	379
DELETE Direct Command Syntax . . . . .	380
FIND, LIST and LISTLIB Direct Command Syntax . . . . .	381
RENAME Direct Command Syntax . . . . .	381
Programming Objects in Batch Mode . . . . .	382
XREF Considerations for Programming Objects . . . . .	382

XREF set to N . . . . .	382
XREF set to Y or F . . . . .	382
XREF set to F . . . . .	382
XREF set to S . . . . .	382
Errors . . . . .	383
<b>SYSMAIN Debug Environments</b> . . . . .	384
SYSMAIN Debug Environments . . . . .	384
Debug Environment Menus . . . . .	384
Processing Status . . . . .	385
Direct Commands for Debug Environments . . . . .	386
COPY and MOVE Direct Command Syntax . . . . .	386
DELETE Direct Command Syntax . . . . .	386
LIST Direct Command Syntax . . . . .	387
RENAME Direct Command Syntax . . . . .	387
Debug Environments in Batch Mode . . . . .	387
<b>SYSMAIN Error Messages</b> . . . . .	388
SYSMAIN Error Messages . . . . .	388
Error Message Menus . . . . .	388
Language Parameter Considerations . . . . .	389
Renumber an Error Message . . . . .	391
Processing Status . . . . .	392
Option L . . . . .	392
Options A, E and S . . . . .	392
Direct Commands for Error Messages . . . . .	393
COPY and MOVE Direct Command Syntax . . . . .	393
DELETE Direct Command Syntax . . . . .	394
FIND Direct Command Syntax . . . . .	394
LIST Direct Command Syntax . . . . .	394
RENAME Direct Command Syntax . . . . .	395
Error Messages in Batch Mode . . . . .	395
<b>SYSMAIN Profiles</b> . . . . .	396
SYSMAIN Profiles . . . . .	396
Profile Menus . . . . .	396
Processing Status . . . . .	398
Direct Commands for Profiles . . . . .	398
COPY and MOVE Direct Command Syntax . . . . .	398
DELETE Direct Command Syntax . . . . .	399
LIST Direct Command Syntax . . . . .	399
RENAME Direct Command Syntax . . . . .	399
Profiles in Batch Mode . . . . .	399
<b>SYSMAIN Rules</b> . . . . .	400
SYSMAIN Rules . . . . .	400
Rule Menus . . . . .	400
Processing Status . . . . .	402
Direct Commands for Rules . . . . .	403
COPY and MOVE Direct Command Syntax . . . . .	403
DELETE Direct Command Syntax . . . . .	404
LIST Direct Command Syntax . . . . .	404
Rules in Batch Mode . . . . .	404
<b>SYSMAIN DDMs</b> . . . . .	405
SYSMAIN DDMs . . . . .	405
DDM Menus . . . . .	405
Processing Status . . . . .	406
Direct Commands for DDMs . . . . .	406
COPY and MOVE Direct Command Syntax . . . . .	407
DELETE Direct Command Syntax . . . . .	407

LIST Direct Command Syntax . . . . .	408
DDMs in Batch Mode . . . . .	408
<b>SYSMAIN DL/I Subfiles</b> . . . . .	409
SYSMAIN DL/I Subfiles . . . . .	409
General Information . . . . .	409
Processing Status . . . . .	410
Direct Commands for DL/I Subfiles . . . . .	410
COPY and MOVE Direct Command Syntax . . . . .	410
DELETE and LIST Direct Command Syntax . . . . .	411
<b>Commands Issued to SYSMAIN</b> . . . . .	412
Commands Issued to SYSMAIN . . . . .	412
<b>SYSMAIN PF Keys</b> . . . . .	414
SYSMAIN PF Keys . . . . .	414
<b>SYSMAIN - Data Rejected</b> . . . . .	415
SYSMAIN - Data Rejected . . . . .	415
Object Rejection and Reasons . . . . .	415
SYSMAIN Error Notification . . . . .	415
Data Entry Errors . . . . .	416
Processing Errors . . . . .	416
<b>Special Considerations for Administrators</b> . . . . .	419
Special Considerations for Administrators . . . . .	419
SYSMAIN Security . . . . .	419
File Security . . . . .	419
Natural Security . . . . .	421
User Exit Routines . . . . .	422
MAINEX01 - First User Exit Routine for Object Interrogation . . . . .	423
MAINEX02 - Second User Exit Routine for Object Interrogation . . . . .	423
MAINEX03 - User Exit Routine for Request Interrogation . . . . .	424
MAINEX04 - User Exit Routine for Modification of File Assignments . . . . .	424
MAINEX05 - User Exit Routine for Verification of Direct Commands . . . . .	424
MAINEX06 - User Exit Routine for SYSMAIN Initialization . . . . .	424
MAINEX07 - User Exit Routine for SYSMAIN Termination . . . . .	425
MAINEX08 - User Exit Routine for Nothing Found in Batch Mode . . . . .	425
MAINEX09 - User Exit Routine for Abnormal Termination in Batch Mode . . . . .	425
MAINEX10 - User Exit Routine for Command Errors in Batch Mode . . . . .	426
MAINEX11 - User Exit Routine for Setting Special Flags to SYSMAIN . . . . .	426
<b>SYSNCP Utility - Overview</b> . . . . .	427
SYSNCP Utility - Overview . . . . .	427
<b>SYSNCP - Introduction</b> . . . . .	428
SYSNCP - Introduction . . . . .	428
Object-Oriented Data Processing . . . . .	429
Features of the Command Processor . . . . .	430
Components of the Command Processor . . . . .	431
What is a Command? . . . . .	432
Creating a Command Processor . . . . .	433
Steps in Command Processor Creation . . . . .	434
<b>SYSNCP - Prerequisites</b> . . . . .	435
SYSNCP - Prerequisites . . . . .	435
Windows . . . . .	435
UNIX . . . . .	435
<b>Invoking SYSNCP</b> . . . . .	436
Invoking SYSNCP . . . . .	436
Help . . . . .	436
<b>SYSNCP - Processor Selection</b> . . . . .	437
SYSNCP - Processor Selection . . . . .	437

<b>SYSNCP - Header Records</b>	438
SYSNCP - Header Records	438
Create New Processor	438
Modify Header - General Explanations	439
Keyword Runtime Options - Header 1	440
Keyword Editor Options - Header 2	443
Miscellaneous Options - Header 3	444
Command Data Handling - Header 4	446
Runtime Error Handling - Header 5	448
Statistics - Header 6	449
Status - Header 7	449
<b>SYSNCP - Keyword Maintenance</b>	450
SYSNCP - Keyword Maintenance	450
Define Keywords	450
Editor Commands	454
Positioning Commands	454
Line Commands	455
<b>SYSNCP - Function Maintenance</b>	456
SYSNCP - Function Maintenance	456
Define Functions	456
Editor Commands	459
Direct Command QUICK-EDIT	459
Local and Global Functions	459
Procedure for Validating Functions	461
<b>SYSNCP - Runtime Actions</b>	462
SYSNCP - Runtime Actions	462
Define Runtime Actions	462
Runtime Action Editor	464
<b>SYSNCP - Processor Cataloging</b>	467
SYSNCP - Processor Cataloging	467
<b>SYSNCP - Administrator Services</b>	468
SYSNCP - Administrator Services	468
Select Processor	468
Copy Processor Source	469
Delete Processor Source	471
Print Source/Object/NCP Buffer	472
Unload Processor	473
Load Processor	474
Freeze Processor Source	475
References from Natural Security	475
<b>SYSNCP - Session Profile</b>	476
SYSNCP - Session Profile	476
Session Profile Name	476
Session Parameters - Profile 1	477
Color Attributes - Profile 2	478
Miscellaneous Attributes - Profile 3	478
<b>SYSPARM Utility</b>	479
SYSPARM Utility	479
Invoking SYSPARM	479
List Profiles	480
Display Profile	481
Add New Profile	481
Modify Profile	481
Editing Profiles	481
Functions	482
Help with Parameters	482

Copy Profile . . . . .	483
Delete Profile . . . . .	483
Direct Commands and Batch Processing . . . . .	483
<b>Batch Processing</b> . . . . .	484
Syntax of SYSPARM . . . . .	484
where-clause . . . . .	487
with-clause . . . . .	488
Example of SYSPARM in Batch . . . . .	490
<b>SYSRDC Utility</b> . . . . .	492
SYSRDC Utility . . . . .	492
Data-Collecting Events . . . . .	493
Data Collected . . . . .	494
General Data . . . . .	494
Event-Specific Data . . . . .	494
Activating the Natural Data Collector . . . . .	495
Trace Recording . . . . .	496
User Exits for External Monitoring/Accounting . . . . .	496
Return Codes . . . . .	497
Examples of User-Exit Programs . . . . .	497
Calling the CMRDC Interface . . . . .	498
Retrieving Trace Records . . . . .	498
Stopping and Restarting Trace Recording . . . . .	499
Selecting Event Types for Trace Recording . . . . .	499
User-Defined Events . . . . .	500
CMRDC Return Codes . . . . .	500
Example Programs in Library SYSRDC . . . . .	501
<b>SYSRPC Utility - Overview</b> . . . . .	503
SYSRPC Utility - Overview . . . . .	503
<b>SYSRPC - Basic Functionality</b> . . . . .	504
SYSRPC - Basic Functionality . . . . .	504
Invoking SYSRPC . . . . .	504
Terminating SYSRPC . . . . .	504
Help . . . . .	504
<b>SYSRPC - Service Directory Maintenance</b> . . . . .	505
SYSRPC - Service Directory Maintenance . . . . .	505
Service Directory Concept . . . . .	505
Physical Nodes and Servers . . . . .	505
Location Transparency . . . . .	506
Invoking Service Directory Maintenance . . . . .	506
Fields . . . . .	507
Selection Criteria for Node and Server . . . . .	508
Commands and PF Keys . . . . .	509
Line Commands . . . . .	509
Direct Commands and PF Keys . . . . .	510
Defining Logical Node Names and Logical Services . . . . .	511
<b>SYSRPC - Stub Generation</b> . . . . .	512
SYSRPC - Stub Generation . . . . .	512
Invoking Stub Generation . . . . .	512
Parameter Specification . . . . .	513
Example of Stub Generation . . . . .	514
<b>SYSRPC - Parameter Maintenance</b> . . . . .	515
SYSRPC - Parameter Maintenance . . . . .	515
Invoking Parameter Maintenance . . . . .	515
Specifying Profile Parameters . . . . .	515
<b>SYSRPC - Server Command Execution</b> . . . . .	517
SYSRPC - Server Command Execution . . . . .	517

Invoking Server Command Execution . . . . .	517
Pinging a Server . . . . .	518
Terminating a Server . . . . .	519
Terminating a Server with Replicates . . . . .	519
<b>SYSRPC - Remote Directory Maintenance . . . . .</b>	<b>521</b>
SYSRPC - Remote Directory Maintenance . . . . .	521
Invoking Remote Directory Maintenance . . . . .	521
Fields . . . . .	522
Commands and PF Keys . . . . .	522
Line Commands . . . . .	522
Direct Commands and PF Keys . . . . .	523
<b>SYSTP Utility - Overview . . . . .</b>	<b>525</b>
SYSTP Utility - Overview . . . . .	525
<b>SYSTP - General Information . . . . .</b>	<b>526</b>
SYSTP - General Information . . . . .	526
General and Environment-Dependent Functions . . . . .	526
Invoking SYSTP . . . . .	526
Using SYSTP Statistics Screens . . . . .	526
<b>General SYSTP Functions . . . . .</b>	<b>528</b>
General SYSTP Functions . . . . .	528
Natural Monitoring - SYSMON . . . . .	528
Activate/Deactivate Monitor . . . . .	528
Display Monitor Terminal Statistics . . . . .	529
Display Monitor Program Statistics . . . . .	531
Natural Print/Work Files - SYSFILE . . . . .	532
Natural Swap Information . . . . .	533
Administration . . . . .	534
Debugging Facilities . . . . .	534
Information . . . . .	535
Parameter Service . . . . .	537
Status Information . . . . .	537
Buffer Usage Statistics - BUS . . . . .	537
Natural Subsystems and Roll Server Information . . . . .	539
Natural Thread Usage Statistics . . . . .	539
Show Physical GETMAIN Statistics . . . . .	542
<b>SYSTP Functions under CICS . . . . .</b>	<b>543</b>
SYSTP Functions under CICS . . . . .	543
Natural User Sessions . . . . .	543
Commands for Natural User Sessions . . . . .	544
Natural User Session Statistics . . . . .	545
Natural Roll Facilities . . . . .	547
Natural Thread Groups . . . . .	548
Natural Storage Threads . . . . .	549
NCI Global System Information . . . . .	550
NCI Generation Options . . . . .	551
Natural Thread Group Definitions . . . . .	552
Own Natural User Session . . . . .	553
CICS Task Information . . . . .	553
System Administration Facilities . . . . .	553
System Snapshot for Logging . . . . .	553
Reset System Highwater Marks . . . . .	553
Applied NCI Zaps . . . . .	554
<b>SYSTP Functions under IMS/TM . . . . .</b>	<b>555</b>
SYSTP Functions under IMS/TM . . . . .	555
Broadcasting . . . . .	555
Display Environment Data . . . . .	555

- Monitoring . . . . . 556
- Multi Session . . . . . 556
- Applied NII Zaps . . . . . 556
- SYSTP Functions under TIAM and UTM . . . . . 557**
- SYSTP Functions under TIAM and UTM . . . . . 557
- P-Key Utility . . . . . 557
- Load User Values . . . . . 558
- Set Key Assignment Mode . . . . . 559
- Loading Send-Key Codes to P Keys . . . . . 560
- Application Programming Interface LPFSUP01 . . . . . 561
- Show Common Memory Pools . . . . . 561
- SYSTP in Batch for CICS Sessions . . . . . 562**
- SYSTP in Batch for CICS Sessions . . . . . 562
- Invoking SYSTP in Batch Mode . . . . . 562
- Evaluating the Log File . . . . . 562
- SYSTRANS Utility - Overview . . . . . 564**
- SYSTRANS Utility - Overview . . . . . 564
- SYSTRANS - Introduction . . . . . 565**
- SYSTRANS - Introduction . . . . . 565
- General Information . . . . . 565
- Invoking SYSTRANS Online . . . . . 566
- Name and Range Specification . . . . . 567
- SYSTRANS and Entire Connection . . . . . 567
- SYSTRANS and Natural Security . . . . . 568
- SYSTRANS - Unload Function . . . . . 569**
- SYSTRANS - Unload Function . . . . . 569
- General Unload Options . . . . . 569
- User-Defined Conversion Table . . . . . 570
- Applying the Use Selection List Option . . . . . 570
- Objects to be Unloaded . . . . . 571
- Unloading Natural Objects . . . . . 571
- Unloading Maps . . . . . 573
- Unloading DDMs . . . . . 574
- Unloading Adabas FDTs . . . . . 575
- Unloading Error Message Texts . . . . . 576
- Unloading Command Processors . . . . . 576
- Unloading Libraries . . . . . 577
- Unloading in Batch Mode under OS/390 . . . . . 578
- User Exits for Unloading in Batch Mode . . . . . 579
- Unload Work File Specifications . . . . . 579
- Work File for Unload Command Input - Work File 2 . . . . . 580
- Fixed Parameter Positions . . . . . 580
- Parameters Separated by a Comma . . . . . 582
- Natural Profile Parameters . . . . . 582
- SYSTRANS - Load, Scan and Restart Load Functions . . . . . 584**
- SYSTRANS - Load, Scan and Restart Load Functions . . . . . 584
- Load and Scan . . . . . 584
- General Load/Scan Options . . . . . 584
- Objects to be Loaded/Scanned . . . . . 586
- Loading in Batch Mode under OS/390 . . . . . 591
- User Exits for Loading in Batch Mode . . . . . 592
- Load Work File Specifications . . . . . 592
- Natural Parameter Settings . . . . . 592
- Restart Load . . . . . 592
- SYSTRANS - Direct Transfer Functions . . . . . 594**
- SYSTRANS - Direct Transfer Functions . . . . . 594

Direct Transfer using RPC . . . . .	594
Restart Direct Transfer . . . . .	595
Get Report from Direct Transfer Load . . . . .	595
Define Local TRANSFER System . . . . .	595
Preparing to Use Natural RPC for a Direct Transfer . . . . .	595
<b>Direct Commands and CALLNAT Interface . . . . .</b>	<b>597</b>
Direct Commands and CALLNAT Interface . . . . .	597
Executing SYSTRANS in Direct Command Mode . . . . .	597
Executing SYSTRANS from within an Application . . . . .	598
Available Keywords and Values . . . . .	599
Functions . . . . .	600
Objects . . . . .	600
Parameters . . . . .	600
Options . . . . .	601
<b>SYSTRANS Profile . . . . .</b>	<b>603</b>
SYSTRANS Profile . . . . .	603
Listing of Text Object PROFILE . . . . .	603
Parameters in the Text Object PROFILE . . . . .	604
Unload Parameters . . . . .	604
Load Parameters . . . . .	605

# Natural Utilities - Overview

This documentation describes how Natural invokes a utility, lists and groups all utilities available in Natural and provides detailed descriptions of each utility.

This section covers the follow

- Utility Activation
  - Utility List
  - Utilities Grouped by Function
- 

## Utility Activation

Natural invokes a Natural utility without performing a logon to the corresponding utility library in the FNAT system file. This applies to all utilities listed in Support of Activation without Logon in the current Natural Release Notes for Mainframes. As a result, Natural preserves the global data area (GDA) and/or application-independent variables (AIV). The current user library and the settings are maintained. (To reset the GDA and/or the AIVs, see the profile parameter FREEGDA in the Natural Parameter Reference documentation.)

To preserve the settings of your application environment, do **not** log on to a utility library. Instead, invoke a utility by using the Natural system command that corresponds to the utility.

After terminating a utility, you will be returned to the library from which you invoked the utility. However, if you explicitly log on to a utility library before invoking the utility, you will stay in this (utility) library after utility termination.

For information on how to control the use of Natural utilities with Natural Security, see the section Protecting Utilities in the Natural Security documentation.

## Utility List

Below is an alphabetical list of the Natural utilities and a brief description of the functions they provide:

Utility	Function
ADACALL (SYSADA)	Issues Adabas direct calls (native commands) directly to an Adabas database.
DBLOG	Logs database calls: indicates which Adabas commands, DL/I calls or SQL statements are issued by a Natural program.
Debugger	Used to test various aspects of a Natural application and assists in locating errors in the processing flow of a program. Provides statistics on objects and statement lines invoked during program execution.
DUMP	Provides information for Software AG technical support personnel in order to locate an error that caused an abnormal termination (abend) of the Natural system. See the system command DUMP in the Natural Command Reference documentation.
INPL	Loads or scans Natural modules or DDMs from Software AG datasets (for example, Natural INPL tapes) from Work File 1. In addition, it provides a Natural Security Recover function that enables you to force an initialization of the Natural Security environment.
NATPAGE Screen Capturing	Captures screens (maps and reports) during a Natural session.
NATRJE	Submits JCL cards from a Natural program to the operating system for scheduling and execution.
NATUNLD/ NATLOAD	NATUNLD: unloads Natural programming objects, error messages and DDMs from a system file to a work file.  NATLOAD: loads Natural programming objects, error messages and DDMs from a work file into a system file.
Object Handler	Processes Natural and non-Natural objects for distribution in Natural environments. This is done by unloading the objects in the source environment into work files and loading them from work files into the target environment.
Recording	Records commands and input data entered during a Natural session. Re-executes a recorded session.
SYSBPM	Monitors and controls the Natural buffer pool.
SYSDDM	Creates and maintains Natural DDMs.
SYSEDT	Displays parameters and runtime information for the editor buffer pool. Modifies parameters and deletes logical work and recovery files.
SYSERR	Creates application-specific messages. In addition, it can be used to modify the texts of the existing Natural system messages (not recommended).
SYSMAIN	Transfers Natural programming objects, error messages, DDMs and other objects from one library to another in your current environment.
SYSNCP	Defines command-driven navigation systems for Natural applications.
SYS Parm	Creates and maintains Natural parameter profiles.
SYSRDC	Records monitoring and accounting data on the processing flow within a Natural application.
SYSRPC	Establishes and maintains Natural Remote Procedure Call environments.
SYSTP	Monitors and controls various TP-monitor-specific characteristics of Natural.
SYSSTRANS	Transfers Natural objects, maps, DDMs, libraries, command processors and error messages as well as Adabas FDTs from one hardware platform to another.

## Utilities Grouped by Function

Below are the Natural utilities grouped according to their functions:

<b>Administration</b>	<b>Debugging</b>	<b>Monitoring</b>	<b>Object Transfer</b>
NATRJE	ADACALL (SYSADA)	SYSBPM	INPL
SYSBPM	DBLOG	SYSEDT	NATUNLD/NATLOAD
SYSDDM	Debugger	SYSRDC	Object Handler
SYSEDT	DUMP	SYSTP	SYSMAIN
SYSERR	NATPAGE Screen Capturing		SYSTRANS
SYSNCP	Recording		
SYSPARM	SYSRDC		
SYSRPC			
SYSTP			

# ADACALL - Issuing Adabas Direct Calls

The utility ADACALL can be used to issue Adabas direct calls (native commands) directly to an Adabas database for learning and testing and for analyzing problems.

The utility ADACALL is contained in the library SYSADA.

The ADACALL documentation covers the following topics:

- Invoking ADACALL
- ADACALL Parameters
- ADACALL Commands and PF Keys
- Adabas OP Command
- User Exit ADAEXIT

## Invoking ADACALL

### To invoke ADACALL

- Enter the system command SYSADA.

The ADACALL main screen similar to the example screen below is displayed:

```

15:53:32          ***** NATURAL ADACALL UTILITY *****          2002-04-04
User SAG              - ADABAS Direct Calls -
Mode Char                                Call No. 45
*** Control Block ***          First Byte 30
  Cmd L3          Cmd ID SAG          File 316          Database 10
Resp 0          ISN 382          ISQ 0          ISL 0
FBL 210          RBL 980          SBL 140          VBL 140          IBL 0
COP1          COP2          User Area          Cmd Time 4
Addition1          Addition2 Addition3          Addition4          Addition5
AA]?          227 48
*** Buffer Areas ***
Format AA,AC,AE.

Record 11111003ARTHUR          DENT

Search

Value

ISN
Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Main Exit          Char Hex View Prnt Run Init Canc

```

On the ADACALL main screen, you enter the necessary parameters on the screen and then execute the Adabas command by either choosing PF10 (Run) or entering the ADACALL command EXEC in the command line.

In the example screen above, the Adabas command L3 was executed for a logical read of the employees file.

Except for the control block, which is shown in full, only a part of the buffer is displayed. You can view the buffers in their entirety by using any of the ADACALL direct commands or PF keys listed below.

## ADACALL Parameters

The parameters which can be specified are listed below. To obtain a summarized explanation of the parameters, use the ADACALL online help function.

### To invoke the online help function

- Place the cursor in the corresponding field and enter a question mark (?) or choose PF1.  
(For read-only fields, only PF1 applies.)

For further details, see the Adabas documentation Command Reference and Messages and Codes.

Parameter	Explanation
Mode	Indicates the display mode of the buffer contents:  Char Character values. Hex Hexadecimal values.  To change modes, see the ADACALL commands CHAR and HEX.
Call No.	Number of commands executed since the start of the session.
First Byte	The first byte of the Adabas control block.  Indicates whether 1-byte or 2-byte database IDs (DBID) and file numbers (FNR) are used:  H'00' = 1-byte DBID, FNR (file numbers 1 - 255) H'30' = 2-byte DBID, FNR (file numbers greater than 255)
Cmd	Adabas command.  Enter and execute the Adabas OP command to specify the parameters described in the relevant section below.
Cmd ID	Command ID.
File	File number.  If First Byte is set to H'00': 3-digit file number, "Database" not equal to 0.  If First Byte is set to H'30': 5-digit file number.
Database	Database number. Defaults to the DBID of the FUSER file of the current Natural session (see File above).  If First Byte is set to H'30', then the database number will be moved to the response code field of the Adabas control block at execution time.
Resp	Response code returned after the command is executed.
ISN	Internal sequence number.
ISQ	ISN quantity.
ISL	Lowest ISN value for ISN lists.
FBL	Format buffer length in bytes (maximum 210).

Parameter	Explanation
RBL	Record buffer length in bytes (maximum 980).
SBL	Search buffer length in bytes (maximum 140).
VBL	Value buffer length in bytes (maximum 140).
IBL	ISN buffer length in bytes (maximum 200).
COP1	Command option 1.
COP2	Command option 2.
User Area	User area for the control block.
Cmd Time	The time taken to execute the command, converted to 1/100th seconds for convenience.
Addition1	Additions 1.
Addition2	Additions 2. If the call was successful, it displays the compressed length of the record being read and the decompressed length of the data requested via the format buffer. If a non-zero response is returned and the error was a result of an invalid format buffer, the field in error and its offset into the format buffer are displayed.
Addition3	Additions 3.
Addition4	Additions 4. If a VSAM file is being read, this field is set to VSAM if initialized.
Addition5	Additions 5.
Format	Format buffer. (The final period is necessary.)
Record	Record buffer.
Search	Search buffer. (The final period is necessary.)
Value	Value buffer.
ISN	ISN buffer.

## Adabas OP Command

When you execute the Adabas command OP (Open), ADACALL provides a window where you can specify the following parameters:

- maximum ISNs to be stored in the internal ISN buffer,
- maximum records permitted in hold status,
- maximum CIDs which may be active,
- maximum time permitted for execution of an Sx command.

In the window, enter the relevant information and choose ENTER.

For an explanation of the parameters and valid values, refer to the Adabas Command Reference documentation.

## ADACALL Commands and PF Keys

The ADACALL direct commands listed below are provided to change ADACALL parameter settings or to switch between screens by either entering a command in the command line or choosing a corresponding PF key.

In addition to ADACALL commands, from the command line, you can also issue Natural system commands.

Command	PF Key	Function						
	PF1	Invoke the help function for ADACALL. If the cursor is positioned on one of the various ADACALL parameters and PF1 is pressed, help information on this parameter is displayed.						
	PF2	Return to the ADACALL main screen. Mode is set to CHAR.						
<u>B</u> ACK	PF5	Page back to the previous buffer when viewing the buffers in their entirety.  Valid only after the VIEW command has been applied, which means that the command is not applicable from the ADACALL main screen.						
CB		Display the control block buffer entirely; valid in hexadecimal mode only.						
CHAR	PF6	Change the current mode to character mode (EBCDIC).						
D		Display extended error message text for response code received. When an Adabas response other than 0 (zero) is returned, the corresponding short error message text is displayed in the message line. The extended text can be viewed by issuing this command.						
EXEC RUN	PF10	Execute the direct command with the parameters specified.						
EXIT STOP Q .	PF3 PF12	Exit. If pressed while on the ADACALL main screen, ADACALL is terminated. If one of the buffer screens is being viewed, the ADACALL main screen is displayed with Mode unchanged.						
FB		Display the format buffer in its entirety.						
<u>F</u> WD	PF4	Page forward to the next buffer when viewing the buffers in their entirety.  Valid only after the VIEW command has been applied, which means that the command is not applicable from the ADACALL main screen.						
HEX	PF7	Change the current mode to hexadecimal.						
IB		Display the ISN buffer in its entirety.						
INIT	PF11	Initialize/reset buffer(s). A window is displayed and one of the following values can be entered for the buffers indicated:  <table style="margin-left: 40px; border: none;"> <tr> <td style="padding-right: 20px;">H</td> <td>Initialize the corresponding buffer(s) with binary zeroes (H'00').</td> </tr> <tr> <td>any character except H or blank</td> <td>Initialize the corresponding buffer(s) with blanks (H'40').</td> </tr> <tr> <td>blank character</td> <td>Do not initialize the corresponding buffer(s).</td> </tr> </table> <p>If you enter INIT ALL, all buffers except the control block are initialized with blanks. Alternatively, the command INIT FB RB SB VB IB (not all buffers need be listed) can be specified and all buffers in the list are initialized with blanks.</p> <p><b>Note:</b> The ISN buffer is always initialized with binary zeroes.</p>	H	Initialize the corresponding buffer(s) with binary zeroes (H'00').	any character except H or blank	Initialize the corresponding buffer(s) with blanks (H'40').	blank character	Do not initialize the corresponding buffer(s).
H	Initialize the corresponding buffer(s) with binary zeroes (H'00').							
any character except H or blank	Initialize the corresponding buffer(s) with blanks (H'40').							
blank character	Do not initialize the corresponding buffer(s).							
PRINT	PF9	Generate and display a report on the status of all buffers.  The Natural terminal command %H can be used to obtain a hardcopy.						

Command	PF Key	Function
	PF1	Invoke the help function for ADACALL. If the cursor is positioned on one of the various ADACALL parameters and PF1 is pressed, help information on this parameter is displayed.
	PF2	Return to the ADACALL main screen. Mode is set to CHAR.
RB		Display the record buffer in its entirety.
RUN		Same as EXEC.
SB		Display the search buffer in its entirety.
VB		Display the value buffer in its entirety.
VIEW	PF8	<p>Display all buffers in their entirety. The first buffer to be displayed is the record buffer. The FWD command can be used to page through the other buffers.</p> <p>If you VIEW the record buffer in hexadecimal mode, the data are displayed on four pages:</p> <p>To page forwards, enter the command FWD or choose PF4.  To page backwards, enter the command BACK or choose PF5.  To display a specific page, enter a page number from 1 to 4 in the field "Specify next page number".</p> <p>To view buffers individually, enter any of the following commands:</p> <p>FB   Format buffer  RB   Record buffer  SB   Search buffer  VB   Value buffer  IB   ISN buffer  CB   Control block (default).  Valid in hexadecimal mode only: change to HEX before executing VIEW.</p>
VSAM		If VSAM has been defined for the current Natural session, this direct command can be issued to access or update VSAM files. When you issue this command, you are prompted by a window for the VSAM file name. When the command is executed, it is directed to the appropriate VSAM file.

## User Exit ADAEXIT

ADACALL allows direct commands to be issued to any database. Therefore, as a means of security, a user exit is supplied. This user exit is called ADAEXIT and is contained in the library SYSADA. You can modify ADAEXIT as required. The Adabas control block is passed as a parameter to ADAEXIT. You can change the source code of the user exit so as to modify the contents of the control block. By simply changing the database ID or file number, or by setting the Command Code to XX, you can prevent database calls from being performed.

# DBLOG Utility - Logging Database Calls

The DBLOG utility is used to log Adabas commands, DL/I (and SYNC/ROLB) calls, or SQL statements. Logging is useful for tuning an application (controlling the flow of commands accessing the database) and for analysing error codes that may be returned from the database.

**Related Documentation:**

Adabas, DL/I and DB2.

The DBLOG documentation covers the following topics:

- Executing DBLOG
- DBLOG Menu
- DBLOG Trace Screen
- DBLOG Snapshot Function
- DBLOG Direct Commands

# Executing DBLOG

The DBLOG utility logs each Adabas command, DL/I (and SYNC/ROLB) calls or SQL statement after it has been processed by the database system. Log recording starts when you activate DBLOG and execute or run a Natural program. Logging remains active until you enter an appropriate command or until you terminate your Natural session.

Below is information on:

- Data Processing and Storage
  - Activating and Deactivating DBLOG
  - Using Selective DBLOG
- 

## Data Processing and Storage

The data logged by the DLBOG utility are written into the Natural debug buffer. The size of the buffer is determined by the profile parameter DSIZE as described in the Natural Parameter Reference documentation. If there is not enough space available, only the most recent log data will be held in the Natural debug buffer.

DBLOG can be used online or in batch mode. DL/I and SYNC/ROLB calls can be logged under CICS, under IMS/TM or in batch mode.

For further details on batch-mode processing, refer to Natural in Batch Mode as described in the Natural Operations for Mainframes documentation.

The logs recorded are displayed on the DBLOG Trace screen.

The DBLOG utility provides default settings for data recording. With the DBLOG Menu, you can specify selection criteria for the commands, calls or statements to be logged and the information displayed. The DBLOG Menu also provides functions for activating or deactivating logging. To control DBLOG execution, you can also use direct commands.

The fields of the DBLOG Trace screen, the DBLOG Menu and direct command execution are explained in the relevant sections of the DBLOG documentation.

## Activating and Deactivating DBLOG

Below are the commands that apply when invoking DBLOG using the default settings for logging records. See also DBLOG Direct Commands for additional information.

### To activate or deactivate DBLOG

- Adabas:  
In the command line, enter the toggle command TEST DBLOG.  
Or, in the command line, enter TEST DBLOG ON or TEST DBLOG OFF.  
Or, on the DBLOG Menu, enter Function Code **B** or Function Code **E**.
- DL/I:  
In the command line, enter the toggle command TEST DBLOG D.  
Or, in the command line, enter TEST DBLOG D ON or TEST DBLOG D OFF.  
Or, on the DBLOG Menu, enter Function Code **B** or Function Code **E**.
- SQL:  
In the command line, enter the toggle command TEST DBLOG Q.  
Or, in the command line, enter TEST DBLOG Q ON or TEST DBLOG Q OFF.

Or, on the DBLOG Menu, enter Function Code **B** or Function Code **E**.

## Using Selective DBLOG

Below is an example of logging Adabas commands, DL/I calls or SQL statements with selection criteria specified with the DBLOG Menu.

### To perform DBLOG with selection criteria

1. Invoke the DBLOG Menu:
  - Adabas:  
In the command line, enter TEST DBLOG MENU.
  - DL/I:  
In the command line, enter TEST DBLOG D MENU.
  - SQL:  
In the command line, enter TEST DBLOG Q MENU.
2. On the DBLOG Menu, specify logging restrictions and activate logging:  
Complete the input fields and enter Function Code **B**.  
The message "DBLOG started now" is displayed.
3. Execute a Natural program which contains Adabas commands, DL/I calls or SQL statements.
4. Invoke the DBLOG Trace screen and deactivate logging:
  - Adabas:  
In the command line, enter TEST DBLOG.
  - DL/I:  
In the command line, enter TEST DBLOG D.
  - SQL:  
In the command line, enter TEST DBLOG Q.

The DBLOG Trace screen appears.
5. Clear the Natural debug buffer and deactivate logging:
  - Adabas:  
In the command line, enter TEST DBLOG OFF.
  - DL/I:  
In the command line, enter TEST DBLOG D OFF.
  - SQL:  
In the command line, enter TEST DBLOG Q OFF.

DBLOG terminates and the NEXT line appears.

See also DBLOG Direct Commands for additional information.

# DBLOG Menu

On the DBLOG Menu, you can activate or deactivate logging and specify which Adabas commands, DL/I calls or SQL statements are to be logged.

## To invoke the DBLOG Menu

- Adabas:  
In the command line, enter TEST DBLOG MENU.
- DL/I:  
In the command line, enter TEST DBLOG D MENU.
- SQL:  
In the command line, enter TEST DBLOG Q MENU.

Below is information on:

- DBLOG Menu - Functions
  - DBLOG Menu - Specifying Restrictions
- 

## DBLOG Menu - Functions

To execute the functions provided on the DBLOG Menu, enter a code in the relevant input field or choose a PF key:

Code or PF Key	Function	Explanation
B PF4/Begin	Begin Logging of ...	Activates the DBLOG logging of the Adabas commands, DL/I calls or SQL statements that match the selection criteria.  See also alternative commands in DBLOG Direct Commands.
	Optional Buffers for Code B	Only applicable to Adabas commands.  Selects additional Adabas buffers to be logged: see Specifying Buffers below.
E PF5/End	End and Display Log Records	Deactivates logging and displays the DBLOG Trace screen of the current log record if data exists in the Natural debug buffer. Current log data are kept in the Natural debug buffer.  See also alternative commands in DBLOG Direct Commands.
S PF6/Snap	Snapshot of Specific...	Adabas: Interrupts a program at a specified Adabas command and displays detailed information on this command only: see Snapshot Function for Adabas commands.  DL/I and SQL: Collects detailed information on a specified DL/I call or SQL statement: see Snapshot Function for DL/I calls and SQL statements.
PF3/Exit		Leaves the DBLOG Menu and returns to the NEXT line. The current log records are kept in the Natural debug buffer.
PF12/Canc		Clears the Natural debug buffer, leaves the DBLOG Menu and returns to the NEXT line.

## DBLOG Menu - Specifying Restrictions

Below are the input fields the DBLOG Menu provides for specifying selection criteria to restrict logging:

Field	Explanation
Skip	Only applicable with Function Code S. Number of commands, calls or statements to be skipped before logging is to start.
Program	Restricts logging to commands, calls or statements issued by the program specified.
DBID	Only applicable to Adabas commands. Restricts logging to commands issued for the database ID specified.
FNR	Only applicable to Adabas commands. Restricts logging to commands issued for the file number specified.
Line from Line to	Restricts logging to commands, calls or statements within the range of the source line numbers specified.
Low Resp High Resp	Only applicable to Adabas commands. Restricts logging to commands which result in a response code within the range specified.
Low Stat High Stat	Only applicable to DL/I calls. Restricts logging to calls which result in a status code within the range specified.
Low SQLC High SQLC	Only applicable to SQL statements. Restricts logging to statements which result in an SQL return code within the range specified.

## Specifying Buffers

Only applicable to Adabas commands.

In addition to the Adabas control block (CB) which is logged by default, you can select for logging one or more of the following Adabas buffers by entering any character in the input field next to the buffer(s) desired:

FB	Format Buffer
RB	Record Buffer
SB	Search Buffer
VB	Value Buffer
IB	ISN Buffer

The logs of the buffers (maximum is 80 bytes per buffer) can be displayed on the DBLOG Trace screen as described in the relevant section under Displaying Buffers.

The snapshot function (see the relevant section) logs all Adabas buffers by default. Therefore, you need not mark any of the optional buffers before you execute this function.

# DBLOG Trace Screen

The DBLOG Trace screen displays recorded log data on Adabas commands, DL/I calls or SQL statements which are kept in the Natural debug buffer.

Below is information on the DBLOG Trace screen for

- Adabas Commands
- DL/I Calls
- SQL Statements

## DBLOG Trace Screen - Adabas Commands

- Invoking DBLOG Trace - Adabas Commands
- DBLOG Trace Adabas Commands - Fields and Functions
- Displaying Buffers on the DBLOG Trace Screen

### Invoking DBLOG Trace - Adabas Commands

Below is an example of invoking the DBLOG Trace screen for Adabas commands.

1. Write the following Natural program in reporting mode:

```
/* PROG1
READ (3) EMPLOYEES BY NAME
DISPLAY NAME
END
```

2. In the command line, enter TEST DBLOG.  
The message "DBLOG started now" is displayed.
3. In the command line, enter RUN.
4. In the command line, enter TEST DBLOG again.

Logging is deactivated and the DBLOG Trace screen appears as shown in the example below:

16:22:50		***** NATURAL TEST UTILITIES *****							2002-03-11			
User SAG		- DBLOG Trace -							Library SAG			
M	No	Cmd	DB	FNR	Rsp	ISN	ISQ	CID	CID(Hex)	OP	Pgm	Line
—	1	L3	10	316		1300		???	00200101	V		0020
—	2	L3	10	316		1530		???	00200101	V		0020
—	3	L3	10	316		478		???	00200101	V		0020
—	4	RC	10	316				???	00200101	SI		0020
—	5	RC	10						00000000	F		0040
—												
—												
Command ==>												
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---												
Help Print Exit Top Posi Bot - + Canc												

### DBLOG Trace Adabas Commands - Fields and Functions

The fields and functions provided on the DBLOG Trace screen for Adabas commands are shown below:

Column	PF Key	Explanation
M		Input option for line commands that invoke extra windows with detailed information on Adabas buffers: see Displaying Buffers below.
No		Sequence number. The commands are displayed in the sequence in which they were executed.
Cmd		Adabas command.
DB		Database ID.
FNR		File number.
Rsp		Adabas response code.
ISN		Internal sequence number of record.
ISQ		ISN quantity.
CID		Command ID.
CID (Hex)		Command ID in hexadecimal format.
OP		Adabas Command Options 1 and 2.
Pgm		Program name.
Line		Source code line number.
	PF2 Print	Prints a hardcopy of a screenshot.
	PF3 Exit	Leaves the DBLOG Trace screen and returns to the NEXT line. The current log records are kept in the Natural debug buffer.
	PF4 Top	Goes to the top of the list.
	PF5 Posi	Moves log entries to the top of the screen: In Column <b>M</b> , position the cursor next to the desired command and sequence number listed in column <b>No</b> and choose PF5. The logs are repositioned starting with the sequence number selected.
	PF6 Bot	Goes to the bottom of the list.
	PF7 -	Scrolls up one page in the list.
	PF8 +	Scrolls down one page in a list.
	PF12 Canc	Clears the Natural debug buffer and deactivates logging.

## Displaying Buffers on the DBLOG Trace Screen

To display detailed buffer information, except for the Adabas control block which is recorded by default, you need to specify the buffer(s) desired in the DBLOG Menu screen (see the relevant sections) when logging is started. Otherwise, no logs will be recorded for the corresponding buffer(s). If, for example, only logging of the format buffer has been specified in the DBLOG Menu, you can only display the Format Buffer window but not the Record Buffer window. For each buffer specified, only the first 80 bytes are logged.

**To display detailed buffer information**

- On the DBLOG Trace screen, in the input field, next to the log(s) desired, enter any of the following line commands that corresponds to the buffer desired:

C	Control Block
F	Format Buffer
R	Record Buffer
S	Search Buffer
V	Value Buffer
I	ISN Buffer

An extra window displays the buffer specified. Below are example screens of the control block and the format buffer.

**Example Screen of Control Block:**

```

16:31:20          ***** NATURAL TEST UTILITIES *****          2002-03-11
User SAG          - DBLOG Trace -          Library SAG
M  No Cmd  DB  FNR  Rsp      ISN      ISQ  CID CID(Hex) OP Pgm      Line
C   1 L3   10  316          1300      ??? 00200101 V SAGTEST 0020
_   2 L3   10  316          1530      ??? 00200101 V SAGTEST 0020
- +-----+-----+-----+-----+-----+-----+-----+-----+
- ! _ SEQ NO ..      1 PROGRAM .. SAGTEST  LINE .. 0020          !
- ! Command Code .. L3      Command ID ... ??? 00200101 CB Start ... 30D5 !
- ! Response Code . 0000      ISN .....          1300      FNR first .. 01  !
- ! ISN Low Limit .          ISN Quantity .          FNR last ... 3C  !
- ! FB Length ..... 0009      RB Length .... 0014          SB Length .. 0008 !
- ! VB Length ..... 0014      IB Length .... 0000          !
- ! Command Opts .. V          !
- ! Additions 1 ... AE]?      C1C5BBCA40404040      Command Time .. 00000009 !
- ! Additions 2 ... ? ?      00120014          User Area ..... 00000000 !
- !          !
- ! Additions 3 ...          00000000000000000000 !
- ! Additions 4 ...          00000000000000000000 !
- ! Global FID .... 0000000000000000 !
- +-----+-----+-----+-----+-----+-----+
Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Print Exit Top  Posi Bot  -      +          Canc
    
```

**Example Screen of Format Buffer:**

```

16:33:37          ***** NATURAL TEST UTILITIES *****          2002-03-11
User SAG          - DBLOG Trace -          Library SAG
M  No Cmd  DB   FNR  Rsp      ISN      ISQ  CID CID(Hex) OP Pgm      Line
f  1 L3    10   316          1300          ??? 00200101 V SAGTEST 0020
_  2 L3    10   316          1530          ??? 00200101 V SAGTEST 0020
_  +-----+-----+-----+-----+-----+-----+-----+-----+
_  !  _      Seq No ..      1  Format Buffer          ! 0
_  ! 0000 * C1C56BF0 F2F06BC1 4B000000 00000000 * AE,020,A.      * 0000 ! 0
_  ! 0010 * 00000000 00000000 00000000 00000000 *          * 0010 !
_  !          !          !          !          !          !          !          !
_  ! 0020 * 00000000 00000000 00000000 00000000 *          * 0020 !
_  ! 0030 * 00000000 00000000 00000000 00000000 *          * 0030 !
_  ! 0040 * 00000000 00000000 00000000 00000000 *          * 0040 !
_  +-----+-----+-----+-----+-----+-----+-----+-----+

```

## DBLOG Trace Screen - DL/I Calls

- Invoking DBLOG Trace - DL/L Calls
- DBLOG Trace DL/L Calls - Fields and Functions

### Invoking DBLOG Trace - DL/L Calls

Below is an example of invoking the DBLOG Trace screen for DL/I calls.

1. Write the following Natural program:

```

DEFINE DATA LOCAL
01 COURSE VIEW OF DNDL01-COURSE
  02 COURSEN (A3)
  02 TITLE (A33)
01 OFFERING VIEW OF DNDL01-OFFERING
  02 COURSEN-COURSE (A3)
  02 LOCATION (A31)
END-DEFINE
READ (5) COURSE BY COURSEN
  IF TITLE = 'NATURAL'
    FIND (1) OFFERING WITH COURSEN-COURSE = COURSEN
      MOVE 'DARMSTADT' TO LOCATION
      UPDATE
    END OF TRANSACTION
  END-FIND
END-IF
END-READ
END

```

2. In the command line, enter TEST DBLOG D.  
The message "DBLOG started now" is displayed.
3. In the command line, enter RUN.
4. In the command line, enter TEST DBLOG D again.  
Logging is deactivated and the DBLOG Trace screen for DL/I screen is displayed:

```

08:13:15          ***** NATURAL DBA Utility *****          2002-03-12
User SAG              - DBLOG Trace -              Library SAG
No  Func PCB NS SC DBD/PSB      First SSA (truncated)      IOA (trunc) Program  Line
-----
  1 PCB          PANII41          SAGTEST 0090
  2 GU   1   1   DNDL01  COURSE *--(COURSEN =>      .          SAGTEST 0090
  3 GN   1   1   DNDL01  COURSE *--(COURSEN =>      .Z01       SAGTEST 0090
  4 GN   1   1   DNDL01  COURSE *--(COURSEN =>      .001       SAGTEST 0090
  5 GN   1   1   DNDL01  COURSE *--(COURSEN =>      .004NATURA SAGTEST 0090
  6 GHNP 1   2   DNDL01  COURSE *- (COURSEN =004  ?010791DAR SAGTEST 0110
  7 REPL 1          DNDL01          ?010791DAR SAGTEST 0130
  8 SYNC
  9 PCB          PANII41          SAGTEST 0110
 10 GU   1   1   DNDL01  COURSE *--(COURSEN = 004  .004NATURA SAGTEST 0110
 11 GHNP 1   2   DNDL01  COURSE *--(COURSEN = 004  ?010791DAR SAGTEST 0110
 12 GN   1   1   DNDL01  COURSE *--(COURSEN =>      +110      SAGTEST 0090
***** End of Log *****
    
```

### DBLOG Trace DL/I Calls - Fields and Functions

Following are the fields provided on the DBLOG Trace screen for DL/I calls. For an explanation of the PF keys provided, refer to the description of the DBLOG Trace menu for Adabas above.

Column	Explanation
No	Sequence number. The calls are displayed in the sequence in which they were executed.
Func	DL/I function.
PCB	PCB number.
NS	Number of SSAs.
SC	DL/I status code.
DBD/PSB	DBD name for DB calls. PSB name for scheduling calls.
First SSA	First 25 bytes of the first SSA.
IOA	First 13 bytes of the I/O area.
Program	Natural program name.
Line	Source-code line number.

### DBLOG Trace Screen - SQL Statements

- Invoking DBLOG Trace - SQL Statements
- DBLOG Trace SQL Statements - Fields and Functions

#### Invoking DBLOG Trace - SQL Statements

Below is an example of the Trace Function for SQL statements.

1. Write the following Natural program:

```

DEFINE DATA LOCAL
01 EMP VIEW OF DSN8510-EMP
02 EMPNO
02 FIRSTNME
    
```

```

02 MIDINIT
02 LASTNAME
02 EDLEVEL
02 SALARY
01 EMPPROJACT VIEW OF DSN8510-EMPPROJACT
02 EMPNO
02 PROJNO
02 ACTNO
02 EMPTIME
END-DEFINE
FIND (1) EMP WITH EMPNO> '000300'
  FIND (1) EMPPROJACT WITH EMPNO = EMPNO(0150)
  MOVE 0.75 TO EMPTIME
  UPDATE
END-FIND
ADD 1 TO EDLEVEL
UPDATE
END-FIND
*
FIND (1) EMP WITH EMPNO> '000300'
  FIND (1) EMPPROJACT WITH EMPNO = EMPNO(0240)
  DISPLAY EMPPROJACT EMP.EDLEVEL
END-FIND
END-FIND
ROLLBACK
END
    
```

2. In the command line, enter TEST DBLOG Q.  
The message "DBLOG started now" is displayed.
3. In the command line, enter RUN.
4. In the command line, enter TEST DBLOG Q again.  
Logging is deactivated and the DBLOG Trace screen for SQL statements is displayed:

16:24:27		***** NATURAL Test Utilities *****							2002-04-03			
User SAG		- DBLOG Trace -							Library SAG			
M No	R	SQL Statement (truncated)	CU	SN	SREF	M	Typ	SQLC/W	Program	Line	LV	
--	1	SELECT EMPNO,FIRSTNME,MIDINIT	01	01	0150	D	DB2		SAGTEST	0150	01	
--	2	FETCH CURSOR	01	01	0150	D	DB2		SAGTEST	0150	01	
--	3	SELECT EMPNO,PROJNO,ACTNO,EMP	02	02	0160	D	DB2		SAGTEST	0160	01	
--	4	FETCH CURSOR	02	02	0160	D	DB2		SAGTEST	0160	01	
--	5	UPDATE DSN8510.EMPPROJACT SET	02	03	0160	D	DB2		SAGTEST	0180	01	
--	6	CLOSE CURSOR	02	02	0160	D	DB2		SAGTEST	0160	01	
--	7	UPDATE DSN8510.EMP SET EDLEVE	01	04	0150	D	DB2		SAGTEST	0210	01	
--	8	CLOSE CURSOR	01	01	0150	D	DB2		SAGTEST	0150	01	
--	9	SELECT EMPNO,FIRSTNME,MIDINIT	05	05	0240	D	DB2		SAGTEST	0240	01	
--	10	FETCH CURSOR	05	05	0240	D	DB2		SAGTEST	0240	01	
--	11	SELECT EMPNO,PROJNO,ACTNO,EMP	06	06	0250	D	DB2		SAGTEST	0250	01	
--	12	FETCH CURSOR	06	06	0250	D	DB2		SAGTEST	0250	01	
--	13	CLOSE CURSOR	06	06	0250	D	DB2		SAGTEST	0250	01	
--	14	CLOSE CURSOR	05	05	0240	D	DB2		SAGTEST	0240	01	
--	15	ROLLBACK	00	00	0000	D	DB2		SAGTEST	0290	01	
--												
--												
Command ==>												
Enter-PF1---		PF2---	PF3---	PF4---	PF5---	PF6---	PF7---	PF8---	PF9---	PF10--	PF11--	PF12---
Help		Print	Exit	Top	Posi	Bot	-	+				Canc

## DBLOG Trace SQL Statements - Fields and Functions

The fields provided on the DBLOG Trace screen for SQL statements are shown below. For an explanation of the PF keys provided, refer to the description of the DBLOG Trace menu for Adabas above.

Column	Explanation
M	<p>Input option for line commands:s</p> <p>E Executes the command EXPLAIN which provides information on the DB2 or SQL/DS optimizer's choice of strategy for executing SQL statements.s</p> <p>See also EXPLAIN Command in: Natural System Commands for DB2, Natural for DB2 documentation. Database Management, Natural for SQL/DS documentation.</p> <p>L Executes the command LISTSQL which lists the Natural statements in the source code of an object and the corresponding SQL statements into which they have been translated. An SQL statement is identified by the library name, program name, and line number taken from the Natural debug buffer.</p> <p>See also LISTSQL Command in: Natural System Commands for DB2, Natural for DB2 documentation. Database Management, Natural for SQL/DS documentation.</p> <p><b>Important:</b> Since both commands obtain their information from the Natural system file, unwanted results may occur if the corresponding Natural program has been recataloged after the logging function was executed with the TEST DBLOG Q command. These unwanted results may be caused by statements modified after the logging.</p>
No	Sequence number; the statements are displayed in the sequence in which they were executed.
R	<p>Only applicable if the Natural File Server for DB2 is in use.</p> <p>Indicates by an asterisk in front of the corresponding statement that a reselection has been performed; if not, the column is left blank.</p> <p>See also Concept of the File Server in Natural File Server for DB2 (Natural for DB2 documentation).</p>
SQL Statement	The first 29 characters of the logged SQL statement.
CU	Cursor number.
SN	Internal statement number.
SREF	Statement reference number.
M	Mode: <b>D</b> for dynamic or <b>S</b> for static.
Typ	Database type: DB2 or /DS.
SQLC/W	Either the SQL return code in the SQLCODE field of the SQLCA, or the warning in the SQLWARN0 field of the SQLCA if SQLCODE is 0.

<b>Column</b>	<b>Explanation</b>
Pgm	Natural program name.
Line	Source code line number.
LV	Program level.

# DBLOG - Snapshot Function

The snapshot function provides detailed information on one particular Adabas command, DL/I call or SQL statement.

Below is information on the snapshot function for

- Adabas Commands
  - DL/I Calls
  - SQL Statements
- 

## Snapshot Function - Adabas Commands

This snapshot function interrupts program execution after executing the first Adabas command that matches the selection criteria specified on the DBLOG Menu. The Snapshot Report (example below) generated for the specified Adabas command is displayed immediately after program interruption.

The snapshot function automatically logs **all** Adabas buffers. Therefore, you do not have to mark any of the optional buffers on the DBLOG Menu before you start the snapshot function. The default Snapshot Report displays the Adabas control block (CB).

Below is information on:

- Invoking Snapshot - Adabas Commands
- Displaying Buffers on the Snapshot Report

## Invoking Snapshot - Adabas Commands

### To invoke the Snapshot Report screen for Adabas commands

- On the DBLOG Menu, specify an Adabas command and additional criteria, if desired, and enter Function Code **S**.  
The message "DBLOG snapshot facility started now" is displayed.
- Execute a Natural program which contains the Adabas command specified on the DBLOG Menu.  
The program stops executing and the Snapshot Report for Adabas commands is displayed as shown in the example below:

```

16:36:39          ***** NATURAL TEST UTILITIES *****          2002-03-11
                    - Snapshot Report -

Command Code : L3          Command ID   :   ??? 00200101 File Number : 013C
Response Code:      0          ISN       :           1300
ISN Low Limit: 00000000    ISN Quantity :           0
FB Length   : 0009          RB Length   : 0014          SB Length   : 0008
VB Length   : 0014          IB Length   : 0000          Com. Option 1:
Com. Option 2: V          Additions 1  : AE]?          Additions 2  : ? ?
Additions 3  :           Additions 4  :
Global FID   : 0000000000000000 Command Time : 00000019 Pgm: SAGTEST Lin: 0020
Control Block
0000 * 30D5D3F3 00200101 013C0000 00000514 * ?NL3 ?????   ?? * 0000
0010 * 00000000 00000000 00090014 00080014 *           ? ? ? ? * 0010
0020 * 000000E5 C1C5BBCA 40404040 00120014 *   VAE]?   ? ? * 0020
0030 * 00000000 00000000 00000000 00000000 *           * 0030
0040 * 00000000 00000000 00000019 00000000 *           ?   * 0040
0050 * 00000000 00000000 00000000 00000000 *           * 0050
0060 * 00000000 00000000 00000000 00000000 *           * 0060
0070 * 00000000 00000000 00000000 00000000 *           * 0070

Command ==> CB
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      Help      Exit  CB   FB   RB   -   +   SB   VB   IB   Canc
    
```

### Displaying Buffers on the Snapshot Report

The Snapshot Report shows the Adabas control block (CB) by default. To display different Adabas buffers and page up or down the screen, choose any of the direct commands or PF keys listed below:

PF Key	Direct Command	Buffer
PF4	CB	Displays the Control Block. This is the default.
PF5	FB	Displays the Format Buffer.
PF6	RB	Displays the Record Buffer.
PF7	-	Scrolls up the screen to display long buffers logs that extend beyond the terminal screen.
PF8	+	Scrolls down the screen to display long buffers logs that extend beyond the terminal screen.
PF9	SB	Displays the Search Buffer.
PF10	VB	Displays the Value Buffer.
PF11	IB	Displays the ISN Buffer.

### Snapshot Function - DL/I Calls

This snapshot function generates the Snapshot Report (example below) of the first DL/I call that matches the selection criteria specified on the DBLOG Menu. A snapshot does not interrupt the program flow. The snapshot data are kept in the Natural debug buffer to be displayed only if the user enters the appropriate DBLOG command as described below.

Below is information on:



- Status Code
- Processing Options
- Segment Name
- Length of the Key Feedback Area
- Number of Sensitive Segments
- Key Feedback Area
- Number of SSAs (Segment Search Argument)
- all SSAs
- the I/O Area

The first 120 bytes of the Key Feedback Area, of all SSAs (up to 15 SSAs are possible) and of the I/O area are displayed, both in decimal and hexadecimal format.

The DBD Name in the PCB is used to read the corresponding NDB (Natural equivalent of DBD) from the Natural FDIC system file. In this NDB, the segment whose name is given in the PCB is located and its minimum/maximum length and segment level number are displayed. The segment level number should match the number in the PCB. In this way, it is possible to detect inconsistencies between Natural NDBs and DL/I DBDs.

The PSB name is used to read the corresponding NSB (Natural equivalent of PSB) from the Natural FDIC system file. From this NSB, the number of sensitive segments is displayed. This number should match the number in the PCB. In this way, it is possible to detect inconsistencies between Natural NSBs and DL/I PSBs.

The snapshot function checks whether the DL/I DBD/PSB and the Natural NDB/NSB contain the same values in the fields "Level Number" and "Number of SENSEGs". The same values, however, do not necessarily ensure that the DL/I DBD/PSB and the Natural NDB/NSB are fully consistent.

In the example above, the values in the "Number of SENSEGs" (Sensitive Segment Type) fields are different, because the Natural NATPSB procedure was not executed after the PSB had been changed by the DL/I PSBGEN procedure.

## Snapshot Function - SQL Statements

The snapshot function generates the Snapshot Report (example below) of the first SQL statement that matches the selection criteria specified on the DBLOG Menu. A snapshot does not interrupt the program flow.

Unlike the statements displayed with the DBLOG Trace function, the snapshot shows the statement in its entirety (limited to 13 lines).

The snapshot data are kept in the Natural debug buffer to be displayed only if the user enters the appropriate DBLOG command as described below.

Below is information on:

- Invoking Snapshot - SQL Statements
- Snapshot Report Information - SQL Statements

### Invoking Snapshot - SQL Statements

#### To invoke the Snapshot Report screen for SQL statements

- On the DBLOG Menu, specify an SQL statement and additional criteria, if desired, and enter Function Code S.  
The message "DBLOG snapshot facility started now" is displayed.
- Execute a Natural program which contains the SQL statement specified on the DBLOG Menu.  
(Log data are written to the Natural debug buffer.)

- Display the snapshot data:  
 In the command line, enter TEST DBLOG Q.  
 Or, on the DBLOG Menu, enter Function Code E.  
 The Snapshot Report for SQL statements is displayed as shown in the example below:

```

10:59:28          ***** NATURAL Test Utilities *****          2002-04-08
User SAG              - Snapshot Report -              Library SAG

CU SN M Typ R SQLC/W      Library Program Store Clock Value Line LV CID(Hex)
01 01 D DB2             SAG      SAGTEST  2002/04/03 14:23:06 0150 01 01500101

SQL Statement
SELECT EMPNO,FIRSTNME,MIDINIT, LASTNAME, EDLEVEL, SALARY FROM DSN8510.EMP WHERE EM
PNO> '000300' FOR UPDATE OF EDLEVEL

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

### Snapshot Report Information - SQL Statements

The following information is provided on the Snapshot Report screen for SQL statements:

Item	Explanation
CU	Cursor number.
SN	Internal statement number.
M	Mode: <b>D</b> for dynamic or <b>S</b> for static.
Typ	Database type: DB2 or SQL/DS.
R	Only applicable if the Natural File Server for DB2 is in use.  Indicates by an asterisk in front of the corresponding statement that a reselection has been performed; if not, the column is left blank.  See also Concept of the File Server in Natural File Server for DB2 (documentation Natural for DB2).
SQLC/W	Either the SQL return code in the SQLCODE field of the SQLCA, or the warning in the SQLWARN0 field of the SQLCA if SQLCODE is 0.
Library	The library where the Natural program with the logged statement was cataloged.
Program	The name of the Natural program which contains the logged statement.
Store Clock Value	The time stamp of the Natural program which contains the logged statement.
Line	The source code line number of the logged statement.
LV	The call level of the Natural program which contains the logged statement.
CID (hex)	The command ID of the logged statement in hexadecimal format.

# DBLOG Direct Commands

The direct command TEST DBLOG can be used to execute DBLOG and display or delete the log records currently stored in the Natural debug buffer. Note that TEST DBLOG does not provide any parameters to specify selection criteria. Selection criteria can only be specified with the DBLOG Menu.

The parameters that apply to the command TEST DBLOG are explained in the syntax diagrams and tables below. There are parameters that can be used to do both, activate and deactivate DBLOG (toggle effect). Activating and deactivating depends on whether or not there is data stored in the Natural debug buffer as described in Parameters below.

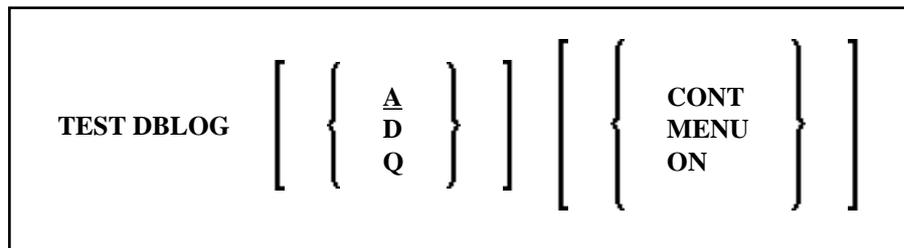
For an explanation of the symbols used in the syntax diagrams, refer to System Command Syntax in the Natural System Command Reference documentation.

Below is information on:

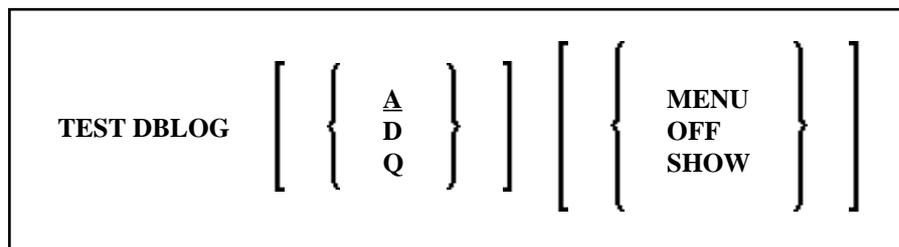
- Syntax Diagrams
- Parameters

## Syntax Diagrams

### Activating DBLOG



### Deactivating DBLOG



### Parameters

Parameter	Function
A	<p>Default value.</p> <p>Toggle function:</p> <p>Activates logging of Adabas commands if no data exist in the Natural debug buffer.</p> <p>Deactivates logging of Adabas commands and displays the DBLOG Trace screen of the current log record if data exist in the Natural debug buffer.</p>
D	<p>Toggle function:</p> <p>Activates logging of DL/I calls if no data exist in the Natural debug buffer.</p> <p>Deactivates logging of DL/I calls and displays the DBLOG Trace screen of the current log record if data exist in the Natural debug buffer.</p>
Q	<p>Toggle function:</p> <p>Activates logging of SQL statements if no data exist in the Natural debug buffer.</p> <p>Deactivates logging of SQL statements and displays the DBLOG Trace screen of the current log record if data exist in the Natural debug buffer.</p>
CONT	Activates or reactivates (restarts) logging. A restart causes DBLOG to continue logging with the next program executed or run after DBLOG execution was stopped, and to add the new logs to the data that exist from previous recordings.
MENU	Invokes the DBLOG Menu which provides the options to activate or deactivate logging and to specify the commands, calls or statements to be logged; see the relevant section.
?	
*	
SHOW	Deactivates logging and displays the DBLOG Trace screen of the current log record if data exists in the Natural debug buffer. Log record data are not deleted but kept in the Natural debug buffer.
ON	Clears the Natural debug buffer and activates logging.
OFF	Clears the Natural debug buffer and deactivates logging.

# Natural Debugger - Overview

The Natural Debugger can be used to locate errors in a Natural object.

Note that the Natural Debugger documentation is part of the Natural Utilities documentation.

The Natural Debugger documentation covers the following topics:

- Concepts of the Natural Debugger
- Start the Natural Debugger
- Set Test Mode ON/OFF
- Debug Environment Maintenance
- Spy Maintenance
- Breakpoint Maintenance
- Watchpoint Maintenance
- Call Statistics Maintenance
- Statement Execution Statistics Maintenance
- Variable Maintenance
- List Object Source
- Execution Control Commands
- Navigation and Information Commands
- Command Summary and Syntax

# Concepts of the Natural Debugger

The Natural Debugger can be used to temporarily take over control of a Natural session for debugging purposes while a Natural object is executing.

The execution flow of an application is not influenced by the Natural Debugger being applied to it; that is, the application itself need not be adapted to or prepared for its being debugged. By setting debug entries in a Natural object, you can follow the processing flow of the program. The Natural Debugger receives control at any debug entry set, thus allowing various program investigations. This helps you to understand poorly written or poorly documented programs, and to identify redundant or dead (never gets executed) programming code. This can be useful, for example, when you take over an application from another developer to maintain, improve or expand it.

In addition, when you write an interface to a Natural program, you can use the Natural Debugger to check the contents of variables at the point when parameters are passed between the Natural program and the interface and thus make sure that parameters are passed correctly.

When Test Mode is set to ON (see Set Test Mode ON/OFF), the Natural Debugger receives control whenever a Natural error occurs during the execution of a Natural object, irrespective of any debug entries defined. You can then, for example, review the contents of the variables in the Natural object to determine the reason for the error.

Below is information on:

- Debug Entries/Spies
  - Debug Window
- 

## Debug Entries/Spies

Debug entries are also referred to as spies in the Natural Debugger environment. Two types of debug entries (spies) are available: breakpoints and watchpoints. Debug entries for the current debug session can be set, modified, listed, displayed, activated, deactivated and deleted. Debug entries can also be saved for future use as described in Debug Environment Maintenance.

When a debug entry is set or modified, Natural internally stores the library, database ID and file number where the object is located. The object may be located in the current library or in one of its steplibs. If an object of the same name is later executed from another library, the corresponding debug entry is not executed.

The Natural Debugger assigns a name and a unique number (Spy Number) to each debug entry. The name assigned to a debug entry (also referred to as Spy Name) can be either a name specified by the user or a default name created by the Natural Debugger. A debug entry can be selected by its number with the corresponding Natural Debugger commands. If more than one debug entry has to be executed at a specific statement line, they are executed in ascending order of their numbers.

Each debug entry has an initial state and a current state. Possible values are **A** (active) and **I** (inactive). The initial value is specified when setting or modifying the breakpoint or watchpoint and determines the state of the debug entry at environment start or after reset. During the debug session, the state can be changed with the debug commands **ACTIVATE** and **DEACTIVATE** (see also the syntax diagrams in Command Summary and Syntax).

Each debug entry has an event count, which is increased every time the debug entry is executed. A debug entry is not executed if the current state is "inactive". The event count of the breakpoint or watchpoint is not increased either.

The number of executions of a debug entry can be restricted in two ways:

- A number of skips can be specified before the debug entry is executed. The debug entry is then ignored until the event count is higher than the number of skips specified.
- A maximum number of executions can be specified, so that the debug entry is ignored, as soon as the event count exceeds the specified number of executions.

For each debug entry (breakpoint or watchpoint), up to six debug commands can be specified. These commands are executed at execution time of the breakpoint or watchpoint. You can use all Natural Debugger commands that can be applied during a debug interrupt. The default command is the BREAK command, which displays the Debug Window, as shown below.

**Attention:**

If you delete the BREAK command when setting a debug entry and you do not enter any command that issues a dialog, there is no way to assume control during program interruption.

## Debug Window

When the Natural Debugger receives control of the session, the Debug Window is displayed.

The Debug Window shows the type and name of the debug entry that has caused the break (that is, the name of the corresponding breakpoint or watchpoint), its source-code line number, and the name of the interrupted Natural object. For a detailed description of the functions listed below see Execution Control Commands.

From the Debug Window, you can select the following functions:

Function	Code	Description
Go	G	Continues the execution of the Natural object up to the next debug entry specified.
List Break	L	Lists the programming code currently active. The last statement executed is highlighted.
Debug Main Menu	M	Invokes the Debug Main Menu which provides all functions needed to maintain debug entries at which control is to be assumed.
Next	N	Executes the next command specified for the current breakpoint or watchpoint.
Run	R	Continues the execution of the Natural object with test mode OFF.
Step	S	Continues the execution of the Natural object in step mode.
Variable Function	V	Displays the program variables currently active and modifies the contents of these variables.

# Start the Natural Debugger

Below are basic operational requirements and a rough guideline on how to proceed when planning to apply the Natural Debugger:

- Object Requirements
  - Invoke the Natural Debugger
  - Default Object
- 

## Operation Requirements

The Natural Debugger can only be applied to stowed or cataloged Natural objects. It cannot be applied to objects executed with the RUN command.

The Natural Debugger investigates all Natural objects stored in the current library or in one of its steplibs. The Natural Debugger does not investigate Natural objects stored in the Natural system library SYSLIB or SYSLIBS.

The Natural Debugger can only be applied to objects of Natural Version 2.3 and above, but not to objects stowed with any previous version.

## Batch Processing

Although the Natural Debugger is mainly designed for interactive usage in online mode, the Natural Debugger commands can also be used for batch execution, such as for setting breakpoints or watchpoints.

### Note:

There are restrictions for batch processing which can cause a Natural Debugger command to be rejected. For example, the Natural Debugger does not support the commands ++ and +4.

## Invoke the Natural Debugger

### To invoke the Natural Debugger

1. Establish a debug environment for a Natural object or application:
  - Invoke the Debug Main Menu by entering the direct command TEST.  
Or, from within a running application, enter the terminal command %<TEST.
2. Use the functions of the Debug Main Menu to specify debug entries for a Natural object or application:
  - Debug Environment Maintenance
  - Spy Maintenance
  - Breakpoint Maintenance
  - Watchpoint Maintenance
  - Call Statistics Maintenance
  - Statement Execution Statistics Maintenance
  - Variable Maintenance
  - List Object Source
3. Activate the Natural Debugger:
  - In the direct command line or in the NEXT line, enter TEST ON.  
Or, on the Debug Main Menu, enter Function Code T.
4. Execute the Natural object or application.

## Default Object

The maintenance functions of the Natural Debugger as described in the relevant sections, refer to objects you specify either in the corresponding name fields of menus or with direct commands. If you do not specify an object name, by default, the Natural Debugger assumes the name of the current object as it is displayed in the field Object, in the upper right corner of the screen. With a default object defined, no object name is required in direct commands and menu options used to specify breakpoints or watchpoints. To change the default object, see the syntax of the command SET in the section Command Summary and Syntax.

# Set Test Mode ON/OFF

To activate a previously established debug environment, test mode must be switched on.

 **To set Test Mode to ON or OFF**

- On the Debug Main Menu, enter Function Code **T** to activate and deactivate.  
Or enter the direct command TEST ON or TEST OFF.

When executing a Natural object with test mode ON, the Natural debugger continuously checks all debug entries for any required action.

When executing a Natural object with test mode OFF, all debug entries are ignored.

The command TEST, and with it the whole application, can be protected by Natural Security. Individual functions within the Natural Debugger, however, cannot be protected.

# Debug Environment Maintenance

Since a debug environment mainly consists of debug entries, it is established by setting breakpoints and watchpoints, as described in the relevant maintenance sections.

Once established, a debug environment can be stored for subsequent usage. The file where debug environments are stored can be set with the direct command PROFILE (see Navigation and Information Commands). You can also delete a debug environment or reset its counters to their initial values.

The following items are also part of a debug environment and are therefore saved or loaded every time you save or load a debug environment:

- Test Mode setting (ON/OFF);
- all options that can be set with the direct command PROFILE except the file for loading or saving debug environments;
- the settings of the statement execution statistics (ON/OFF/COUNT).

You can perform all these functions using the Debug Environment Maintenance menu, which is invoked by entering Function Code **E** on the Debug Main Menu.

## Note:

The Natural Debugger supports only debug environments which were created with Natural versions higher than Version 2.3; debug environments created with any previous version will be ignored.

The functions provided with the Debug Environment Maintenance menu are listed below. With each function selected, you must enter the name of the debug environment to be maintained:

- Set Test Mode ON/OFF  
(see the relevant section)
  - Load Debug Environment
  - Save Debug Environment
  - Reset Debug Environment
  - Delete Debug Environment
- 

## Load Debug Environment

### To load a debug environment from your user system file (FUSER)

- On the Debug Environment Maintenance menu, enter Function Code **L** and the name of an environment. Or enter the direct command LOAD ENVIRONMENT *name*. The specified debug environment is loaded.

If you do not specify a name, the default environment with the name Noname is loaded.

Enter an asterisk (\*) to obtain a list all available debug environments. On the list, you can mark the desired environment with the line command LO to load it into the debug buffer, or with the line command DE to delete it.

## Save Debug Environment

### To save a debug environment

- On the Debug Environment Maintenance menu, enter Function Code **S** and the name of an environment.  
Or enter the direct command `SAVE ENVIRONMENT name`.  
The specified environment is reset (see below) and saved to the file location specified with the `PROFILE` command as mentioned earlier.

If you do not specify a name, the environment is saved with the name Noname.

If a debug environment with the specified name already exists, you are prompted for confirmation to overwrite the old environment.

## Reset Debug Environment

The debug environment should be reset before each test run. Resetting the environments leads to the following results:

- The current states of all debug entries are set to their initial states,
- All event counts are set to zero,
- The call statistics in the debug buffer are cleared as described in the section Call Statistics Maintenance.

### To reset a debug environment

- On the Debug Environment Maintenance menu, enter Function Code **R** and the name of an environment.  
Or enter the direct command `RESET ENVIRONMENT name`.  
The specified debug environment is reset.

If you do not specify an environment name, the current debug environment is reset.

## Delete Debug Environment

### To delete a debug environment

1. On the Debug Environment Maintenance menu, enter Function Code **D** and the name of the environment.  
Or enter the direct command `DELETE ENVIRONMENT name`.  
The confirmation window appears.
2. In the confirmation window, enter Y (Yes) to confirm the deletion.  
The debug specified environment is deleted.

If you do not specify an environment name, the current debug environment is deleted.

# Spy Maintenance

The Spy Maintenance functions are used to activate, deactivate, list or delete **all** debug entries (spies) that is, breakpoints **and** watchpoints. Besides, Spy Maintenance is an alternative method of accessing the breakpoint or watchpoint maintenance screens.

## ▶ To invoke the Spy Maintenance menu

- On the Debug Main Menu, enter Function Code **S**.  
Or enter the direct command **SM**.
- The Spy Maintenance menu appears.

The Spy Maintenance menu provides the following functions:

- Set Test Mode ON/OFF  
(see the relevant section)
  - Activate Spy
  - Deactivate Spy
  - Delete Spy
  - Display Spy
  - Modify Spy
- 

## Activate Spy

### ▶ To set the current state of specified spies to "active"

- On the Spy Maintenance menu, enter Function Code **A** and a spy number **or** a spy name.  
Or enter the direct command `ACTIVATE SPY name or number`  
(see also the syntax of `ACTIVATE` in Command Summary and Syntax).

If you do not specify an object name or a line number, **all** spies (breakpoints and watchpoints) are activated.

## Deactivate Spy

### ▶ To set the current state of specified spies to "inactive"

- On the Spy Maintenance menu, enter Function Code **B** and a spy number **or** a spy name.  
Or enter the direct command `DEACTIVATE SPY name or number`  
(see also the syntax of `DEACTIVATE` in Command Summary and Syntax).

If you do not specify an object name or a line number, **all** spies (breakpoints and watchpoints) are deactivated.

## Delete Spy

### ▶ To delete specified spies

- On the Spy Maintenance menu, enter Function Code **C** and a spy number **or** a spy name.  
Or enter the direct command `DELETE SPY name or number`  
(see also the syntax of `DELETE` in Command Summary and Syntax).

If you do not specify an object name or a line number, **all** spies (breakpoints and watchpoints) are deleted.

## Display Spy

### ▶ To display specified spies

- On the Spy Maintenance menu, enter Function Code **D** and a spy number **or** a spy name.  
Or enter the direct command `DISPLAY SPY name or number`  
(see also the syntax of `DISPLAY` in Command Summary and Syntax).

If the specified spy is unique, the Display Breakpoint or Display Watchpoint screen (see the relevant sections) appears respectively and all specifications of this breakpoint or watchpoint are displayed.

If the specified spy is not unique, a list of the spies concerned is displayed. On the list, you can activate, deactivate, display, modify or delete a spy by marking it with the line command `AC`, `DA`, `DI`, `MO` or `DE` respectively.

If you do not specify an object name or a line number, **all** spies (breakpoints and watchpoints) are displayed.

## Modify Spy

### ▶ To modify specified spies

- On the Spy Maintenance menu, enter Function Code **M** and a spy number **or** a spy name.  
Or enter the direct command `MODIFY SPY name or number`  
(see also the syntax of `MODIFY` in Command Summary and Syntax).

If the specified spy is unique, the Modify Breakpoint or Modify Watchpoint screen appears respectively and the breakpoint or watchpoint specifications can be modified.

If the specified spy is not unique, a list of the spies concerned is displayed. On the list, you can activate, deactivate, display, modify or delete a spy by marking it with the line command `AC`, `DA`, `DI`, `MO` or `DE` respectively.

If you do not specify an object name or a line number, **all** spies (breakpoints and watchpoints) are displayed for selection and modification.

# Breakpoint Maintenance

A breakpoint causes the execution of a Natural object to be interrupted at a predefined statement line. This section describes how and when to set breakpoints. Note that the maintenance functions described here may also be invoked from an object source by using the function List Object Source.

Below is information on:

- Conditions of Use
  - Activate Breakpoint
  - Deactivate Breakpoint
  - Delete Breakpoint
  - Display Breakpoint
  - Modify Breakpoint
  - Set Breakpoint
- 

## Conditions of Use

A breakpoint is defined by specifying the name of the Natural object to be processed and the line number of the object's source code where the breakpoint is to be executed.

Once a breakpoint has been specified, it remains set for the entire Natural session, unless you delete it.

A breakpoint refers to a specific line number in a source code. A subsequent change of the source code itself may therefore lead to the breakpoint no longer applying to the desired statement, and thus the Natural object not being interrupted at the desired position. To circumvent this problem with program loops, labels can be set within these loops. Breakpoints set for these labels are adjusted to the correct line number if statement lines are inserted or deleted.

The unique identifier for a breakpoint is the Spy Number as assigned by the Natural Debugger.

Breakpoints cannot be set in copycode, on comment lines, on any statement line other than the first one (if a single statement occupies more than one program line), and on lines that contain one of the following statements only:

- AT BREAK OF
- AT END OF DATA
- AT END OF PAGE
- AT START OF DATA
- AT TOP OF PAGE
- BEFORE BREAK
- DECIDE
- DEFINE SUBROUTINE
- DEFINE WINDOW
- FORMAT
- IF NO RECORDS FOUND
- ON ERROR
- OPTIONS

For objects cataloged in Natural versions earlier than Version 4.1, breakpoints can only be set on lines with WHEN and VALUE clauses. These restrictions do not apply to objects cataloged in Natural Version 4.1 and above.

Whether it is possible or not to set breakpoints for lines compiled with the Natural Optimizer Compiler depends on the NODBG option of the OPTIONS statements as described in "Switching on the Optimizer Compiler" in the Natural Optimizer Compiler documentation.

## Activate Breakpoint

### ▶ To set the current state of specified breakpoints to "active"

- On the Breakpoint Maintenance menu, enter Function Code **A**, an object name and/or a line number.  
Or enter the direct command `ACTIVATE BREAKPOINT object and/or line`  
(see also the syntax of `ACTIVATE` in Command Summary and Syntax).

If you do not specify an object name or a line number, **all** breakpoints are activated.

## Deactivate Breakpoint

### ▶ To set the current state of specified breakpoints to "inactive"

- On the Breakpoint Maintenance menu, enter Function Code **B**, an object name and/or a line number.  
Or enter the direct command `DEACTIVATE BREAKPOINT object and/or line`  
(see also the syntax of `DEACTIVATE` in Command Summary and Syntax).

If you do not specify an object name or a line number, **all** breakpoints are deactivated.

## Delete Breakpoint

### ▶ To delete specified breakpoints

- On the the Breakpoint Maintenance menu, enter Function Code **C**, an object name and/or a line number.  
Or enter the direct command `DELETE BREAKPOINT object and/or line`  
(see also the syntax of `DELETE` in Command Summary and Syntax).

If you do not specify an object name or a line number, **all** breakpoints are deleted.

## Display Breakpoint

### ▶ To display specified breakpoints

- On the Breakpoint Maintenance menu, enter Function Code **D**, an object name and/or a line number.  
Or enter the direct command `DISPLAY BREAKPOINT object and/or line`  
(see also the syntax of `DISPLAY` in Command Summary and Syntax).

If a unique breakpoint has been specified, the Display Breakpoint screen appears and all specifications of this breakpoint are displayed. The Display Breakpoint screen is identical to the Modify Breakpoint screen. For an explanation of the fields, see Display/Modify Breakpoint Screen below.

If no unique breakpoint has been specified, a list displays all breakpoints set for the current environment. On the list, you can activate, deactivate, display, modify or delete a breakpoint by marking it with the line command AC, DA, DI, MO or DE respectively.

If you do not specify an object name or a line number, **all** breakpoints are displayed.

## Modify Breakpoint

### To modify specified breakpoints

1. On the Breakpoint Maintenance menu, enter Function Code **M**, an object name and/or a line number.  
Or enter the direct command `MODIFY BREAKPOINT object and/or line`  
(see also the syntax of `MODIFY` in Command Summary and Syntax).  
If a unique breakpoint has been specified, the Modify Breakpoint screen appears and the breakpoint specifications can be modified. The Modify Breakpoint screen is identical to the Display Breakpoint screen.  
For an explanation of the fields, see Display/Modify Breakpoint Screen below.  
If no unique breakpoint has been specified, a list displays all breakpoints set for the current environment.  
On the list, you can activate, deactivate, display, modify or delete a breakpoint by marking it with the line command `AC`, `DA`, `DI`, `MO` or `DE` respectively.
2. On the Modify Breakpoint screen, choose `PF3/Exit` or `PF5/Save` to save any modification.  
If you choose `PF12/Canc`, the breakpoint remains unchanged.

If you do not specify an object name or a line number, **all** breakpoints are displayed for selection and modification: see Display Breakpoint above.

Below is information on:

- Display/Modify Breakpoint Screen

### Display/Modify Breakpoint Screen

The Modify Breakpoint screen provides the following fields:

Field	Explanation						
Spy Number	A unique number assigned by the Natural Debugger when setting the breakpoint.						
Initial State	Specifies the current state of the breakpoint: active or inactive.						
Breakpoint Name	The name of the breakpoint. Valid input: range from 1 to 12 characters. The default name for breakpoints consists of the object name and the line number.						
Object Name	The name of an object available in the current library or one of its steplibs. The default name is the name of the default object (see the section Start the Natural Debugger) if defined.						
Line Number	The line number of a statement in the object source code. See also Conditions of Use above. You can also specify BEG, END or ALL as line numbers:  <table border="0"> <tr> <td style="padding-right: 20px;">BEG</td> <td>The breakpoint is to be executed at the first statement executed.</td> </tr> <tr> <td>END</td> <td>The breakpoint is to be executed at the last statement executed.</td> </tr> <tr> <td>ALL</td> <td>The breakpoint is to be executed for each program line of the object specified.</td> </tr> </table>	BEG	The breakpoint is to be executed at the first statement executed.	END	The breakpoint is to be executed at the last statement executed.	ALL	The breakpoint is to be executed for each program line of the object specified.
BEG	The breakpoint is to be executed at the first statement executed.						
END	The breakpoint is to be executed at the last statement executed.						
ALL	The breakpoint is to be executed for each program line of the object specified.						
Label	Refers to a label set earlier in the source code of an object for statements that define processing loops: see also Conditions of Use above. Valid input: range from 1 to 32 characters.						
Skips before Execution	Determines that the breakpoint is not to be executed until the corresponding statement line has been executed a certain number of times. Valid input: range from 0 (default) to 32767.						
Max Number Executions	Any value greater than zero (0) determines the maximum number of breakpoint executions. Valid input: range from 0 (default) to 32767.						
Number of Activations	Indicates how many times a breakpoint was activated for the relevant statement line. The counter is reset when a Natural object is started at Level 1.						
Commands	Up to six debug commands. Enter one command per line. For a summary of all available commands, see Command Summary and Syntax.  <b>Attention:</b> If you delete the command BREAK when modifying a breakpoint and you do not enter any command that issues a dialog, there is no way for the Natural Debugger to receive control during program interruption.						

## Set Breakpoint

 To add a breakpoint to a session

- On the Breakpoint Maintenance menu, enter Function Code **S**, an object name and/or a line number.

Or enter the direct command `SET BREAKPOINT object and/or line`  
(see also the syntax of SET in Command Summary and Syntax).

If object name and line number are specified correctly, the breakpoint is set immediately and a corresponding confirmation message is displayed on the screen. The breakpoint receives the default command (BREAK), its initial and current state are set to "active" and no execution restrictions are specified. Note that if you delete the command BREAK when setting a breakpoint and you do not enter any command that issues a dialog, there is no way for the Natural Debugger to receive control during program interruption.

If you specify not an object name but a valid line number, the name of the default object (see the section Start the Natural Debugger) is assumed and the breakpoint is also set immediately. If there is no default object defined, a selection window appears that displays all objects available in the current library.

# Watchpoint Maintenance

A watchpoint causes the execution of a Natural object to be interrupted whenever the value of a variable changes. In addition, you can make the interruption dependent on a condition related to a specific variable value as described under Watchpoint Operators (Set Watchpoint) below.

The use of watchpoints allows you to detect unintended alterations of variables caused by objects that contain errors.

A variable is considered to have changed either when its current value differs from the value recorded when the watchpoint was last triggered or when it differs from the initial value. Comparative validation of watchpoint values is restricted to a field length of 253 bytes. For large variables that exceed the maximum length, only the first 253 bytes are used in the comparison.

A watchpoint is defined by specifying the name of the Natural object or GDA (global data area) to be processed and the local or global variable it is to refer to.

The unique identifier for a breakpoint is the Spy Number as assigned by the Natural Debugger.

Once a watchpoint has been specified, it remains set for the entire Natural session, unless you delete it.

Below is information on:

- Invoke Watchpoint Maintenance
  - Activate Watchpoint
  - Deactivate Watchpoint
  - Delete Watchpoint
  - Display Watchpoint
  - Modify Watchpoint
  - Set Watchpoint
- 

## Invoke Watchpoint Maintenance

### To invoke the Watchpoint Maintenance menu

- On the Debug Main Menu, enter Function Code **W**.  
Or enter the direct command **WM**.

The Watchpoint Maintenance menu provides the following functions:

- Set Test Mode ON/OFF  
(see the relevant section)
- Activate Watchpoint
- Deactivate Watchpoint
- Delete Watchpoint
- Display Watchpoint
- Modify Watchpoint
- Set Watchpoint

## Activate Watchpoint

### ▶ To set the current state of specified watchpoints to "active"

- On the Watchpoint Maintenance menu, enter Function Code **A**, an object name and a variable name.  
Or enter the direct command `ACTIVATE WATCHPOINT object and variable`  
(see also the syntax of `ACTIVATE` in Command Summary and Syntax).

If you do not specify an object or a variable (or leave the default asterisk in the field Variable), **all** watchpoints are activated.

## Deactivate Watchpoint

### ▶ To set the current state of specified watchpoints to "inactive"

- On the the Watchpoint Maintenance menu, enter Function Code **B**, an object name and a variable name.  
Or enter the direct command `DEACTIVATE WATCHPOINT object and variable`  
(see also the syntax of `DEACTIVATE` in Command Summary and Syntax).

If you do not specify an object name or a variable (or leave the default asterisk in the field Variable), **all** watchpoints are deactivated.

## Delete Watchpoint

### ▶ To delete specified watchpoints

- On the Watchpoint Maintenance menu, enter Function Code **C**, an object name and a variable name.  
Or enter the direct command `DELETE WATCHPOINT object and variable`  
(see also the syntax of `DELETE` in Command Summary and Syntax).

If you do not specify an object name or a variable (or leave the default asterisk in the field Variable), **all** watchpoints are deleted.

## Display Watchpoint

### ▶ To display specified watchpoints

- On the Watchpoint Maintenance menu, enter Function Code **D**, an object name and/or a variable name.  
Or enter the direct command `DISPLAY WATCHPOINT object and/or variable`  
(see also the syntax of `DISPLAY` in Command Summary and Syntax).

If a unique watchpoint has been specified, the Display Watchpoint screen appears and all specifications of this watchpoint are displayed. The Display Watchpoint screen is identical to the Modify Watchpoint screen. For an explanation of the fields, see Display/Modify Watchpoint Screen below.

If no unique watchpoint has been specified, a list displays all watchpoints set for the current environment. On the list, you can activate, deactivate, display, modify or delete a watchpoint by marking it with the line command AC, DA, DI, MO or DE respectively.

If you want to know whether, or what kind of, a condition for the watchpoint to be activated has been specified, you can display the corresponding value and watchpoint operator (see below) by choosing either PF10 (if you want to display the value in alphanumeric format) or PF11 (if you want to display the value in hexadecimal format). PF22 takes you back to the Display Watchpoint screen.

If you do not specify an object name or a variable (or leave the default asterisk in the field Variable), **all** watchpoints are displayed.

## Modify Watchpoint

### ▶ To modify specified watchpoints

- On the Watchpoint Maintenance menu, enter Function Code **M**, an object name and a variable name. Or enter the direct command `MODIFY WATCHPOINT object and a variable name` (see also the syntax of `MODIFY` in Command Summary and Syntax).

If a unique watchpoint has been specified, the Modify Watchpoint screen (see below) appears and the watchpoint specifications can be modified. The Modify Watchpoint screen is identical to the Display Watchpoint screen. For an explanation of the fields, see Display/Modify Watchpoint Screen below.

If no unique watchpoint has been specified, a list displays all watchpoints set for the current environment. On the list, you can activate, deactivate, display, modify or delete a watchpoint by marking it with the line command AC, DA, DI, MO or DE respectively.

If you want to modify the condition for the watchpoint to be activated, you can modify the corresponding value and watchpoint operator (see below) by choosing either PF10 (if you want to modify the value in alphanumeric format) or PF11 (if you want to modify the value in hexadecimal format). PF22 takes you back to the Modify Watchpoint screen (see Display/Modify Watchpoint Screen below).

- On the Modify Watchpoint screen (see Display/Modify Watchpoint Screen below), choose PF3/Exit or PF5/Save to save any modification. If you choose PF12/Canc, the watchpoint remains unchanged.

If you do not specify an object name or a variable (or leave the default asterisk in the field Variable), **all** watchpoints are displayed for selection and modification.

Below is information on:

- Display/Modify Watchpoint Screen

## Display/Modify Watchpoint Screen

Field	Explanation
Spy Number	A unique number assigned by the Natural Debugger when setting the watchpoint.
Initial State	Specifies the initial state of the watchpoint: active or inactive. Default: active
Watchpoint Name	The name of the watchpoint.  The default name for watchpoints is the name of the variable concerned. Names that exceed the field size will be truncated after 12 characters.  Valid input: range from 1 to 12 characters.
Object Name	The name of an object available in the current library or one of its steplibs.

Variable Name	<p>The name of a user-defined, global or system variable. If the variable is part of a structure, it may be prefixed by the structure name.</p> <p>For an array, an index description has to be specified (watchpoints can be defined for single elements only).</p> <p>Use Asterisk (*) notation for a selection list of variables.</p> <p>See also Variable Maintenance for further details.</p>
Skips before Execution	<p>Determines that the watchpoint is not to be executed until the condition set for the watchpoint has been fulfilled (see also Watchpoint Operators below).</p> <p>Valid input: range from 0 (default) to 32767.</p>
Max Number Executions	<p>Any value greater than zero (0) determines the maximum number of watchpoint executions.</p> <p>Valid input: range from 0 (default) to 32767.</p>
Number of Activations	<p>Indicates how many times the watchpoint condition for the variable was met as specified with the watchpoint operator (see below).</p> <p>The counter is reset when a program is started at Level 1.</p>
Persistent	<p>Marks a watchpoint as persistent. Persistent watchpoints are not restricted to the Natural object for which they are defined, but apply additionally to all subordinate program levels.</p> <p>Persistent watchpoints only make sense for variables that are passed to a subprogram "by reference" and not BY VALUE RESULT: see the relevant parameter description in CALLNAT, Parameters - operand2, in the Natural Statements documentation.</p> <p><b>Restriction:</b> Persistent watchpoints are not allowed for variables defined in a parameter or context clause.</p> <p>Valid input: Y (Yes) or N (No). Default: N.</p>
Act. Level	<p>Refers to Persistent above.</p> <p>Indicates the program level at which a persistent watchpoint was activated automatically.</p>
Commands	<p>Up to six debug commands. Enter one command per line. For a summary of all available commands, see Command Summary and Syntax.</p> <p><b>Attention:</b> If you delete the command BREAK and you do not enter any command that issues a dialog, there is no way for the Natural Debugger to receive control during program interruption.</p>

## Set Watchpoint

### To add a watchpoint to a session

- On the Watchpoint Maintenance menu, enter Function Code **S**, an object name and a variable name.  
Or enter the direct command SET WATCHPOINT *object variable*  
(see also the syntax of SET in Command Summary and Syntax).  
Or, **before** executing a Natural object:
  - Invoke the List Object Source screen (see the relevant section).
  - In the Source column, position the cursor at a variable name and choose PF18/Se Wp.

If object name and variable names are specified correctly, the watchpoint is set immediately and a corresponding confirmation message is displayed on the screen. The watchpoint receives the default command (BREAK), its initial and current state are set to "active" and no execution restrictions are specified. Note that if you delete the default command BREAK when setting a watchpoint and you do not enter any command that issues a dialog, there is no way for the Natural Debugger to receive control during program interruption.

If you specify not an object name but a valid variable name, the name of the default object (see the section Start the Natural Debugger) is assumed and the watchpoint is also set immediately. If there is no default object defined, a selection window appears that displays all objects available in the current library.

Below is information on:

- Watchpoint Operators

## Watchpoint Operators

### To specify a condition for the watchpoint to be activated

- On the Set Watchpoint or Modify Watchpoint screen, choose PF10 if you want to specify the value in alphanumeric format,  
Or, on the Set Watchpoint or Modify Watchpoint screen, choose PF11 if you want to specify the value in hexadecimal format.
- In the left input field, enter any of the watchpoint operators listed below.  
In the right input field, enter the value to be compared with the variable.

For watchpoints with operators specified for dynamic variables (alphanumeric or binary), the values will be compared from left to right. Since the field length of a dynamic variable varies, up to 253 bytes can be entered as comparative value. If the current length of the dynamic variable is shorter than the maximum comparative length of 253 bytes, the comparison is made only in the current length of the dynamic variable.

- Choose PF22 to return to the Set Watchpoint screen.

Valid watchpoint operators are:

Operator	Explanation
MOD	Modification. Activates the watchpoint each time a modification of the variable occurs.  This is the default setting.
EQ	Equal. Activates the watchpoint when the variable has been modified and when the current value of the variable is equal to the specified value.
NE	Not Equal. Activates the watchpoint when the variable has been modified and when the current value of the variable is not equal to the specified value.
GT	Greater Than. Activates the watchpoint when the variable has been modified and when the current value of the variable is greater than the specified value.
GE	Greater or Equal. Activates the watchpoint when the variable has been modified and when the current value of the variable is greater than or equal to the specified value.
LT	Less Than. Activates the watchpoint when the variable has been modified and when the current value of the variable is less than the specified value.
LE	Less or Equal. Activates the watchpoint when the variable has been modified and when the current value of the variable is less than or equal to the specified value.
INV	Invalid Contents. Activates the watchpoint each time the value assigned to a variable of the Type N, P, D or T does not comply with the following conditions:  N      Numeric unpacked.  P      Packed numeric.  D      Date range from 1582-01-01 to 2700-12-31.  T      Time range from 1582-01-01 00:00:00.0 to 2700-12-31 23:59:59.9.

# Call Statistics

The Call Statistics function can be used to obtain statistical information on which Natural objects were invoked during the execution of an application, and information on how often an object was invoked. Call statistics are deleted after resetting the debug environment.

## To invoke the Call Statistics Maintenance function

- On the Debug Main Menu enter Function Code **C**.  
Or enter the direct command **CS**.  
The Call Statistics Maintenance screen is displayed which provides the following functions:
  - Set Test Mode ON/OFF  
(see the relevant section)
  - Set Call Statistics ON/OFF
  - Display All Objects
  - Display Called Objects
  - Display Non-Called Objects
  - Print All Objects
  - Print Called Objects
  - Print Non-Called Objects

The print functions above are described under Print Objects below.

---

## Set Call Statistics ON/OFF

When executing a Natural object with Call Statistics ON, all calls made to a specific object are counted and the resulting statistics can afterwards be displayed or printed.

### To activate or deactivate Call Statistics

- On the Call Statistics Maintenance menu enter Function Code **C** to activate and deactivate.  
Or enter the direct command **SET CALL ON** or **SET CALL OFF**.

## Display All Objects

The Display Call Statistics screen provides an overview of the call frequency of the objects in your current library.

### To display all objects of a library

- On the Display Call Statistics Maintenance screen, enter Function Code **1** and a library name.  
Or enter the direct command **DISPLAY OBJECT *library***  
(see also the syntax of **DISPLAY** in Command Summary and Syntax).  
If you do not specify a library name, the library where you are currently logged on is assumed by default.

The Display Call Statistics screen lists all objects in your current library and indicates their call frequency in the Calls column on the right-hand side. For each call statement, such as **FETCH** or **CALLNAT**, an entry with the name of the object and a counter variable is written into the debug buffer. The counter is then increased for each call of the corresponding object.

For an example screen, see Display Called Objects below.

## Display Called Objects

The Display Called Objects screen corresponds to the Display Call Statistics screen, but only the objects that have been invoked are displayed.

### To display called objects of a library

- On the Call Statistics Maintenance menu, enter Function Code **2** and a library name.  
Or enter the direct command `DISPLAY CALL library`  
(see also the syntax of `DISPLAY` in Command Summary and Syntax).  
The Display Called Objects screen appears:

16:06:53		***** NATURAL TEST UTILITIES *****						2002-02-15		
Test mode ON		- Display Called Objects -						Object		
Object	Library	Type	DBID	FNR	S/C	Ver	Cat	Date	Time	All Calls
*_____	SAG_____									
MAINPGM	SAG	Program	10	32	S/C	3.1		2002-02-15	11:51	1
SUBPGM	SAG	Subprogram	10	32	S/C	3.1		2002-02-15	11:50	3
EMP-PGM	SAG	Program	10	32	S/C	3.1		2002-01-22	11:49	2
EMPLIND	SAG	Program	10	32	S/C	3.1		2001-08-13	11:18	1

If you do not specify a library name, the library where you are currently logged on is assumed by default.

## Display Non-Called Objects

The Display Non-Called Objects screen corresponds to the Display Call Statistics screen, but only the objects that have **not** been invoked are displayed.

### To display non-called objects

- On the Call Statistics Maintenance menu, enter Function Code **3** and a library name.  
Or enter the direct command `DISPLAY NOCALL library`  
(see also the syntax of `DISPLAY` in Command Summary and Syntax).

If you do not specify a library name, the library where you are currently logged on is assumed by default.

For an example screen, see Display Called Objects above.

## Print Objects

With the print functions, you can directly route a generated list of call statistics to a printer.

If you do not specify a library name, the library where you are currently logged on is assumed by default.

As listed under Print Options below, to invoke one of the print functions, you can either enter a function code on the Call Statistics Maintenance menu, or enter a direct command:

## Print Options

Print Function	Function Code	Direct Command
All Objects	4	PRINT OBJECT <i>library</i> .
Called Objects	5	PRINT CALL <i>library</i> .
Non-Called Objects	6	PRINT NOCALL <i>library</i> .

See also the syntax of PRINT in Command Summary and Syntax.

# Statement Execution Statistics

The Statement Execution Statistics function is used to obtain statistical information on which statement lines of invoked Natural objects were executed. The function also provides information on how often an object was invoked and how often a statement line was executed.

Statement execution statistics can be used for

- detecting dead (never gets executed) programming code in an application.
- estimating the coverage of an application test (how many statement lines have not been executed at least once for testing).
- locating frequently executed code segments that could have an impact on the application's performance.

This section covers the following topics:

- Set Statement Execution Statistics ON/OFF/COUNT
  - Invoke Statement Execution Statistics
  - Delete Statement Execution Statistics
  - Display Statement Execution Statistics
  - Print Statements
- 

## Set Statement Execution Statistics ON/OFF/COUNT

With this function, you activate statistics about executed statement lines of Natural objects.

Below is information on:

- Setup Options
- Activate/deactivate Statistics

### Setup Options

When starting a Natural object with Statement Execution Statistics ON or COUNT, all statement lines executed within a specific object are listed in a statistical report.

With the option ON, the Debugging utility only retains whether a specific statement line was executed or not; with the COUNT option, it counts how often a statement line was executed. You can specify a library and an object name to restrict statement execution statistics to the desired Natural objects. The default is to collect statistics for all objects of the current library. Asterisk (\*) notation is possible.

If you switch Statement Execution Statistics from ON to COUNT or vice versa, existing statistics are not affected, that is, their status of ON or COUNT remains.

The statistical data collected is stored in the debug buffer. The amount of storage that is required to store statistical information for a Natural object is approximately

(number of source lines) / 8 + 100 bytes with Statement Execution Statistics ON and  
(number of source lines) \* 4 + 100 bytes with Statement Execution Statistics COUNT.

If you modify a Natural object by inserting or deleting lines and you do not renumber the object lines before you STOW it, the amount of storage required for the object's statistics may increase. To avoid this, set Auto Renumber to Y (Yes) in the editor profile or use the command CATALL with the function Renumber source code lines option enabled (this is the default).

You can use the direct command PROFILE to limit the size of the debug buffer. With Statement Execution

Statistics set to option COUNT, no statement execution statistics are collected for objects with more than 8000 statement lines.

Statement Execution Statistics are part of the debug environment; therefore, they are affected by the direct commands SAVE ENVIRONMENT and LOAD ENVIRONMENT (see also the section Debug Environment Maintenance).

## Activate/deactivate Statistics

Below are instructions on how to activate or deactivate the function Statement Execution Statistics Maintenance. You can specify a library and/or an object name to restrict statement execution statistics to the desired Natural objects. The default is to collect statistics for all objects of the current library. Asterisk (\*) notation is possible.

### ▶ To activate Statement Execution Statistics

- On the Statement Execution Statistics Maintenance screen, enter Function Code **S**, the name of a library and/or the name of an object. In the State field, change the value to ON.  
Or enter the direct command SET XSTATISTICS ON or COUNT *library (object)*  
(see also the syntax of SET in Command Summary and Syntax).

If you do not specify a library and/or an object, the statistics data about all objects in your current library are activated.

### ▶ To deactivate Statement Statement Execution Statistics

- On the Statement Execution Statistics Maintenance screen, enter Function Code **S**, the name of a library and/or the name of an object. In the State field, change the value to OFF.  
Or enter the direct command SET XSTATISTICS OFF *library (object)*  
(see also the syntax of SET in Command Summary and Syntax).

If you do not specify a libber and/or an object, the statistics data about all objects in your current library are deactivated.

## Invoke Statement Execution Statistics

### ▶ To invoke the Statement Execution Statistics Maintenance function

- On the Debug Main Menu, Function Code **X**.  
Or enter the direct command XS.  
The Statement Execution Statistics Maintenance screen is displayed which provides the following functions:
  - Set Test Mode ON/OFF  
(see the relevant section)
  - Set Statement Execution Statistics ON/OFF/COUNT
  - Delete Statement Execution Statistics
  - Display Statement Execution Statistics
  - Print Statement Execution Statistics
  - Print All Statements
  - Print Executed Statement
  - Print Non-Executed Statements

The print functions are described under Print Statements below.

## Delete Statement Execution Statistics

### ▶ To delete Statement Execution Statistics

- On the Statement Execution Statistics Maintenance menu, enter Function Code **C** and the name of a library and/or the name of an object.  
Or enter the direct command `DELETE XSTATISTICS library (object)`  
(see also the syntax of `DELETE` in Command Summary and Syntax).

If you do not specify a library and/or an object, the statistics data about all objects in your current library are deleted.

## Display Statement Execution Statistics

The List Statement Execution Statistics screen displays a list of the specified statement execution statistics.

### ▶ To invoke the List Statement Execution Statistics screen

- On the Statement Execution Statistics Maintenance menu, enter Function Code **D**.  
Or enter direct command `DISPLAY XSTATISTICS`.  
The List Statement Execution Statistics screen is displayed:

16:02:01		***** NATURAL TEST UTILITIES *****							2002-02-15	
Test Mode ON		- List Statement Execution Statistics -							Object	
Co	Object	Library	Type	DBID	FNR	Obj.Called	Exec	Exec	%	Total No.
	*	*				n Times	able	uted		Executions
___	PGM01	SAG	Program	10	32	4	20	17	85	95
___	MAP01	SAG	Map	10	32	6	2	2	100	12
___	SPGM02	SAG	Subprogram	10	32	2	6	2	33	4
___	SAGTEST1	SAG	Program	10	32	2	20	10	50	17
___	DEBPGM	SAG	Program	10	32	1	6	6	100	34

For each object, the following information is displayed:

- the call frequency;
- the number of executable statement lines (a statement line is executable if a breakpoint could be set on it; see the description of the command `SET BREAKPOINT` in the section Breakpoint Maintenance for more information);
- the number of executed statement lines;
- the percentage of executed statement lines as related to the total number of executable statement lines;
- the total number of executed statement lines.

A list entry is highlighted if data is missing or possibly inconsistent.

- On the statistics list, you can mark statement execution statistics with a line command for further processing:

Command	Explanation
DE	Deletes statement execution statistics as described above.
DS	Displays all statement lines.
DX	Displays executed statement lines only.
DN	Displays non-executed statement lines only.
I	Displays information on the cataloged object and errors.
PS	Prints all statement lines.
PX	Prints executed statement lines only.
PN	Prints non-executed statement lines only.

The print functions are also described under Print Statements below.

Below is information on:

- Display All Statement Lines
- Display Executed Statement Lines
- Display Non-Executed Statement Lines

## Display All Statement Lines

The Display Statement Lines screen shows the object source and indicates whether or not a statement line has been executed.

### To invoke the Display Statement Lines screen

- On the List Statement Execution Statistics screen, mark an entry with the line command **DS**. Or enter the direct command `DISPLAY STATEMENT library (object)` (see also the syntax of `DISPLAY` in Command Summary and Syntax).

The Display Statement Lines screen appears. If Statement Execution Statistics has been set to `COUNT`, the execution frequency of the statement line is displayed as shown in the example screen below:

```

16:04:01          *****NATURAL TEST UTILITIES *****          2002-02-15
Test Mode ON          - Display Statement Lines -          Object SAGTEST

Line Source                                     Count
0200  RD1. READ EMPLOYEES-VIEW BY NAME                2
0210          STARTING FROM #NAME-START THRU #NAME-END
0220 *
0230  IF LEAVE-DUE>= 20                                1
0240          PERFORM MARK-SPECIAL-EMPLOYEES          not executed
0250  ELSE                                             not executed
0260          RESET #MARK                              1
0270  END-IF
0280 *
0290  RESET #MAKE #MODEL                              1
0300  CALLNAT 'SPGM02' PERSONNEL-ID #MAKE #MODEL      1
0310 *
0320  WRITE TITLE / '*** PERSONS WITH 20 OR MORE DAYS LEAVE DU
0330  / '*** ARE MARKED WITH AN ASTERISK          ***' //
0340  DISPLAY '//N A M E' NAME                        2

```

If no unique object has been specified, the List Statement Execution Statistics screen is displayed.

## Display Executed Statement Lines

The Display Executed Statement Lines screen corresponds to the Display Statement Lines screen, but only the statement lines that have been executed are displayed.

### To invoke the Display Executed Statement Lines screen

- On the List Statement Execution Statistics screen, mark an entry with the line command **DX**.  
Or enter the direct command `DISPLAY EXEC library (object)`  
(see also the syntax of DISPLAY in Command Summary and Syntax).

If no unique object has been specified, the List Statement Execution Statistics screen is displayed.

## Display Non-Executed Statement Lines

The Non-Executed Statement Lines screen corresponds to the Display Statement Lines screen, but only the statement lines that have not been executed are displayed.

### To invoke the Display Non-Executed Statement Lines screen

- On the List Statement Execution Statistics screen, mark an entry with the line command **DN**.  
Or enter the direct command `DISPLAY NOEXEC library (object)`  
(see also the syntax of DISPLAY in Command Summary and Syntax).

If no unique object has been specified, the List Statement Execution Statistics screen is displayed.

## Print Statements

With the print functions, you can directly route a generated list of statement execution statistics to a printer.

If you do not specify a library name, the library where you are currently logged on is assumed by default.

As listed under Print Options below, to invoke one of the print functions, you can either enter a function code on the Statement Execution Statistics Maintenance menu, enter a line command on the Display Statement Lines screen, or enter a direct command:

### Print Options

Print Function	Function Code	Line Command	Direct Command
Statement Execution Statistics	1		<code>PRINT XSTATISTICS <i>library (object)</i></code> .
All Statements	2	PS	<code>PRINT STATEMENT <i>library (object)</i></code> .
Print Executed Statements	3	PX	<code>PRINT EXEC <i>library (object)</i></code> .
Print Non-Executed Statements	4	PN	<code>PRINT NOEXEC <i>library (object)</i></code> .

See also the syntax of PRINT in Command Summary and Syntax.

# Variable Maintenance

With the Variable Maintenance function, you can display and modify variables within the Natural Debugger when a Natural object has been interrupted.

For the interrupted Natural object, the variable function displays user-defined variables, global variables and the database-related system variables \*COUNTER, \*ISN and \*NUMBER, together with their formats, lengths and contents.

Below is information on:

- Display User-defined, Global and DB-related System Variables
- Display System Variables
- Modify Variables  
(not applicable to system variables)

## Display User-defined, Global and DB-related System Variables

Below is information on invoking either the Display Variables (summary) screen with a list of all variables, or the Display Variable (individual) screen with all details on a particular variable:

- Display Variables - Summary
- Display Variable - Individual

### Display Variables - Summary

 **To display a summary of user-defined, global and database-related system variables**

- On the Debug Main Menu or in the Debug Window, enter Function Code **V**.

Or enter the direct command `DISPLAY VARIABLE variable,variable,...`  
(see also the syntax of `DISPLAY` in Command Summary and Syntax).

The Display Variables (summary) screen provides a list of the variables specified for the interrupted Natural object. Long values may be displayed truncated on the screen. For arrays, only the contents of the first occurrence are displayed.

To switch between alphanumeric and hexadecimal representation of the variable contents, choose PF10 (Alpha) and PF11 (Hex).

To toggle between the truncated display of a variable and the Full Name display with the group name, variable name and indices (if relevant), choose PF5/Zoom.

### Display Variable - Individual

 **To display an individual variable in its entirety**

- From the Display Variables (summary) screen, select a variable by marking it with the line command **DI**.
- Or enter the direct command `DISPLAY VARIABLE variable`  
(see also the syntax of `DISPLAY` in Command Summary and Syntax).
- Or, on the List Object Source screen, in the Source column, position the cursor at a variable name and choose PF18/Di Va.

- The following restrictions apply when using PF 18/Di Va:

If a variable name (including the occurrences of an array) spans more than one line, only the contents of the first line are evaluated.

If the index of an array is constant, for example, array (3,2,6), only this occurrence is displayed.

If the index of an array is variable, for example, array (i,j) or array (3:i), the entire array is displayed.

The Display Variable (individual) for the particular variable appears with all relevant specifications.

For large variables that exceed the length of 256 bytes, the screen displays the first 256 bytes by default.

To display the entire contents of the variable or navigate within the contents:

- Choose PF22 to page backward or PF23 to page forward.
- Or, in the field Position, enter a numeric value to start the display at a particular position.

To switch between alphanumeric and hexadecimal representation of the variable contents, choose PF10 (Alpha) and PF11 (Hex).

#### To display all occurrences of an array

- From the Display Variables screen, select a variable by marking it with the line command **DI**.
- Choose PF7 (-) and PF8 (+) to page between the individual occurrences or  
Or, enter the direct command `DIS VAR variable(*)` to display all occurrences.

## Display System Variables

#### To display system variables (except database-related system variables)

- Enter the direct command `SYSVARS`.  
The System Variables screen appears with a list of all system variables.

For variables of the type Handle, the name of the class of the instance that the Handle refers to is displayed in alphanumeric representation. If the class name is not available, the Globally Unique ID (GUID) is displayed instead. If the class was defined within Natural, the class name or GUID is suffixed with "(NAT)".

The contents of properties of an instance of a class cannot be displayed within the Natural Debugger.

## Modify Variables

**Not applicable to system variables.**

With the Modify Variable function, you can change the value of user-defined and global variables and the database-related system variables.

#### To modify the contents of a variable from the Modify Variable screen

- Invoke the Modify Variable screen by marking the variable with the line command **MO**.  
Or, on the Display Variable screen, choose PF5.
- On the Modify Variable screen, in the field Contents, change the value of the variable.  
The new contents must be valid for the format of the modified variable since the format of a variable cannot be modified within the Natural Debugger.  
On the Modify Variable screen, you can toggle between alphanumeric and hexadecimal representation of the variable value using PF10 (Alpha) and PF11 (Hex).

 **To modify the contents of a variable via direct command**

- Enter the direct command `MODIFY VARIABLE variable = new value` (see also the syntax of MODIFY in Command Summary and Syntax).

A message appears that confirms modification of the variable value.

# List Object Source

With the List Object Source function, you can display the source code of an object and maintain breakpoints. For you to be able to use List Object Source, the corresponding source must be in your current library or in one of its steplibs.

## ▶ To list the source code of an object

- On the Debug Main Menu, enter Function Code L and an object name.  
Or enter the direct command LIST *object*  
(see also the syntax of LIST in Command Summary and Syntax).  
The List Object Source screen appears and the object source is displayed with all current breakpoints listed in the Message column on the right-hand side of the screen.  
Choose PF8 or PF9 to scroll up or down one page.

If you execute a Natural object, the Natural Debugger interrupts execution at each breakpoint or watchpoint you have set and the Debug Window appears (see the relevant section in Concepts of the Natural Debugger).

## ▶ To list the source code of an interrupted Natural object

- From the Debug Window, choose Function Code L for List Break.  
Or, if relevant on other debug screens, choose PF9 (Li Br) or enter the direct command LIST BREAK.  
  
The List Object Source screen appears with the source code of the object displayed at the position where a break (breakpoint or watchpoint) occurred. The name of the breakpoint or watchpoint is displayed in the Message column on the right-hand side of the screen. The corresponding source code line is highlighted.

Below is information on:

- Maintain Breakpoints
- 

## Maintain Breakpoints

The List Object Source function, may be used to invoke or directly execute breakpoint maintenance functions from within an object source. For instructions on how to set breakpoints and general information on breakpoints, see Conditions of Use in Breakpoint Maintenance.

## ▶ To invoke a breakpoint maintenance function from an object source

- On the Debug Main Menu, enter Function Code L and an object name.  
Or enter the direct command LIST *object*  
(see also the syntax of LIST in Command Summary and Syntax).  
The source code of the specified object is displayed.  
The names of breakpoints already set are displayed in the Message column on the right-hand side of the screen.
  - To scroll the listing:
    - In the command line, enter a plus (+) or a minus (-) sign  
or enter the direct commands TOP, BOTTOM, LEFT and RIGHT.
- In the object source, mark the line(s) desired with any of the commands listed below:

AC	Activates breakpoints.
DA	Deactivates breakpoints.
DE	Deletes breakpoints.
DI	Displays breakpoints.
MO	Goes to the Modify Breakpoint maintenance screen.
SE	Sets breakpoints.
SM	Goes to the Set Breakpoint maintenance screen.

- Upon successful command execution, a corresponding message is displayed in the Message column on the right-hand side of the screen.

# Execution Control Commands

Listed below are commands the Natural Debugger provides for controlling the program flow during a debugging session. For a summary of all commands available with the Natural Debugger, refer to Command Summary and Syntax.

The commands listed below only apply when the Natural Debugger interrupts program execution.

- ESCAPE BOTTOM
  - ESCAPE ROUTINE
  - EXIT
  - GO
  - NEXT
  - RUN
  - STEP
  - STEP SKIPSUBLEVEL
  - STOP
- 

## ESCAPE BOTTOM

This command can only be used when a Natural object has been interrupted within a processing loop.

When you enter this command, the interrupted Natural object will be continued with the first statement following the processing loop.

## ESCAPE ROUTINE

When you enter this command, processing of the interrupted Natural object will be stopped and processing will continue with the object from which the interrupted Natural object was invoked; it will continue with the statement following the corresponding CALLNAT, PERFORM or FETCH RETURN statement.

If you apply the command ESCAPE ROUTINE to a main program, Natural ends the program and returns to the command mode.

## EXIT

If you are displaying the Debug Main Menu and invoke the EXIT function, choose PF3 (Exit) or enter the execution control command EXIT, the debugging utility returns either to the calling program (that is, to the interrupted Natural object which is then continued) or to the NEXT line if the debugging utility has been invoked with the direct command TEST, or to the corresponding input field if it has been invoked by the terminal command %<TEST. However, if a breakpoint or watchpoint is currently active, the next command of this breakpoint or watchpoint is executed.

If you are not in the Debug Main Menu and enter the direct command EXIT or choose PF3 (Exit), you leave the current function and return to the previous step of your debugging session.

## GO

When you enter the direct command GO (or choose PF14), the debugging utility returns control to the execution of the interrupted Natural object. If a breakpoint or watchpoint was active at the time the Natural object was interrupted, the remaining commands of this break or watchpoint are **not** executed.

## NEXT

When you enter the direct command NEXT (or choose PF13), the next command specified for a breakpoint or watchpoint is executed. If no further command has been specified, program execution continues.

## RUN

When you enter the direct command RUN, test mode is switched off and program execution continues, without investigating any further breakpoints and watchpoint.

## STEP

When you enter the direct command STEP (or choose PF2), an interrupted Natural object is continued for *n* statement lines. The default value for *n* is 1.

## STEP SKIPSUBLEVEL

When you enter the direct command STEP SKIPSUBLEVEL (or choose PF17) upon a statement which invokes another object (for example, CALLNAT), processing is continued with the next statement line in the interrupted Natural object (instead of the first executed statement in the invoked object).

With the command, you can specify a level number *n*. *n* may be the level of the interrupted Natural object (this is the default) or a superior level. Step mode then continues with the next object at the specified level.

If this command is applied to a statement that does not invoke another object, the debugging utility reacts as if the command STEP had been entered.

Level information can be obtained with the command OBJCHAIN as described in the section Navigation and Information.

## STOP

When you enter the direct command STOP, both the debugging utility and any interrupted Natural object are terminated; the NEXT line is displayed.

# Navigation and Information Commands

Listed below are direct commands the Natural Debugger provides for navigating through the debugging areas, scrolling screen displays, obtaining various information on objects and variables, and specifying profiles. For a summary of all commands available with the Natural Debugger, refer to Command Summary and Syntax.

- BREAK
  - FLIP
  - LAST
  - OBJCHAIN
  - ON/OFF
  - PROFILE
  - SCAN
  - SCREEN
  - SET OBJECT
  - STACK
  - SYSVARS
  - TEST ON/OFF
- 

## BREAK

The command **BREAK** is the default command which is automatically set when creating a new debug entry. It displays the Debug Window as described in Concepts of the Natural Debugger.

When the command **BREAK** is deleted upon modification of the corresponding debug entry, no Debug Window appears. However, other specified commands are executed and the event count is increased.

## FLIP

The command **FLIP** switches between the display of the two PF-key lines (PF1 to PF12 and PF13 to PF24).

## LAST

The command **LAST** displays the command last entered. The last three commands are stored and can be recalled.

## OBJCHAIN

The command **OBJCHAIN** can only be used when a Natural object has been interrupted.

This command displays the objects on the current level and all superior levels, as well as the current GDA (Global Data Area), if applicable, and provides information on the interruption.

## ON/OFF

When you enter the command **ON/OFF** in the Natural debug utility, test mode is switched on or off respectively. See also **TEST ON/OFF** below.

## PROFILE

The command PROFILE displays the Edit Profile screen where you can modify the profile of the debugging utility.

### Edit Profile Screen

The Edit Profile screen provides you with the following options:

Option	Explanation
Reset Debug Environment Automatically on Exit	Specifies an automatic reset of your current debug environment once you exit the debugging utility. The default is N (No).
File for Loading/Saving Debug Environments	Specifies to/from which system file debug environments are to be saved/loaded: FUSER (default), FNAT or SPAD ( scratch-pad file).
Confirm EXIT/CANCEL Before Execution	Specifies a confirmation of an EXIT or CANCEL command before execution. The default is N (No).
Stack Unknown Commands	Specifies that any unknown debug command which is entered (for example, the name of a called program) is to be stacked. If so, once you enter an unknown debug command, you immediately exit the debugging utility and the command is executed. If this option has not been specified, an unknown debug command leads to a corresponding error message. The default is Y (Yes).
Output Device	Specifies an output device for the Call Statistics Maintenance function; the default value is HARDCOPY.
Maximum debug buffer size in KB	Specifies the maximum size (in kilobytes) of the debug buffer. The debug buffer is automatically enlarged as required, but only up to the specified maximum. Enter 0 to indicate no limit or enter a value from 4 - 16384 (must be a multiple of 4). If the limit would be exceeded, no further debug entries can be defined and no additional call or statement execution statistics entries are generated.

## SCAN

Only applies to the List Object Source (see the relevant section) function.

This command searches for a string of characters within an object source:

SCAN searches for the value specified which may be delimited by blanks or by any characters that are neither letters nor numeric characters.

SCAN ABS results in an absolute scan of the source code for the specified value regardless of what other characters may surround the value.

See also the syntax diagrams in Command Summary and Syntax.

## SCREEN

When you enter the command SCREEN upon interruption of a Natural object, the current screen output of the interrupted Natural object is displayed. ENTER takes you back to debug mode.

## SET OBJECT

The command SET OBJECT changes the name of the default object as described in the relevant section in Start the Natural Debugger. See also the syntax of SET in the section Command Summary and Syntax.

## STACK

When you enter the command STACK, the contents of the entry at the top of the Natural stack is displayed. Up to 15 individual top entry elements can be displayed. Elements longer than 55 characters are truncated and marked with an asterisk (\*).

**Note:**

An error message is displayed if any single element is longer than 249 characters.

## SYSVARS

When you enter this command, the current values of system variables are displayed.

## TEST ON/OFF

The command TEST ON/OFF switches test mode on or off respectively. In the Natural debug utility, you only need to enter ON/OFF as described above.

# Command Summary and Syntax

This section provides a list of all commands available with the Natural Debugger. An underlined portion of a keyword represents an acceptable abbreviation. For an explanation of more complex command structures with user-defined operands, see Syntax Diagrams below.

Below is information on:

- All Debug Commands
- Syntax Diagrams

## All Debug Commands

Command	Subcommand(s)	Explanation
-		Scrolls one page down in a list.
--		Scrolls to the beginning of a list.
<u>TOP</u>		
+		Scrolls one page down in a list.
++		Scrolls to the end of a list.
<u>BOTTOM</u>		
<u>ACTIVATE</u> (syntax below)	<u>BREAKPOINT</u> <u>BP</u>	Activates breakpoints as described in the relevant section in Breakpoint Maintenance.
	<u>SPY</u>	Activates breakpoints <b>and</b> watchpoints: see also Activate Spy in Spy Maintenance.
	<u>WATCHPOINT</u> <u>WP</u>	Activates watchpoints as described in the relevant section in Watchpoint Maintenance.
<u>BREAK</u>		Displays the Debug Window: see also <u>BREAK</u> in Navigation and Information Commands.
<u>CANCEL</u>		Cancels the current operation and/or exits screens without saving modifications.
<u>DBLOG</u>	<u>A</u> <u>Q</u> <u>D</u>	<p>Invokes the DBLOG utility (see the relevant documentation) from within the Natural Debugger.</p> <p>To specify a database environment, use any of the subcommands provided:</p> <p>A = Adabas (this is the default),</p> <p>Q = SQL,</p> <p>D = DL/I</p> <p><b>Note:</b> During a debug interrupt, you can only specify any of the subcommands listed above.</p>

Command	Subcommand(s)	Explanation
<u>DEACTIVATE</u> <u>DA</u>  (syntax below)	<u>BREAKPOINT</u> <u>BP</u>	Deactivates breakpoints as described in the relevant section in Breakpoint Maintenance.
	<u>SPY</u>	Deactivates breakpoints <b>and</b> watchpoints: see also Deactivate Spy in Spy Maintenance.
	<u>WATCHPOINT</u> <u>WP</u>	Deactivates watchpoints as described in the relevant section in Watchpoint Maintenance.
<u>DELETE</u>  (syntax below)	<u>BREAKPOINT</u> <u>BP</u>	Deletes breakpoints as described in the relevant section in Breakpoint Maintenance.
	<u>SPY</u>	Deletes breakpoints <b>and</b> watchpoints: see also Delete Spy in Spy Maintenance.
	<u>WATCHPOINT</u> <u>WP</u>	Deletes watchpoints as described in the relevant section in Watchpoint Maintenance.
	<u>ENVIRONMENT</u>	Deletes the specified debug environment: see also Delete Debug Environment in Debug Environment Maintenance.
<u>DISPLAY</u>  (syntax below)	<u>BREAKPOINT</u> <u>BP</u>	Displays breakpoints as described in the relevant section in Breakpoint Maintenance.
	<u>SPY</u>	Displays breakpoints <b>and</b> watchpoints: see also Display Spy in Spy Maintenance.
	<u>WATCHPOINT</u> <u>WP</u>	Displays watchpoints as described in the relevant section in Watchpoint Maintenance.
	<u>CALL</u>	Displays statistics on Natural objects invoked during the execution of an application: see also Display Called Objects in Call Statistics Information.
	<u>EXEC</u>	Displays statistics on executed statement lines of invoked Natural objects: see also Display Executed Statement Lines in Statement Execution Statistics.
	<u>HEXADECIMAL</u>	Displays the contents of variables in hexadecimal format.
	<u>NOCALL</u>	Displays statistics on Natural objects that have not been invoked during the execution of an application: see also Display Non-Called Objects in Call Statistics Information.
	<u>NOEXEC</u>	Displays statistics on non-executed statement lines of invoked Natural objects: see also Display Non-Executed Statement Lines in Statement Execution Statistics.
	<u>OBJECT</u>	Displays statistics on the call frequency of objects: see also Display All Objects in Call Statistics Information.
	<u>STATEMENT</u>	Display statistics on executed and non-executed statement lines of invoked Natural objects: see Display All Statement Lines in Statement Execution Statistics.
	<u>VARIABLE</u>	Displays variables for interrupted Natural objects as described in the relevant section in Variable Maintenance.
<u>XSTATISTICS</u>	Displays a statistical summary of execution statistics: see also Display Statement Execution Statistics in Statement Execution Statistics.	

Command	Subcommand(s)	Explanation
<u>ESCAPE</u>	<u>BOTTOM</u>	Stops processing a loop and escapes to the first statement after the loop: see ESCAPE BOTTOM in Execution Control Commands.
	<u>ROUTINE</u>	Stops processing an interrupted Natural object and continues with another object, if available: see ESCAPE ROUTINE in Execution Control Commands.
<u>EXIT</u>		Leaves the current screen: see EXIT in Execution Control Commands.
<u>FLIP</u>		Switches between the display of the two PF-key lines (PF1 to PF12 and PF13 to PF24).
<u>GO</u>		Returns control to the execution of the interrupted Natural object: see GO in Execution Control Commands.
<u>LAST</u>		Displays the command entered last. The last three commands are stored and can be recalled.
<u>LEFT</u>		Shifts to the left side of a source code listing.
<u>LIST</u> (syntax below)		Displays the source code of a object.
	<u>BREAK</u>	Shows the object source with the current break. The relevant statement line is highlighted.
	<u>LASTLINE</u>	Shows the object source with the last line executed before the current break.
<u>LOAD</u> (syntax below)	<u>ENVIRONMENT</u>	Loads the debug environment specified: see Load Debug Environment in Debug Environment Maintenance.
<u>MENU</u>		Invokes the Debug Main Menu.
<u>MODIFY</u> (syntax below)	<u>BREAKPOINT BP</u>	Modifies breakpoints as described in the relevant section in Breakpoint Maintenance.
	<u>SPY</u>	Invokes the Modify Breakpoint or Modify Watchpoint screen: see also Modify Spy in Spy Maintenance.
	<u>WATCHPOINT WP</u>	Modifies watchpoints as described in the relevant section in Watchpoint Maintenance.
	<u>HEXADECIMAL</u>	Modifies the contents of variables in hexadecimal format.
	<u>VARIABLE</u>	Invokes the Display Variable screen for modification as described in the relevant section in Variable Maintenance. See also Modify Variable.
<u>NEXT</u>		Executes the next command specified for a breakpoint or watchpoint.
<u>OBJCHAIN</u>		Displays executed objects at various program levels: see OBJCHAIN in Navigation and Information Commands.
<u>ON/OFF</u>		Switches test mode on/off.

Command	Subcommand(s)	Explanation
<u>P</u> RINT (syntax below)	<u>C</u> ALL	Prints statistics on Natural objects invoked during the execution of an application: see also Display Called Objects in Call Statistics Information.
	<u>E</u> XEC	Prints statistics on executed statement lines of invoked Natural objects: see also Display Executed Statement Lines in Statement Execution Statistics.
	<u>N</u> O <u>C</u> ALL	Prints statistics on Natural objects that have not been invoked during the execution of an application: see also Display Non-Called Objects in Call Statistics Information.
	<u>N</u> O <u>E</u> XEC	Prints statistics on non-executed statement lines of invoked Natural objects: see also Display Non-Executed Statement Lines in Statement Execution Statistics.
	<u>O</u> BJECT	Prints statistics on the call frequency of objects: see also Display All Objects in Call Statistics Information.
	<u>S</u> TATEMENT	Prints statistics on executed and non-executed statement lines of invoked Natural objects: see also Display All Statement Lines in Statement Execution Statistics.
	<u>X</u> STATISTICS	Prints statistics on executed statement lines: see also Display Statement Execution Statistics in Statement Execution Statistics.
<u>P</u> ROFILE		Displays the Edit Profile screen where you can modify the profile of the Natural Debugger. For details on the Edit Profile screen, see the relevant section in Navigation and Information Commands.
<u>R</u> ESET (syntax below)	<u>E</u> NVIRONMENT	Resets the current debug environment: see Reset Debug Environment in Debug Environment Maintenance.
<u>R</u> IGHT		Shifts to the right side of a source code listing.
<u>R</u> UN		Switches off test mode and continues program execution.
<u>S</u> AVE (syntax below)	<u>E</u> NVIRONMENT	Resets the current environment and saves the debug specifications. See also Save Debug Environment in Debug Environment Maintenance.
<u>S</u> CAN	<u>A</u> BS	Only applies to the List Object Source (see the relevant section) function.  Searches for a value in the source code of an object: see SCAN in Navigation and Information commands and Syntax Diagrams below.
<u>S</u> CREEN		When entered upon interruption of an object, the current screen output of the interrupted Natural object is displayed. ENTER takes you back to debug mode.

Command	Subcommand(s)	Explanation
<u>SET</u> (syntax below)	<u>BREAKPOINT</u> <u>BP</u>	Invokes the Set Breakpoint menu.
	<u>CALL ON/OFF</u>	Activates/deactivates the Call Statistics Maintenance as described in the relevant section.
	<u>OBJECT</u>	Changes the default object defined for the Natural debugging utility. See also <u>SET OBJECT</u> in Navigation and Information.
	<u>WATCHPOINT</u> <u>WP</u>	Invokes the Set Watchpoint menu.
	<u>XSTATISTICS</u> <u>ON/OFF/COUNT</u>	Activates the statistics function about executed statements of Natural objects. See also Set Statement Execution Statistics in Statement Execution Statistics.
<u>SM</u>		Invokes the Spy Maintenance menu as described in the relevant section.
<u>STACK</u>		Displays the contents of the entry at the top of the Natural stack: see <u>STACK</u> in Navigation and Information Commands.
<u>STEP</u>	[ <i>n</i> ]	Continues an interrupted Natural object for <i>n</i> statement lines. The default value for <i>n</i> is 1.
	<u>SKIPSUBLEVEL</u>	Continues step-mode processing of interrupted Natural objects without entering programs at sub-levels: see also <u>SKIPSUBLEVEL</u> in Execution Control Commands.
<u>STOP</u>		Terminates both the Natural Debugger and any interrupted Natural object; the <u>NEXT</u> line is displayed.
<u>SYSVARS</u>		Displays the current values of system variables (except the database-related system variables). See also Display Variables in Variable Maintenance.
<u>TEST ON/OFF</u>		Switches test mode on/off. Can also be entered at the <u>NEXT</u> command level.
<u>WM</u>		Invokes the Watchpoint Maintenance menu as described in the relevant section.

## Syntax Diagrams

The syntax diagrams listed below refer to more complex command sequences.

For a detailed explanation of the symbols used within the syntax descriptions, see the section System Command Syntax in the Natural System Command Reference documentation.

For better readability, synonymous keywords are omitted from the syntax diagrams below. An underlined portion of a keyword represents an acceptable abbreviation.

Valid synonyms are:

Keyword	Synonym
<u>BREAKPOINT</u>	BP
<u>DEACTIVATE</u>	DA
<u>WATCHPOINT</u>	WP

Below is information on:

- ACTIVATE
- DEACTIVATE
- DELETE
- DISPLAY
- LIST
- LOAD
- MODIFY
- PRINT
- RESET
- SAVE
- SET

## ACTIVATE

<b>ACTIVATE</b>	{	<b>SPY</b>	[ { <i>name</i> } ]	}
		<b>BREAKPOINT</b>	[ <i>object</i> ] [ <i>line</i> ]	
		<b>WATCHPOINT</b>	[ [ <i>object</i> ] <i>variable</i> ]	

## DEACTIVATE

<b>DEACTIVATE</b>	{	<b>SPY</b>	[ { <i>name</i> } ]	}
		<b>BREAKPOINT</b>	[ <i>object</i> ] [ <i>line</i> ]	
		<b>WATCHPOINT</b>	[ [ <i>object</i> ] <i>variable</i> ]	

## DELETE

<b><u>DELETE</u></b>	<b><u>SPY</u></b> [ { <i>name</i> <i>number</i> } ]
	<b><u>BREAKPOINT</u></b> [ <i>object</i> ] [ <i>line</i> ]
	<b><u>WATCHPOINT</u></b> [ [ <i>object</i> ] <i>variable</i> ]
	<b><u>XSTATISTICS</u></b> [ [ <i>library</i> ] <i>object</i> ]
	<b><u>ENVIRONMENT</u></b> [ <i>name</i> ]

## DISPLAY

<b><u>DISPLAY</u></b>	<b><u>SPY</u></b> [ { <i>name</i> <i>number</i> } ]
	<b><u>BREAKPOINT</u></b> [ <i>object</i> ] [ <i>line</i> ]
	<b><u>WATCHPOINT</u></b> [ [ <i>object</i> ] <i>variable</i> ]
	<b><u>CALL</u></b>
	<b><u>OBJECT</u></b>
	<b><u>NOCALL</u></b>
	<b><u>XSTATISTICS</u></b> <i>library</i> [ <i>object</i> ]
	<b><u>STATEMENT</u></b>
	<b><u>EXEC</u></b>
	<b><u>NOEXEC</u></b>
	<b><u>VARIABLE</u></b> [ <i>variable,...</i> ]
	<b><u>HEXADECIMAL</u></b>

## LIST

<b><u>LIST</u></b>	{ <b><u>LASTLINE</u></b> <b><u>BREAK</u></b> <i>object</i> [ <i>line</i> ] }
--------------------	--

**LOAD**

**LOAD ENVIRONMENT** [ *name* ]

**MODIFY**

<b><u>MODIFY</u></b>	{	<b><u>SPY</u></b> [ { <i>name</i> <i>number</i> } ]	}
		<b><u>BREAKPOINT</u></b> [ <i>object</i> ] [ <i>line</i> ]	
		<b><u>WATCHPOINT</u></b> [ [ <i>object</i> ] <i>variable</i> ]	
		<b><u>VARIABLE</u></b> [ <i>variable</i> [ = <i>new value</i> ] ]	
		<b><u>HEXADECIMAL</u></b>	

**PRINT**

<b><u>PRINT</u></b>	{	<b><u>CALL</u></b>	}
		<b><u>OBJECT</u></b>	
		<b><u>NOCALL</u></b>	
		<b><u>XSTATISTICS</u></b> [ <i>library</i> ] [ <i>object</i> ]	
		<b><u>STATEMENT</u></b>	
		<b><u>EXEC</u></b>	
		<b><u>NOEXEC</u></b>	

**RESET**

**RESET ENVIRONMENT** [ *name* ]

**SAVE**

**SAVE ENVIRONMENT** [ *name* ]

SET

SET	<b>OBJECT</b>	<i>object</i>		
	<b>BREAKPOINT</b>	<i>object</i>	{	<i>line</i> <i>label</i>
	<b>WATCHPOINT</b>	[	[ <i>object</i> ]	<i>variable</i> ]
	<b>CALL</b>	{	<b>OFF</b> <b>ON</b>	}
	<b>XSTATISTICS</b>	{	<b>OFF</b> <b>ON</b> <b>COUNT</b>	[ <i>library</i> [ <i>object</i> ] ] }

# INPL Utility - Overview

The INPL utility documentation covers the following topics:

- INPL - Introduction
- Load Library Only
- Load DDMs Only
- Load Error Messages Only
- Load All Objects  
(Load Library, DDMs and Error Messages)
- Scan INPL File
- Natural Security Recover

# INPL Utility - Introduction

The INPL (Initial Natural Program Load) utility is used to load Natural modules, DDMs or error messages from Software AG datasets (installation or update datasets) from Work File 1 into Natural system files. INPL is only used for initial Natural program loading or for implementing program corrections supplied by Software AG.

In addition, the INPL utility provides a scan function to check the contents of the datasets assigned to Work File 1 and a Natural Security Recover function which forces an initialization of the Natural Security environment.

When executed, the INPL utility will delete buffer pool entries of any cataloged object loaded to the Natural system file from the buffer pool used by the Natural session of the INPL utility.

If an error occurs during INPL execution, the INPL will be interrupted and terminate abnormally with Condition Code 40.

- Restrictions
  - Special Cases
  - Invoking INPL
  - Options Available
  - INPL Report
- 

## Restrictions

Only datasets which are marked as "SAG system INPL dataset" are processed. To process datasets created with the NATUNLD utility, use the NATLOAD utility.

## Special Cases

In any of the following cases, the INPL command must be specified using the dynamic Natural profile parameter STACK:

- when an INPL is to be performed on an empty FNAT system file;
- when a new system maintenance upgrade INPL is to be performed;
- when an INPL is to be performed in a Natural Security environment.

## Invoking INPL

### To invoke the INPL utility

1. In the direct command line, enter the system command INPL and confirm your entry.  
The Natural INPL Utility menu is displayed:

```

17:05:09          ***** NATURAL INPL UTILITY *****          2002-02-08
User: SAG                                     Library: SYSTEM

          Code   Function

          L     Load Library Only
          D     Load DDMs Only
          E     Load Error Messages Only
          B     Load Library, DDMs and Error Messages
          S     Scan INPL File
          R     NATURAL SECURITY Recover
          ?     Help
          .     Exit

Code ..... B
Replace .... Y (Y/N/O)   Load Except . N (Y/N)
DDM Name ....
Library .....
Object Name .           Date .....          (YYYY-MM-DD)
Check Date .. N (Y/N)   Number ..... 0

```

2. From the INPL menu, choose one of the following functions by entering the corresponding function code in the Code field:

- Load Library Only
- Load DDMs Only
- Load Error Messages Only
- Load Library, DDMs and Error Messages  
(see the section All Objects)
- Scan INPL File
- Natural Security Recover

For details on these functions, use the hyperlinks, or refer to the corresponding sections. Modify or complete the remaining input fields as described in Options Available below.

3. Choose ENTER to confirm your entries.

## Options Available

When you enter a function, you can specify the following parameters:

Parameter	Meaning
Library	This parameter only applies to the functions Load Library Only, Load DDMs Only, Load Error Messages Only, Load All Objects and Scan INPL File. It indicates the name of the library to be loaded/scanned. If the value ends with an asterisk (*), each library whose name begins with the specified value will be loaded/scanned. The library name is mandatory if Object Name is specified.
Object Name	This parameter only applies to the functions Load Library Only, Load All Objects and Scan INPL File. It indicates the name of the programming object(s) to be loaded/scanned. If the value ends with an asterisk (*), each object whose name begins with the specified value will be loaded/scanned. If the field is left blank, all objects within the library specified under Library will be loaded/scanned.

Parameter	Meaning
DDM Name	This parameter only applies to the functions Load DDMs Only, Load All Objects and Scan INPL File. It indicates the name of the DDM(s) to be loaded/scanned. If the value entered ends with an asterisk (*), each DDM whose name begins with the specified value will be loaded. If only an asterisk is entered or this field is left blank, all DDMs will be loaded/scanned.
Date (start)	This parameter only applies to functions Load Library Only, Load DDMs Only, Load All Objects and Scan INPL File. It enables you to restrict loading/scanning to those programming objects which were saved/cataloged on or after a given date. The date is entered in the format YYYY-MM-DD.
Number	This parameter enables you to restrict loading/scanning to a certain number of programming objects and/or DDMs. All objects which meet the selection criteria are counted. If the number of objects processed has reached the value of Number, processing is terminated with a corresponding message.
Replace	This parameter does not apply to the scan functions.  Specifies whether objects/DDMs/error messages to be loaded are to replace any that already exist on the system file. With Option Y, the existing objects/DDMs/error messages are replaced. With Option N, the existing objects/DDMs/error messages are not replaced. Default for Replace is Y. See also the Check Date parameter if only older objects are to be replaced.  In the function Natural Security Recover, you can enter the Option O in this field to reset the owner information of selected objects.
Check Date	This parameter only applies to the functions Load Library Only, Load DDMs Only and Load All Objects. It indicates whether programming objects or DDMs are to be replaced depending on their time stamp. If set to Y, only older objects are replaced (that is, objects which were saved or cataloged before the objects of the same names which are to be loaded). If set to N, all objects are replaced. This parameter has no effect if Replace is set to N. The default for Check Date is N.
Load Except	If Y is entered, the specified DDM name, Library and Object name selection are not loaded. This means, that all objects <b>except</b> the specified objects are to be loaded/scanned. Default for Load Except is N.  <b>Examples with Load Except = Y:</b> Code = L Library = ABC: All libraries except the library ABC are loaded. Code = D DDM Name = XY*: All DDMs with a prefix other than XY are loaded. Code = B Library = AB* DDM Name = CD*: All libraries with a prefix other than AB and all DDMs with a prefix other than CD are loaded.

## INPL Report

When the selected INPL function is complete, a corresponding INPL report is displayed on a screen.

# INPL - Load Library Only

This function of the INPL utility is used to load Natural object modules and source programs into specified libraries in the Natural system file FNAT or FUSER.

## To load libraries

1. On the INPL menu, enter Function Code **L**.
2. You can specify parameters to be valid during execution of this function:
  - Replace (Y/N)
  - Load Except (Y/N)
  - Library
  - Object Name
  - Date (start)(YYYY-MM-DD)
  - Check Date (Y/N)
  - Number

For detailed information on these parameters, use the hyperlinks, or refer to Options Available in the section Introduction.

3. Confirm your entry.

When the function is complete, a corresponding INPL Report (see the section Introduction) is output.

# INPL - Load DDMs Only

This function of the INPL utility is used to load DDMs into the Predict system file FDIC.

## To load DDMs

1. On the INPL menu, enter Function Code **D**.
2. You can specify parameters to be valid during execution of this function:
  - Replace (Y/N)
  - Load Except (Y/N)
  - DDM Name
  - Number

For detailed information on these parameters, use the hyperlinks, or refer to Options Available in the section Introduction.

3. Confirm your entry.

When the function is complete, a corresponding INPL Report (see the section Introduction) is output.

# INPL - Load Error Messages Only

This function of the INPL utility is used to load error messages into specified libraries in the Natural system file FNAT or FUSER.

## To load error messages

1. On the INPL menu, enter Function Code **E**.
2. You can specify parameters to be valid during execution of this function:
  - Replace (Y/N)
  - Library
  - Object Name

For detailed information on these parameters, use the hyperlinks, or refer to Options Available in the section Introduction.

3. Confirm your entry.

When the function is complete, a corresponding INPL Report (see the section Introduction) is output.

# INPL - Load All Objects

(Load Library, DDMs and Error Messages)

This function of the INPL utility is used to load Natural object modules, source programs and error messages into specified libraries in the Natural system files FNAT and FUSER. DDMs are loaded into the Predict system file FDIC.

## To load libraries, DDMs and error messages

1. On the INPL menu, enter Function Code **B**.
2. You can specify parameters to be valid during execution of this function:
  - Replace (Y/N)
  - Load Except (Y/N)
  - DDM Name
  - Library
  - Object Name
  - Date (YYYY-MM-DD)
  - Check Date (Y/N)
  - Number

For detailed information on these parameters, use the hyperlinks, or refer to Options Available in the section Introduction.

3. Confirm your entry.

When the function is complete, a corresponding INPL Report (see the section Introduction) is output.

# INPL - Scan INPL File

This function of the INPL utility is used to scan the contents of the dataset assigned to Work File 1.

## To scan an INPL File

1. On the INPL menu, enter Function Code **S**.
2. You can specify parameters to be valid during execution of this function:
  - Scan/Load Except (Y/N)
  - DDM Name
  - Library
  - Object Name
  - Date (YYYY-MM-DD)
  - Number

For detailed information on these parameters, use the hyperlinks, or refer to Options Available in the section Introduction.

3. Confirm your entry.

When the function is complete, a corresponding INPL Report (see the section Introduction) is output.

# INPL - Natural Security Recover

This function of the INPL utility is used to force an initialization of the Natural Security environment. You can

- Reset the Environment or
  - Remove Owners
- 

## Reset Environment



Execution of this function will reset the user profile DBA and the library profile SYSSEC as well as the link between these two objects as they were after the initial installation; all other links to the library SYSSEC will be canceled. Other Natural Security profiles and links will not be modified.

**Contact Software AG Support for further information.**



### To reset the environment

- On the INPL menu, enter Function Code **R**.

## Remove Owners



### To remove owners

- On the INPL menu, enter Function Code **R** and Code **O** in the Replace field to reset the owner information of the objects selected.

# NATPAGE Utility - Screen Capturing

The utility NATPAGE is used to capture screen output data (maps and reports) during a Natural session. The term screen in this context means the contents of the page buffer; that is, the logical page output by Natural.

The screen captures are stored in the Natural scratch-pad file as described in the relevant section in the Natural Operations for Mainframes documentation.

The maximum number of screens that can be captured is determined by the session parameter PD, which is described in the Natural Parameter Reference documentation.

The NATPAGE utility consists of the following Natural terminal commands:

Command	Function
%P	Activates NATPAGE and captures the contents of the current screen and all subsequent screens. Screens captured previously are deleted.
%I	Activates NATPAGE and captures the contents of the current screen.
%O	Deactivates NATPAGE.
%S	Resumes NATPAGE.
%E	Displays the screens captured with NATPAGE.

See the Natural Terminal Commands documentation for a detailed description of these terminal commands.

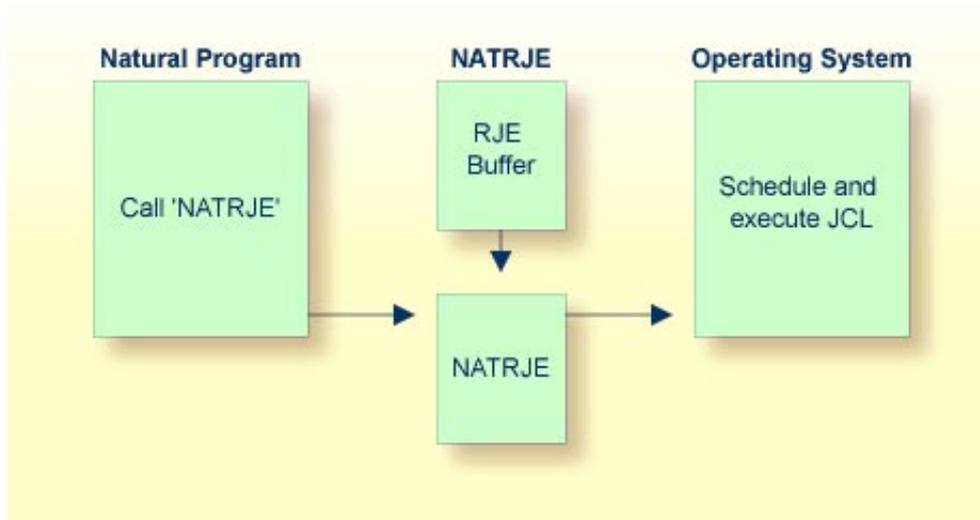
# NATRJE - Natural Remote Job Entry

The NATRJE utility (Natural Remote Job Entry) can be used to submit JCL cards from a Natural program to the operating system for scheduling and execution. For example, it is possible to start a Natural batch job with NATRJE.

The NATRJE documentation covers the following topics:

- NATRJE General Information
- Calling NATRJE from a Natural Program
- NATRJE Return Codes
- NATRJE User Exit
- NATRJE Features Applicable to UTM/TIAM

## NATRJE General Information



1. The Natural program calls the NATRJE utility for the purpose of submitting JCL cards to be executed by the operating system.
2. NATRJE collects the JCL cards into the RJE buffer until the Natural program indicates that the job is complete. The RJE buffer holds the JCL cards before they are submitted. The initial size of the RJE buffer is determined by the RJESIZE profile parameter (as described in the Natural Parameter Reference documentation). If a given job does not fit into the RJE buffer, the buffer is automatically enlarged. The maximum size of a job is determined by the thread or region size.
3. NATRJE transfers the JCL cards to the operating system internal job queue for scheduling and execution by the operating system.

### Note for BS2000/OSD:

In BS2000/OSD environments, when generation of the job is completed, NATRJE transfers the JCL cards to a BS2000/OSD dataset, which is generated by NATRJE. The dataset is a SAM file and is submitted via ENTER to the BS2000/OSD operating system.

## Calling NATRJE from a Natural Program

Below is information on:

- Invoking NATRJE
- Example Programs

### Invoking NATRJE

 **To invoke the NATRJE utility**

- Use a CALL statement in the Natural program.

The CALL statement has the following syntax:

```
CALL 'NATRJE' parm1 parm2 parm3 parm4
```

The parameters specified in the CALL statement are:

Parameter	Explanation
<i>parm1</i>	The starting JCL card of the table which contains one or more 80-character JCL cards to be submitted.
<i>parm2</i>	A 4-byte binary field which contains the number of 80-character JCL cards to be submitted.
<i>parm3</i>	<p>A 1-byte alphanumeric field used to indicate if all JCL cards have been submitted:</p> <ul style="list-style-type: none"> <li>' ' Not the last call for the current job. A further JCL card follows with the next CALL statement. The JCL cards are collected into the RJE buffer.</li> <li>B BS2000/OSD and OS/390 only: Last call for the current job.</li> </ul> <p>Under BS2000/OSD: The job is generated, written to the dataset, but not started automatically.</p> <p>Under OS/390 (batch and TSO, IMS/TM and CICS): The job is written to the internal reader dataset but not submitted. If function <b>L</b> is called subsequently, the internal reader is closed and the job(s) is submittedand. In addition, the internal reader is closed and the job is submitted:</p> <ul style="list-style-type: none"> <li>on a screen I/O (IMS/TM), or</li> <li>on session termination (OS/390 batch, TSO and IMS/TM).</li> </ul> <ul style="list-style-type: none"> <li>C Flush the current job. The job is not submitted to the system. (Under BS2000/OSD, no dataset is created.)</li> <li>L Last call for the current job. The job is submitted to the system.</li> </ul> <p>BS2000/OSD environments: see Additional Values for the <i>parm3</i> Parameter.</p>
<i>parm4</i>	A 2-byte binary field in which NATRJE returns a response code.

## Example Programs

Below are Natural example programs for use under the operating systems:

- OS/390
- VSE/ESA
- BS2000/OSD (Example 1 and Example 2)

### Example Program - OS/390:

The following is a Natural example program that submits, in one call to NATRJE, a three-card JCL stream.

```

DEFINE DATA LOCAL
01 COUNT (B4)
01 FLAG (A1)
01 RETHEX (B2)
01 CARDS (A240)
01 REDEFINE CARDS
    02 CARD1 (A80)
    02 CARD2 (A80)
    02 CARD3 (A80)
END-DEFINE
MOVE '//JOB JOB CLASS=G,MSGCLASS=X' TO CARD1
MOVE '//XXX EXEC PGM=IEFBR14' TO CARD2
MOVE '//DD1 DD DSN=NATRJE.SOURCE,DISP=SHR' TO CARD3
MOVE 3 TO COUNT
MOVE 'L' TO FLAG
CALL 'NATRJE' CARDS COUNT FLAG RETHEX
IF RETHEX = H'0000'
    WRITE 'JOB submitted successfully'
ELSE
    WRITE 'ERROR from NATRJE' RETHEX
END-IF
END

```

### Example Program - VSE/ESA:

The following is a Natural example program that submits, in three calls to NATRJE, a seven-card JCL stream.

```

DEFINE DATA LOCAL
01 COUNT (B4)
01 FLAG (A1)
01 RETHEX (B2)
01 CARDS (A240)
01 REDEFINE CARDS
    02 CARD1 (A80)
    02 CARD2 (A80)
    02 CARD3 (A80)
END-DEFINE
MOVE '* $$ JOB JNM=DSERV,CLASS=0,DISP=D' TO CARD1
MOVE '* $$ LST CLASS=A,DISP=D' TO CARD2
MOVE '// JOB DSERV TO DSERV SOURCE MEMBERS' TO CARD3
MOVE 3 TO COUNT
CALL 'NATRJE' CARDS COUNT FLAG RETHEX
PERFORM RETCODE-CHECK
MOVE '// EXEC PROC=NATSPLP' TO CARD1
MOVE '// EXEC DSERV' TO CARD2
MOVE ' DSPLYS SD' TO CARD3
MOVE 3 TO COUNT
CALL 'NATRJE' CARDS COUNT FLAG RETHEX
PERFORM RETCODE-CHECK

```

```

MOVE '/' TO CARD1
MOVE '&' TO CARD2
MOVE '* $$ EOJ' TO CARD3
MOVE 3 TO COUNT
MOVE 'L' TO FLAG
CALL 'NATRJE' CARDS COUNT FLAG RETHEX
DEFINE SUBROUTINE RETCODE-CHECK
IF RETHEX NE H'0000'
    WRITE 'ERROR from NATRJE:' RETHEX
STOP
END-IF
END-SUBROUTINE
END

```

**Example Program 1 - BS2000/OSD:**

The following is a Natural example program that submits, in three calls to NATRJE, a nine-card JCL stream.

```

DEFINE DATA LOCAL
    01 COUNT (B4)
    01 FLAG (A1)
    01 RETHEX (B2)
    01 CARDS (A240)
    01 REDEFINE CARDS
        02 CARD1 (A80)
        02 CARD2 (A80)
        02 CARD3 (A80)
END-DEFINE
MOVE '/LOGON' TO CARD1
MOVE '/SYSFILE SYSDTA=(SYSCMD)' TO CARD2
MOVE '/SYSFILE SYSIPT =IPT.PARM' TO CARD 3
MOVE 3 TO COUNT
CALL 'NATRJE' CARDS COUNT FLAG RETHEX
    IF RETHEX NE H'0000' DO
        WRITE RETHEX (EM=HH)
    END-IF
MOVE '/SETSW ON=2' TO CARD1
MOVE '/EXEC NATB21' TO CARD2
MOVE 'LOGON APPLIC' TO CARD3
MOVE 3 TO COUNT
CALL 'NATRJE' CARDS COUNT FLAG RETHEX
    IF RETHEX NE H'000' DO
        ...
        ...
    END-IF
MOVE 'RUNPGM' TO CARD1
MOVE 'FIN' TO CARD2
MOVE '/LOGOFF' TO CARD3
MOVE 3 TO COUNT
MOVE 'L' TO FLAG
CALL 'NATRJE' CARDS COUNT FLAG RETHEX
    ...
    ...
    ...
END

```

**Example Program 2 - BS2000/OSD:**

The following is a Natural example program that submits, in one call to NATRJE, a nine-card JCL stream.

```

DEFINE DATA LOCAL
  01 COUNT  (B4)
  01 FLAG   (A1)
  01 RETHEX (B2)
  01 CARD1  (A80)
  01 CARD2  (A80)
  01 CARD3  (A80)
  01 CARD4  (A80)
  . . .
  01 CARD9  (A80)
END-DEFINE
MOVE '/LOGON' TO CARD1
MOVE '/SYSFILE SYSDTA=(SYSCMD)' TO CARD2
. . .
MOVE '/LOGOFF' TO CARD9
MOVE 9 TO COUNT
MOVE 'L' TO FLAG
CALL 'NATRJE' CARD1 COUNT FLAG RETHEX
. . .
END
    
```

## NATRJE Return Codes

A CALL to the module NATRJE results in one of the following return codes being returned in the fourth parameter of the CALL statement. There are return codes that apply to all environments and additional codes that are dependent on the operating system:

- Return Codes Common to all Environments
- Additional Return Codes for VSE/ESA
- Additional Return Codes for BS2000/OSD

### Return Codes Common to all Environments

Hexadecimal	Decimal	Meaning
00	00	Normal Return
04	04	RJE utility not available
08	08	RJE utility disabled; a possible reason is that the RJESIZE parameter is set to 0
0C	12	Invalid Number of JCL cards
10	16	Invalid Function Code
14	20	No RJE Buffer Space available
18	24	Invalid Number of Parameters
1C	28	I/O Error during Submit
20	32	Job flushed by user exit NREXPG (see also NATRJE User Exit below)

### Additional Return Codes for VSE/ESA

01xx	PUTSPOOL Error, xx is R15 Contents
02xx	PUTSPOOL Error, xx is XECB+4
03xx	XECBTAB Define Error, xx is Return Code
04xx	XECBTAB Delete Error, xx is Return Code

### Additional Return Codes for BS2000/OSD

9001	No RJE buffer found
9002	No buffer space available
9003	Missing LOGON command
9004	Only LOGON cards generated
9005	Too many LOGON parameters
D010	Error in ENTER macro
Dxxx	Operating system error: The error message is sent directly to the user program; the BS2000/OSD HELP command provides additional information.

## NATRJE User Exit

A user exit capability for Natural Remote Job Entry is provided. After the job is complete, each JCL card is passed to the exit before it is submitted to the operating system. The following data are available to the exit:

- the JCL card to be submitted,
- a return code field,
- the name of the Natural program currently being executed,
- the Natural user identification,
- a 240-byte work area.

After each call, the exit passes a return code to NATRJE indicating one of the following events:

Code	Explanation
0	Submission: the card is submitted; the exit may modify the card before submission.
4	Termination: the card is submitted; the exit is disabled for further cards of the current job.
8	Insertion: the card is skipped (based on the assumption that it contains only an INSERT character); additional specified cards are submitted.
10	Deletion: the card is not submitted.
12	The current job is flushed.

An example of the user exit, called NREXPG, is available as member XNATRJE in the Natural source library. The exit can be assembled and linked according to the rules of programs specified as CSTATIC. However, a CSTATIC entry for NREXPG is not required.

## NATRJE Features Applicable to UTM/TIAM

Below is information on:

- Additional Values for the parm3 Parameter
- Name of BS2000/OSD Dataset

### Additional Values for the parm3 Parameter

Value	Explanation
A	Combination of values T and E.
E	The job is generated and completed. Before submission to the BS2000/OSD operating system, the parameter ERASE=YES is added to the ENTER parameter.
T	The job is generated and completed. Before submission to the BS2000/OSD operating system, a time limit is calculated using the Natural MT parameter. If MT is set to 0, the time limit is generated as NTL. The calculated time limit is added to the ENTER parameter via the TIME= operand.

When using the values T, E or A, NATRJE does not check whether the parameters TIME= or ERASE= exist in the user-created LOGON cards.

### Name of BS2000/OSD Dataset

The name of the BS2000/OSD dataset created by NATRJE for the JCL cards is as follows:

*E.DDMMYY.HHMMSSSS.program-name.user-id*

Parameter	Specifies
DD	The day of the dataset creation.
MM	The month of the dataset creation.
YY	The year of the dataset creation.
HH	The hour of the dataset creation.
MM	The minute of the dataset creation.
SSSS	The seconds and milliseconds of the dataset creation.
<i>program-name</i>	The name of the Natural program that creates the dataset.
<i>user-id</i>	The corresponding Natural user ID.

# NATUNLD/NATLOAD Utilities - Overview

The library SYSUNLD contains two utilities: NATUNLD and NATLOAD.

The utilities NATUNLD and NATLOAD are available with Natural on mainframes and UNIX. The description in this section applies to NATUNLD and NATLOAD on any of these platforms.

The usage of NATUNLD and NATLOAD can be controlled with Natural Security, so that some NATUNLD/NATLOAD functions may not be available to you.

If you have specified MENU as startup program, set the parameter "Batch execution" to **N** in the Natural Security library profile of the library SYSUNLD.

The NATUNLD/NATLOAD utilities documentation covers the following topics:

- **Introduction**      General Information on the use of NATUNLD and NATLOAD.  
                         Invoking NATUNLD or NATLOAD.
- **NATUNLD**      Unloading Natural programming objects, error messages and DDMs from system files to  
  Utility            a work file.
- **NATLOAD**      Loading Natural programming objects, error messages and DDMs from a work file to  
  Utility            system files.
- **Direct Commands** Executing direct commands available with NATUNLD and NATLOAD.

# NATUNLD/NATLOAD - Introduction

- General Information
  - Invoking NATUNLD or NATLOAD
  - Options and Specifications
- 

## General Information

The utilities NATUNLD and NATLOAD are contained in the library SYSUNLD.

- NATUNLD is used to unload Natural programming objects, error messages and DDMs from system files to a work file.
- NATLOAD is used to load Natural programming objects, error messages and DDMs from a work file into system files.

The unloaded/loaded programming objects can be saved objects, cataloged objects and stowed objects. The programming objects to be unloaded can also be objects defined in a Predict set.

The unloaded/loaded error messages can be Natural system messages or user-written messages.

NATUNLD unloads from the following system files:

- Programming objects are unloaded from the Natural system files FNAT and FUSER. For objects defined in a Predict set, the corresponding set information is read from the Predict system file FDIC.
- Error messages are unloaded from the Natural system files FNAT and FUSER.
- DDMs are unloaded from the Predict system file FDIC.

NATUNLD generates variable-length records with a maximum of 252 characters per record. These records are written to Natural Work File 1 in a format which can be used for loading with NATLOAD.

To write records to the work file with a fixed record length of 252 characters per record, in the WITH clause of the NATUNLD direct command, specify keyword FIXEDLENGTH as described in NATUNLD Direct Command Syntax.

NATLOAD reads the records from Natural Work File 1. NATLOAD can only load work files created by NATUNLD.

**Note:**

If the records are unloaded/loaded with Entire Connection, Work File 7 is used instead of Work File 1.

NATLOAD loads into the following system files:

- Programming objects are loaded into the Natural system files FNAT and FUSER: Objects in libraries whose names begin with SYS (except for the library SYSTEM) are loaded into the FNAT file; objects in all other libraries are loaded into the FUSER file. If the library SYSTEM is loaded from a Software AG installation dataset, it is loaded into the FNAT file, otherwise it is loaded into the FUSER file.
- Error messages are loaded into the Natural system files FNAT and FUSER.
- DDMs are loaded into the Predict system file FDIC.

In addition, NATUNLD allows you to write delete instructions for specific objects to the work file. When the work file is read with NATLOAD, these instructions cause the objects concerned to be deleted (see the delete instructions in the section NATUNLD Utility) from the target environment.

The utilities NATUNLD and NATLOAD can be used online as well as in batch mode.

**Note:**

If you enter a Natural system command in a command line within NATUNLD or NATLOAD, you have to specify two slashes (//) before the command to identify it as a system command.

## Invoking NATUNLD or NATLOAD

### ▶ To invoke NATUNLD/NATLOAD from the SYSUNLD Main Menu

1. Enter the system command SYSUNLD.  
The SYSUNLD Main Menu is displayed.
2. Select the Unload or Load/Scan function.  
The main menu of NATUNLD or NATLOAD is displayed.

### ▶ To invoke NATUNLD/NATLOAD directly

- See the relevant sections in NATUNLD and NATLOAD.

## Options and Specifications

This section provides explanations and instructions on recurring options and specifications used in the NATUNLD/NATLOAD documentation. Be aware that there are exceptions which are documented separately in the particular sections.

Below is information on:

- Names and Ranges
- Dates and Ranges
- File Assignments

### Names and Ranges

To select Natural libraries, programming objects and DDMs, specify a name or a range of names. Options are:

	<i>value</i> is any combination of one or more characters.
<i>value</i>	Select a single item.
*	Select all items.
<i>value</i> <	Select all items whose names are less/equal <i>value</i> .
<i>value</i> >	Select all items whose names are greater/equal <i>value</i> .
<i>value</i> *	Select all items whose names begin with <i>value</i> .

### Dates and Ranges

To select Natural programming objects, specify a date on which or a date range within which an object was saved.

The date has to be specified according to the setting of the DTFORM profile parameter as described in your Natural Operations documentation; the time has to be specified in the format HH:II (HH = hours, II = minutes).

As abbreviations for special dates or date ranges, the following strings can be entered:

String	Explanation
<u>T</u> ODAY	The date of the current day. The day can be followed by <b>+nnnn</b> or <b>-nnnn</b> (where <i>nnnn</i> are numeric values). The resulting date is computed as the date of the current day plus or minus <i>nnnn</i> days.
<u>Y</u> ESTERDAY	The date of the day before the current day.
<u>M</u> ONTH	The date range of the current month.
<u>Y</u> EAR	The date range of the current year.

## File Assignments

To unload or load from an FUSER or FDIC file other than the current one, either enter the command FILES in the command line or press PF10 (Files). The "File Assignments for NATUNLD/NATLOAD" window will then be displayed. In this window, you can specify a different FUSER and FDIC file.

The database ID (DBID) and file number (FNR) of the system file from or to which a programming object/DDM is unloaded or loaded is determined as follows:

- Under Natural Security, for each library, the DBID/FNR specified in the corresponding library security profile is always used.
- Without Natural Security, and for libraries in whose security profiles no DBID/FNR is specified, the DBIDs/FNRs from your current system files are used when you invoke NATUNLD or NATLOAD.

### Note for NATUNLD:

In the "File Assignments for NATUNLD" window, you can also specify a user ID (Set User) for a Predict set to be unloaded. See also Set Number in the section Parameters for Programming Objects (NATUNLD Utility).

# NATUNLD Utility

- Invoking NATUNLD
  - Programming Objects
  - DDMs
  - Error Messages
  - Delete Instructions for Programming Objects
  - Delete Instructions for DDMs
  - Delete Instructions for Error Messages
  - Executing NATUNLD in Batch Mode
  - NATUNLD Report
- 

## Invoking NATUNLD

### To invoke the NATUNLD utility

- In batch mode (see also Executing NATUNLD in Batch Mode), enter a direct command.
- Online, enter the system command NATUNLD.  
The Unload Programming Objects menu of the NATUNLD utility is displayed.

## Programming Objects

When you invoke NATUNLD online, the Unload Programming Objects screen is displayed. For unloading in batch mode, see Executing NATUNLD in Batch Mode.

Below is information on:

- Functions for Programming Objects
- Parameters for Programming Objects

## Functions for Programming Objects

From the Unload Programming Objects screen, you can invoke the following functions by entering a code, or a command in the command line, or by pressing a PF key:

Code	PF Key	Function	Explanation
	Command		
A		Unload All/Individual Objects	Unloads objects which exist in any form: in saved form, or cataloged form or both.
S		Unload only Saved Objects	Unloads only objects which exist in saved form.
C		Unload only Cataloged Objects	Unloads objects which exist in cataloged form.
W		Unload Saved and Cataloged Objects	Unloads objects which exist in both saved and cataloged form. (*)
B		Unload Stowed Objects	Unloads objects which exist in both saved and cataloged form and which were saved and cataloged at the same time. The check that the saving and cataloging of an object occurred at the same time ensures that the saved and cataloged form of an object actually belong together. (*)
	PF4 (E-Msg) ERROR	Unload Error Messages	See the section Error Messages.
	PF5 (DDMs) DDM	Unload DDMs	See the section DDMs.
	PF7 (Del.) DELETE	Unload Delete Instructions	See the section Delete Instructions for Programming Objects.
	PF10 (Files) FILES	File Assignments for NATUNLD	See File Assignments in the section Introduction.

\* Objects of type copycode, text and recording - although they can exist only in saved, but not in cataloged form - are also processed by these functions.

If you want to unload DDMs, you press PF5 (DDMs) on the Unload Programming Objects screen to invoke the function Unload DDMs, as described in the section DDMs.

If you want to write delete instructions to the work file, you press PF7 (Del.) on the Unload Programming Objects screen to invoke the function Unload Delete Instructions, as described in the section Delete Instructions for Programming Objects.

Each of the above functions unloads Natural programming objects from one or more libraries on the system files FNAT or FUSER. Libraries whose names begin with SYS (except for the library SYSTEM) are, by default, unloaded from the FNAT file; all other libraries are unloaded from the FUSER file. See also File Assignments in the section Introduction.

To execute one of the functions on the Unload Programming Objects screen, enter the corresponding function code and, if required, the parameters described below.

## Parameters for Programming Objects

The Unload Programming Objects screen provides the following parameters:

Field	Explanation
From Library	<p>The name of the library from which the objects are to be unloaded. By default, the library is selected from which NATUNLD was called. To select multiple libraries, see Names and Ranges in the section Introduction.</p> <p><b>Note:</b> Under Natural Security, the setting of the NATUNLD utility profile or the Utility option in the corresponding library profile determines whether you are authorized to unload objects from a selected library. If the Natural Security option Transition Period Logon is set to <b>Y</b>, objects from libraries which have not been defined to Natural Security can also be unloaded.</p>
To Library	<p>The name of the library into which the unloaded objects are to be loaded with NATLOAD. If you leave this field blank or enter an asterisk (*), the name of the library specified in the From Library field is used.</p> <p><b>Note:</b> If you have specified a range as library in the From Library field, the specification in the To Library field name may be a range too, but the number of characters before the asterisk (*) <b>must not</b> exceed the number of characters before the asterisk in the From Library field.</p>
Object Name	<p>The name of the object to be unloaded. If you leave this field blank or enter an asterisk (*), all objects in the library (or libraries) specified under From Library are unloaded.</p> <p>Within the object name, you can specify any combination of asterisk notation (*) and wildcard notation (?) in the same manner as described for the system command LIST (Natural System Command Reference documentation) to unload a specific range of objects.</p> <p>See also Names and Ranges in the section Introduction.</p>
User ID	<p>If you enter a user ID in this field, only those objects are unloaded which were saved or cataloged under this user ID. To specify a range of user IDs, see Names and Ranges in the section Introduction.</p>
Symbol Table	<p>Specifies whether the objects' symbol tables are also to be unloaded:</p> <p>Y The selected objects will be unloaded with their corresponding symbol tables (this is the default).</p> <p>N The selected objects will be unloaded without their corresponding symbol tables. This will reduce the amount of disk storage required. However, this is only useful for a production environment, as several application development functions which require the symbol tables will then not be available; in addition, the profile parameter RECAT=ON will not apply.</p>
PC Download	<p>Only applies if Entire Connection is installed and if you have entered the terminal command "%+" before invoking NATUNLD; otherwise, you cannot enter anything in this field.</p> <p>In this field, you specify whether the work file containing the specified objects is to be downloaded to a PC using Work File 7.</p>

Field	Explanation
Set Number	<p>It is possible to unload objects that are defined in a Predict set as created with the Predict utility List XREF.</p> <p>To unload the object contained in a Predict set, specify the number of that set in this field. The set is identified by the set number, the library ID as specified in the From Library field, and the ID of the user who created the set as specified in the Set User field in the "File Assignments for NATUNLD" window. See also File Assignments in the section Introduction. If you specify a set number, you can only specify the name of a single library (no range) in the From Library field.</p> <p>By default, your current user ID (that is, the value of the system variable *USER) is used as user ID for the set. To change the user ID for the set, either enter the command FILES in the command line or press PF10 (Files) and then overwrite the value of the Set User field in the "File Assignments for NATUNLD" window. See also File Assignments in the section Introduction.</p>
Xref Data	<p>Only applies with Predict and if the Predict Active References feature is installed.</p> <p>In this field, you specify whether the corresponding cross-reference data are to be unloaded with the objects:</p> <p>Y If a cataloged object has cross-reference data, these are unloaded with the object.  N Only the cataloged object is unloaded, but any cross-reference data are ignored (this is the default).</p>
Object Type	<p>The type of objects to be unloaded.</p> <p>If you leave this field blank or enter an asterisk (*), all objects with the names specified under Object Name are unloaded regardless of their types.  For a selection list of possible object types, either enter a question mark (?) in this field or press PF1 (Help).</p> <p><b>Note:</b>  You can specify several object types at the same time and in any sequence; for example, specifying PAM unloads programs, parameter data areas and maps.</p>
Date/Time From	<p>Only those objects are unloaded which were saved or cataloged on or after/before this date and time.</p>
Date/Time To	<p>See Dates and Ranges in the section Introduction for further details.</p>

## DDMs

### To invoke the Unload DDMs function

- Enter the command DDM in the command line.  
Or press PF5 (DDMs) on the Unload Programming Objects or Unload Error Messages screen.  
The Unload DDMs screen is displayed.

For unloading in batch mode, see Executing NATUNLD in Batch Mode.

Below is information on:

- Functions for DDMs
- Parameters for DDMs

## Functions for DDMs

The Unload DDMs function on mainframes unloads DDMs from the system file FDIC. See also File Assignments in the section Introduction.

From the Unload DDMs screen, invoke the following functions by entering a code, or a command in the command line, or by pressing a PF key:

Code	PF Key	Function	Explanation
	Command		
A		Unload DDMs	Unloads all DDMs.
	PF4 (E-Msg) ERROR	Unload Error Messages	See the section Error Messages.
	PF5 (Objct) OBJECTS	Unload Programming Objects	See the section Programming Objects.
	PF7 (Del.) DELETE	Unload Delete Instructions	See the section Delete Instructions for DDMs.
	PF10 (Files) FILES	File Assignments for NATUNLD	See File Assignments in the section Introduction.

## Parameters for DDMs

To execute one of the functions on the Unload DDMs screen, enter the corresponding function code and, if required, the following parameters:

Field	Explanation
DDM Name	The name of the DDM to be unloaded.  To unload all DDMs from the system file, leave this field blank or enter an asterisk (*). To unload multiple DDMs, use the same range notations as described for Object Name in Parameters for Programming Objects.
DDM DBID	To unload only DDMs which have a specific database ID, enter that database ID in this field. Database ID is the ID of the physical database file of which the DDM is the logical representation.
DDM FNR	To unload only DDMs which have a specific file number, enter that file number in this field. File number is the number of the physical database file of which the DDM is the logical representation.
PC Download	Here the same applies as described for PC Download in Parameters for Programming Objects.

## Error Messages

Each of the functions below unloads Natural system error messages from the system file FNAT, and user-defined error messages from one or more libraries on the system files FNAT or FUSER; libraries whose names begin with SYS (except for the library SYSTEM) are, by default, unloaded from the FNAT file; all other libraries are unloaded from the FUSER file. See also File Assignments in the section Introduction.

**▶ To invoke the Unload Error Message function**

- Enter the command ERROR in the command line.  
Or press PF4 (E-Msg) on the Unload Programming Objects or Unload DDMs screen.  
The Unload Error Messages screen is displayed.

For unloading in batch mode, see Executing NATUNLD in Batch Mode.

Below is information on:

- Functions for Error Messages
- Parameters for Error Messages

### Functions for Error Messages

From the Unload Error Messages screen, invoke the following functions by entering a code, or a command in the command line, or by pressing a PF key:

Code	PF Key	Function	Explanation
	Command		
A		Unload Short and Long Error Messages	Unloads error messages which exist in short and/or long form.
S		Unload only Short Error Messages	Unloads only error messages which exist in short form.
L		Unload only Long Error Messages	Unloads only error messages which exist in long form.
H		Unload Natural Help Texts	Natural help texts from the FNAT system file.
	PF4 (Objct) OBJECTS	Unload Programming Objects	See the section Programming Objects.
	PF5 (DDMs) DDM	Unload DDMs	See the section DDMs.
	PF7 (Del.) DELETE	Unload Delete Instructions for Error Messages	See the section Delete Instructions for Error Messages.
	PF10 (Files) FILES	File Assignments for NATUNLD	See the section File Assignments in the Introduction.

### Parameters for Error Messages

To execute one of the functions on the Unload Error Messages screen, enter the corresponding function code and, if required, the following parameters (the function Unload Natural Help Texts requires no parameters):

Field	Explanation
User/Natural	Type of message to be unloaded:  U user-defined error messages, N Natural system messages.
From Library	Only applies to user-defined error messages.  The name of the library from which the user-defined error messages are to be unloaded. Here the same applies as described for From Library in Parameters for Programming Objects.
To Library	Only applies to user-defined error messages.  The name of the library into which the user-defined error messages are to be loaded with NATLOAD. Here the same applies as described for To Library in Parameters for Programming Objects.
Error Number	The range of error message numbers to be unloaded.
Language Codes	The language code(s) of the error messages to be unloaded. For valid language codes, see the system variable *LANGUAGE in the Natural System Variables documentation. By default (*), messages in any language are unloaded.
PC Download	Here the same applies as described for PC Download in Parameters for Programming Objects.

## Delete Instructions for Programming Objects

Use this function to write delete instructions for specific programming objects to the work file. When these instructions are loaded with NATLOAD into the target environment, they cause the specified objects to be deleted from the target environment.

### To invoke the Unload Delete Instructions function

- Enter the command DELETE in the command line.  
Or press PF7 (Del.) on the Unload Programming Objects screen.  
The Unload Delete Instructions screen is displayed.

Below is information on:

- Functions for Delete Instructions - Programming Objects
- Parameters for Delete Instructions - Programming Objects

## Functions for Delete Instructions - Programming Objects

From the Unload Delete Instructions screen, you can invoke the following functions by entering a code, or a command in the command line, or by pressing a PF key:

Code	PF Key	Function	Explanation
	Command		
A		Delete All/Individual Objects	Delete instructions for objects which exist in any form: in saved form, or cataloged form or both.
S		Delete only Saved Objects	Delete instructions for objects which exist in saved form.
C		Delete only Cataloged Objects	Delete instructions for objects which exist in cataloged form.
	PF4 (E-Msg) ERROR	Unload Delete Instructions for Error Messages	See the section Delete Instructions for Error Messages.
	PF5 (DDMs) DDM	Unload Delete Instructions for DDMs	See the section Delete Instructions for DDMs.
	PF7 (Unld) UNLOAD	Unload Programming Objects	See the section Programming Objects.
	PF10 (Files) FILES	File Assignments for NATUNLD	See File Assignments in the section Introduction.

### Parameters for Delete Instructions - Programming Objects

To execute one of the functions, enter the appropriate function code and the following parameters:

Field	Explanation
Library	The name of the library from which the objects are to be deleted. You can only specify the name of a single library; a range of multiple libraries cannot be specified. By default, the library is selected from which NATUNLD was called.
Object Name	The name of the object to be deleted. If you leave this field blank or enter an asterisk (*), all objects in the specified library will be deleted. To delete multiple objects, see Names and Ranges in the section Introduction.
PC Download	Here the same applies as described for PC Download in Parameters for Programming Objects.

### Delete Instructions for DDMs

Use this function to write delete instructions for specific DDMs to the work file. When these instructions are loaded with NATLOAD into the target environment, they cause the specified DDMs to be deleted from the target environment.

 **To invoke the Unload Delete Instructions for DDMs function**

- Enter the command DELETE in the command line.  
Or press PF7 (Del.) on the Unload DDMs screen.  
The "Unload Delete Instructions for DDMs" screen is displayed.

Below is information on:

- Functions for Delete Instructions - DDMs
- Parameters for Delete Instructions - DDMs

## Functions for Delete Instructions - DDMs

From the Unload Delete Instructions for DDMs screen, invoke the following functions by entering a code, or a command in the command line, or by pressing a PF key:

Code	PF Key	Function	Explanation
	Command		
A		Delete DDMs	All DDMs.
	PF4 (E-Msg) ERROR	Unload Delete Instructions for Error Messages	See the section Delete Instructions for Error Messages.
	PF5 (Objct) OBJECTS	Unload Delete Instructions for Programming Objects	See the section Delete Instructions for Programming Objects.
	PF7 (Unld) UNLOAD	Unload DDMs	See the section DDMs.

## Parameters for Delete Instructions - DDMs

To execute one of the functions, enter the appropriate function code and the following parameters:

Field	Explanation
DDM Name	The name of the DDM to be deleted. If you leave this field blank or enter an asterisk (*), all DDMs will be deleted from the target system file. To delete multiple objects, see Names and Ranges in the section Introduction.
PC Download	Here the same applies as described for PC Download in Parameters for Programming Objects.

## Delete Instructions for Error Messages

Use this function to write delete instructions for specific error messages to the work file. When these instructions are loaded with NATLOAD into the target environment, they cause the specified error messages be deleted from the target environment.

### To invoke the Unload Delete Instructions function for error messages

- Enter the command DELETE in the command line.  
Or press PF7 (Del.) on the Unload Error Messages screen.  
The Unload Delete Instructions screen for error messages is displayed.

Below is information on:

- Functions for Delete Instructions - Error Messages
- Parameters for Delete Instructions - Error Messages

## Functions for Delete Instructions - Error Messages

From the Unload Delete Instructions screen for error messages you invoke the following functions by entering a code, or a command in the command line, or by pressing a PF key:

Code	PF Key Command	Function	Explanation
A		Delete Short and Long Error Messages	Delete instructions for error messages which exist in short and/or long form.
L		Delete only Long Error Messages	Delete instructions for error messages which exist in long form.
	PF4 (Object) OBJECTS	Unload Delete Instructions for Programming Objects	See the section Delete Instructions for Programming Objects.
	PF5 (DDMs) DDM	Unload Delete Instructions for DDMs	See the section Delete Instructions for DDMs.
	PF7 (Unld) UNLOAD	Unload Error Messages	See the section Error Messages.
	PF10 (Files) FILES	File Assignments for NATUNLD	See File Assignments in the section Introduction.

### Parameters for Delete Instructions - Error Messages

To execute one of the functions, enter the appropriate function code and the following parameters:

Field	Explanation
User/Natural	Type of message to be deleted:  U user-defined error messages, N Natural system messages.
Library	Applies to user-defined error messages only.  The name of the library from which the messages are to be deleted. You can only specify the name of a single library; a range of multiple libraries cannot be specified. By default, the library is selected from which NATUNLD was called.
Error Number	The range of error message numbers to be deleted.
Language Codes	The language code(s) of the error messages to be deleted. For valid language codes, see the system variable *LANGUAGE in the Natural System Variables documentation. By default (*), messages in any language are deleted.
PC Download	Here the same applies as described for PC Download in Parameters for Programming Objects.

### Executing NATUNLD in Batch Mode

For the execution of NATUNLD in batch mode, use direct commands.

When you use a direct command, observe the following rules:

- The profile parameters PS and LS should be set to at least PS=78 and LS=220.
- The keyword NATUNLD can be placed in a line by itself.
- To separate the individual parameters of a direct command from one another, use either blanks or the input delimiter character as defined by the session parameter ID; the default is comma.
- If the string of parameters is longer than a single line, enter the character defined with the session parameter

CF (default is %) at the end of every line that belongs to the command. This indicates continuation on the next line.

However, this is only possible if you specify the command NATUNLD in a line by itself. That is, you cannot use CF, if you enter NATUNLD in the same line where a multi-line command starts.

- To indicate the end of the command input, specify a line containing the keyword END or one of its synonyms.

The syntax of the NATUNLD direct commands and Examples of Direct Commands for NATUNLD in Batch Mode are shown in the section NATUNLD/NATLOAD Direct Commands.

Below is information on:

- Condition Codes and User Exits in Batch Mode

## Condition Codes and User Exits in Batch Mode

Below is a description of NATUNLD condition codes and user exits available in batch mode.

The sources of NATUNLD user exits are named U-S-EX $nn$ , where  $nn$  denotes the number of the user exit. The user exits are delivered in the Natural system library SYSUNLD.

To make a user exit available, catalog it under the name UNLDEX $nn$ , either in the Natural system library SYSUNLD or in one of its steplibs.

The name of each user exit source is different from the name of the corresponding cataloged object. This guarantees that the object is not affected if the user exit source is overwritten by an installation update.

NATUNLD processing in batch mode terminates with one of the following condition codes:

Code	Explanation
0	Unloading executed successfully.
37	Only applies in connection with the user exit UNLDEX01 (*). This user exit can be used to terminate unloading with Condition Code 37 if the objects requested cannot be found. For details, see the source code of UNLDEX01.  If the user exit UNLDEX01 is not available, unloading continues and no Condition Code 37 is issued.
38	Unloading request rejected by Natural Security: If you attempt to unload objects from a library you are not allowed to access due to Natural Security definitions, NATUNLD terminates with Condition Code 38. If available, user exit UNLDEX02 will be invoked. Here you can specify whether to continue unloading without that library or terminate NATUNLD with Condition Code 38.  The source code of user exit UNLDEX02 is provided under the name U-S-EX02 in the library SYSUNLD.
40	An error occurred; unloading terminated.  With Condition Code 40, the user exit UNLDEX01 (*), if available, will be invoked at the termination of processing.

\* The source code of the user exit UNLDEX01 is provided under the name U-S-EX01 in the library SYSUNLD.

## NATUNLD Report

The NATUNLD utility generates (online and in batch mode) a report containing a list of the objects that were unloaded.

When you leave NATUNLD after the report has been displayed, a statistical report will be displayed listing the number of objects processed.

See also NATUNLD/NATLOAD Commands for Reports in the section NATUNLD/NATLOAD Direct Commands.

# NATLOAD Utility

- Invoking NATLOAD
- Functions
- Parameters
- Verifying Subroutines and Classes
- Executing NATLOAD in Batch Mode
- NATLOAD Report

## Invoking NATLOAD

### To invoke the NATLOAD utility

- In batch mode, enter a direct command (see also Executing NATLOAD in Batch Mode).
- Online, enter the system command NATLOAD.  
The Load Functions menu is displayed.

## Functions

From the Load Functions menu, invoke the following functions by entering a code, or a command in the command line, or by pressing a PF key:

Code	PF Key	Function	Explanation
A		Load All Objects, DDMs and Error Messages	Loads programming objects, DDMs and error messages.
L		Load Objects	Loads programming objects only.
D		Load DDMs	Loads DDMs only.
E		Load Error Messages	Loads error messages only.
S		Scan All Objects, DDMs and Error Messages	Scans programming objects, DDMs and error messages.
O		Scan Objects	Scans programming objects only.
V		Scan DDMs	Scans DDMs only.
M		Scan Error Messages	Scans error messages only
	PF10 (Files)	File Assignments for NATLOAD	See File Assignments in the section Introduction.

Below is information on:

- Scan
- Load

### Scan

With the four Scan functions listed above, you can scan the work file to be loaded to find out what it contains, before you actually start to load its contents.

The range of programming objects, DDMs and/or error messages to be scanned can be narrowed down by the parameters you can specify on the Scan screens. The parameters for the Scan functions correspond to those of the Load functions, as described below.

## Load

With the four Load functions listed above, you load Natural programming objects, error messages (and help texts), DDMs and delete instructions from a work file created by NATUNLD into system files:

- Programming objects are loaded into the Natural system files FNAT and FUSER. Objects in libraries whose names begin with SYS (except for the library SYSTEM) are loaded into the FNAT file; objects in all other libraries are loaded into the FUSER file. If the library SYSTEM is loaded from a Software AG installation dataset, it is loaded into the FNAT file, otherwise it is loaded into the FUSER file. See also File Assignments in the section Introduction.
- Natural system error messages are loaded into the system file FNAT. User-defined error messages are loaded into the files FNAT or FUSER like programming objects (see above). See also File Assignments in the section Introduction.
- DDMs are loaded into the Predict system file FDIC. See also File Assignments in the section Introduction.

When you select a load function from the Load Functions menu, the processing screen for that function will be displayed.

On that processing screen, enter a function code and, if required, the parameters described below to start the loading process.

### Note:

Some of the parameters apply only to programming objects or DDMs or error messages, and, therefore, do not appear on all processing screens.

## Parameters

The Load Functions screen provides the following parameters:

Field	Explanation
Load Except	<p>N This field is not used (this is the default).</p> <p>Y Field specifications are negated and all programming objects/DDMs/error messages <b>except</b> the specified ones are loaded. This applies to the following fields:</p> <p style="padding-left: 40px;">DDM Name    Date/Time From DDM DBID    Date/Time To DDM FNR     User ID Library      Error Number from/to Object Name   Language Codes Object Type</p> <p>Examples:</p> <p style="padding-left: 40px;">Load Except: Y        All programming objects in all libraries except those in the Object Name: *        library RPD are loaded. Library ...: RPD</p> <p style="padding-left: 40px;">Load Except: Y        All DDMs except those whose names begin with AA are DDM Name ..: AA*     loaded.</p> <p><b>Warning:</b> For delete instructions, the Load Except field is ignored. See also the field Allow Delete below.</p>
Object Name	<p>The name of the programming object which is to be loaded. If you leave this field blank or enter an asterisk (*), all objects in the libraries specified under Library are loaded. To load a specific range of multiple objects, you can use the same range notations as described for Object Name in the section NATUNLD Utility.</p>
Number	<p>The maximum number of objects to be loaded. Every programming object, error message and DDM that meets the selection criteria is counted against this limit. If a saved object and a cataloged object of the same name are loaded one immediately after the other, they are counted as one object. If a short error message and a long error message of the same number and language code are loaded one immediately after the other, they are counted as one object. A delete instruction is counted as one object. When the specified number is reached, loading is terminated with an appropriate message.</p>
Library	<p>The name of the library to be loaded. If you leave this field blank or enter an asterisk (*), all libraries are loaded.  To select a specific range of multiple libraries for loading, see Names and Ranges in the section Introduction.</p>

Field	Explanation
New Library	<p>If you leave this field empty, the programming objects are loaded into the target library as specified with the NATUNLD utility when the objects are unloaded.</p> <p>If you want the object to be loaded into a different library, you specify the name of that library in this field.</p> <p><b>Attention:</b></p> <p>If you have specified a range as library to be loaded, the new library name may be a range too, but the number of characters before the asterisk (*) <b>must not</b> exceed the number of characters before the asterisk in the Library field.</p>
Object Type	<p>The type(s) of programming objects to be loaded.</p> <p>If you leave this field blank or enter an asterisk (*), all object types of the object with the name specified under Object Name are loaded.</p> <p>For a selection list of possible object types, you either enter a question mark (?) in this field or press PF1 (Help).</p> <p><b>Note:</b></p> <p>Several programming object types can be specified at the same time and in any sequence. For example, specifying PAM loads programs, parameter data areas and maps.</p>
Xref Data	<p>Only applies with Predict and if the Predict Active References feature is installed.</p> <p>In this field, you specify whether cataloged objects are to be loaded together with their corresponding cross-reference data:</p> <p>N Ignores any existing cross-reference data for the cataloged object being loaded.</p> <p>S Loads cataloged object and its cross-reference data (if any).</p> <p>Y Loads cataloged object and its cross-reference data only if cross-reference data exist.</p> <p>F Loads cataloged object and its cross-reference data only if cross-reference data exist and if the object is documented in Predict.</p> <p>Under Natural Security, the Cross-Reference option in the respective library profile determines whether objects without cross-reference data can be loaded at all:                      If the Cross-Reference option is set to YES or FORCE, objects without cross-reference data are ignored by NATLOAD, regardless of the setting of the Xref data field (see also the XREF system command in the Natural System Command Reference documentation).</p>
S/C Type	<p>The form of programming objects to be loaded:</p> <p>A Both saved and/or cataloged objects (this is the default).</p> <p>S Saved objects only.</p> <p>C Cataloged objects only.</p>
User ID	<p>If you enter a user ID in this field, only those objects are loaded which were saved under this user ID.</p> <p>To specify a range of user IDs, see Names and Ranges in the section Introduction.</p>
Date/Time From	<p>Only those objects are loaded which were saved or cataloged on or after/before this date and time.</p>
Date/Time To	<p>See Dates and Ranges in the section Introduction for further details.</p>

Field	Explanation
DDM Name	<p>The name of the DDM to be loaded.</p> <p>If you leave this field blank or enter an asterisk (*), all DDMs are loaded.</p> <p>To load multiple DDMs, you can use the same range notations as described for Object Name in the section NATUNLD.</p>
DDM DBID	<p>Only DDMs with a specific database ID.</p> <p>Here the same applies as described for DDM DBID in the section NATUNLD Utility.</p>
DDM FNR	<p>Only DDMs with a specific file number.</p> <p>Here the same applies as described for DDM FNR in the section NATUNLD Utility.</p>
Error Msg Type	<p>The types of error message to be loaded:</p> <p>N Natural system error messages.</p> <p>U User-defined error messages.</p> <p>A All (Natural and user-defined) error messages.</p>
Short/Long/All	<p>The types of error message text to be loaded:</p> <p>A All (that is, short and long) error message texts.</p> <p>S Short message texts only.</p> <p>L Long message texts only.</p> <p>If you specify L, for each long error message, the corresponding short error message must be available. Otherwise, the long error message cannot be loaded.</p>
Error Number	The first and last numbers of the error message range to be loaded.
Language Codes	The language code(s) of the error messages to be loaded. For valid language codes, see the system variable *LANGUAGE in the Natural System Variables documentation.
Allow Delete	<p>This field is a safeguard against the accidental deletion of programming objects/error messages/ DDMs in the target system files. If the work file contains delete instructions (see Delete Instructions for Programming Objects, Delete Instructions for DDMs and Delete Instructions for Error Messages in the section NATUNLD Utility), this field determines whether the delete instructions are to be processed or not:</p> <p>N Delete instructions are not processed (this is the default).</p> <p>Y Delete instructions are processed and the objects concerned will be deleted from the target system files. If this field is set to Y, the entries in the fields Load Except and New Library are ignored; all other fields are evaluated to determine the range of objects to be deleted.</p>
PC Upload	<p>Only applies if Entire Connection is installed and you have entered the terminal command "%+" before invoking NATLOAD; otherwise you cannot enter anything in this field.</p> <p>In this field, you specify whether the work file containing the specified objects is to be uploaded from a PC.</p>
Replace	<p>Y Programming objects/error messages/DDMs which are already present on the Natural system file, are replaced by the newly loaded ones of the same names.</p> <p>N Programming objects/error messages/DDMs which are already present on the Natural system file are not replaced (this is the default).</p>

Field	Explanation
Check Date	<p>Only applies, if Replace is set to <b>Y</b>. It allows you to replace only "old" objects:</p> <p>N All existing objects will be overwritten (this is the default).</p> <p>Y Existing objects will be overwritten depending on their time stamps: only objects which were saved/cataloged before the objects of the same names to be loaded will be overwritten.</p>
Check Version	<p>Compares the version of the cataloged objects to be loaded with the current Natural version:</p> <p>N Performs no version check. All objects will be loaded (this is the default).</p> <p>Y Performs a version check. Objects cataloged under a Natural version higher than the current one will be rejected.</p>

## Verifying Subroutines and Classes

- Subroutine Names
- Class Names and GUIDs

### Subroutine Names

When a cataloged object of type subroutine is loaded, NATLOAD also verifies the name of the subroutine it contains, that is, the name used in the DEFINE SUBROUTINE statement. The cataloged subroutine cannot be loaded if an object of type subroutine which contains a subroutine of the same name, already exists in the target library. The subroutine will then not be loaded; however, NATLOAD processing will continue.

### Class Names and GUIDs

When a cataloged object of type class is loaded, NATLOAD also verifies the name of the class it contains, that is, the name used in the DEFINE CLASS statement. The cataloged class will not be loaded if an object of type class which contains a class of the same name or of the same GUID, already exists in the target library. NATLOAD processing, however, will continue.

## Executing NATLOAD in Batch Mode

For the execution of NATLOAD in batch mode, use direct commands.

When you use a direct command, observe the following rules:

- The keyword NATLOAD can be placed in a line by itself.
- To separate the individual parameters of a direct command from one another, use either blanks or the input delimiter character (as defined by the session parameter ID; the default is comma).
- If the string of parameters is longer than a single line, enter the character defined with the session parameter CF (default is %) at the end of every line that belongs to the command. This indicates continuation on the next line.  
However, this is only possible if you specify the command NATLOAD in a line by itself. That is, you cannot use CF, if you enter NATLOAD in the same line where a multi-line command starts.

The syntax of the NATLOAD direct commands for loading/scanning and Examples of Direct Commands for NATLOAD in Batch Mode are shown in the section NATUNLD/NATLOAD Direct Commands.

Below is information on:

- Condition Codes and User Exits in Batch Mode

## Condition Codes and User Exits in Batch Mode

Below is a description of NATLOAD condition codes and user exits available in batch mode.

The sources of NATLOAD user exits are named L-S-EX $nn$ , where  $nn$  denotes the number of the user exit. The user exits are delivered in the Natural system library SYSUNLD.

To make a user exit available, catalog it under the name LOADEX $nn$ , either in the Natural system library SYSUNLD or in one of its steplib.

The name of each user exit source is different from the name of the corresponding cataloged object. This guarantees that the object is not affected if the user exit source is overwritten by an installation update.

NATLOAD processing in batch mode terminates with one of the following condition codes:

Code	Explanation
0	<p>Loading executed successfully.</p> <p>If one or more objects were rejected during loading, the user exit LOADEX03 (source code L-S-EX03 in the library SYSUNLD) is invoked, if available. An object can, for example, be rejected if it has not been replaced.</p> <p>In the user exit LOADEX03, you can set a condition code.</p>
33	<p>An object to be replaced could not be deleted from the buffer pool:</p> <p>If "Replace=Y" was specified, an object to be replaced will, if necessary, also be deleted from the buffer pool. If the object cannot be deleted from the buffer pool, the new object will be loaded anyway and NATLOAD processing will continue.</p>
35	Duplicate subroutine names or class names/GUIDs were found (see also Class Names and GUIDs in the section "Verifying Subroutines and Classes").
37	Objects requested for loading could not be found on work file.
38	<p>Natural Security error occurred.</p> <p>With Condition Code 38, the user exit LOADEX02 (source code L-S-EX02) will be invoked. It allows you to stop processing when a Natural Security error occurs during the LOAD function.</p>
40	An error occurred; loading terminated.

With any condition code except 0 and 38, the user exit LOADEX01 will be invoked, if available. The source code of that user exit is provided under the name L-S-EX01 in the library SYSUNLD.

## NATLOAD Report

The NATLOAD utility generates (online and in batch mode) a report containing information on the objects loaded.

When you leave NATLOAD after the report has been displayed, a statistical report will be displayed, listing the number of objects processed.

After that, another screen will be displayed, showing the parameters used for processing.

See also NATUNLD/NATLOAD Commands for Reports in the section NATUNLD/NATLOAD Direct Commands.

# NATUNLD/NATLOAD - Direct Commands

The direct commands are available in batch mode and online. When you enter a direct command online, the entire command must fit into one command input line.

This section covers the following topics:

- NATUNLD Direct Command Syntax
- NATLOAD Direct Command Syntax
- NATUNLD/NATLOAD Commands for Reports

See also Executing NATUNLD in Batch Mode and Executing NATLOAD in Batch Mode in the sections NATUNLD Utility and NATLOAD Utility.

# NATUNLD Direct Command Syntax

This section covers the following topics:

- General Information
  - NATUNLD Syntax Diagrams
  - Examples of Direct Commands for NATUNLD in Batch Mode
- 

## General Information

After the execution of the direct command, you can enter another NATUNLD direct command, or END to leave NATUNLD, on the Input Command Data screen.

The individual items within each "with-...-clause" and "where-...-clause" can be specified in any sequence.

If you specify the parameter PC, specify it only once (as it applies to the entire work file); you have to specify it in the first possible "with-...-clause".

For error messages, the parameters FROM LIBRARY and TO LIBRARY are only valid for user-defined messages, not for Natural system messages.

The values following the keyword DIC in the "object-with-clause" and "ddm-with-clause" must be separated by commas. The commas must also be specified for omitted values.

For example: DIC (10,,SECRET,12)

If the session parameter ID has been set to comma, use slashes (/) instead of commas as separators.

For better readability, synonymous keywords are omitted from the syntax diagrams below. An underlined portion of a keyword represents an acceptable abbreviation.

Valid synonyms are:

Keyword	Synonym
DDM	<u>V</u> IEW
END	STOP
	QUIT
	FIN
	.
FROM	FM
PASSWORD	PSW
UNLOAD	UNLD

## NATUNLD Syntax Diagrams

```

NATUNLD {
  unload-objects-clause
  unload-ddms-clause
  unload-messages-clause
  delete-objects-clause
  delete-ddms-clause
  delete-messages-clause } ...
END
    
```

Below is information on:

- unload-objects-clause
- unload-ddms-clause
- unload-messages-clause
- object-where-clause
- object-with-clause
- ddm-where-clause
- ddm-with-clause
- message-with-clause
- delete-objects-clause
- delete-ddms-clause
- delete-messages-clause

### unload-objects-clause - for Programming Objects

```

[ UNLOAD ] {
  ALL
  SAVED
  CATALOGED } object-name [ FROM [ LIBRARY ] library-name ]
  BOTH
  STOWED } [ object-where-clause ]
  [ TO [ LIBRARY ] library-name ]
  [ object-with-clause ]
    
```

### unload-ddms-clause - for DDMs

```

[ UNLOAD ] DDM ddm-name [ ddm-where-clause ]
  [ ddm-with-clause ]
    
```

### unload-messages-clause - for Error Messages

[ UNLOAD ] <b>ERROR</b>	}	<b>USER</b> <b>NATURAL</b>	}	<b>BOTH</b> <b>SHORT</b> <b>LONG</b> <b>HELP</b>	}	<i>start-error-number</i>
						[ <b>TOERR</b> <i>end-error-number</i> ] [ <b>FROM</b> [ <b>LIBRARY</b> ] <i>library-name</i> ] [ <i>object-where-clause</i> ] [ <b>TO</b> [ <b>LIBRARY</b> ] <i>library-name</i> ] [ <i>message-with-clause</i> ]

### object-where-clause - for Programming Objects and Error Messages

[ <b>WHERE</b> ] [ <b>DBID</b> <i>dbid</i> ] [ <b>FNR</b> <i>fnr</i> ] [ <b>NAME</b> <i>vsam-name</i> ] [ <b>CIPHER</b> <i>cipher-key</i> ] [ <b>PASSWORD</b> <i>password</i> ]
--

### object-with-clause - for Programming Objects

[ <b>WITH</b> ] [ <b>DIC</b> ( <i>dbid,fnr,password,cipher-key</i> ) ] [ <b>FMDATE</b> <i>date</i> ] [ <b>TODATE</b> <i>date</i> ] [ <b>FMTIME</b> <i>time</i> ] [ <b>TOTIME</b> <i>time</i> ] [ <b>SETNO</b> <i>set-number</i> ] [ <b>SETUSER</b> <i>set-user-id</i> ] [ <b>NOSYMBOL-TABLE</b> ] [ <b>TID</b> <i>terminal-id</i> ] [ <b>TYPE</b> <i>object-types</i> ] [ <b>USER-ID</b> <i>user-id</i> ] [ <b>XREF</b> <i>xref-setting</i> ] [ <b>PC</b> ] [ <b>FIXEDLENGTH</b> ]
---

### ddm-where-clause

[ <b>WHERE</b> ] [ <b>DDMDBID</b> <i>dbid</i> ] [ <b>DDMFNR</b> <i>fnr</i> ]
--

### ddm-with-clause

[ <b>WITH</b> ] [ <b>DIC</b> ( <i>dbid,fnr,password,cipher-key</i> ) ] [ <b>PC</b> ] [ <b>FIXEDLENGTH</b> ]
--

### message-with-clause

```
[ WITH ] [ LANGUAGE language-codes ] [ PC ]
[ FIXEDLENGTH ]
```

### delete-objects-clause - Delete Instructions for Programming Objects

```
DELETE { ALL
        SAVED
        CATALOGED } object-name FROM [ LIBRARY ] library-name
        [ WITH ] [ PC ]
```

### delete-ddms-clause - Delete Instructions for DDMs

```
DELETE DDM dmm-name [ PC ]
```

### delete-messages-clause - Delete Instructions for Error Messages

```
DELETE ERROR { USER
              NATURAL } { BOTH
                        SHORT
                        LONG } start-error-number
                        [ TOERR end-error-number ]
                        [ FROM [ LIBRARY ] library-name ]
                        [ message-with-clause ]
```

## Examples of Direct Commands for NATUNLD in Batch Mode

In this example, all saved and/or cataloged programming objects of type program whose names start with PGM are unloaded from the library TESTLIB.

```
NATUNLD ALL PGM* FROM TESTLIB WITH TYPE P
END
```

In this example, NATUNLD is placed in a separate line, and commas instead of blanks are used as separator characters. Apart from that, the example is identical to the previous one.

```
NATUNLD
ALL,PGM*,FROM,TESTLIB,WITH,TYPE,P
END
```

In this example, all programming objects saved on or after the 1st May 2000 are unloaded from the library OLDLIB, which is located on Database 100 in File Number 160. The objects are to be loaded into the library NEWLIB.

This example also shows the use of the continuation indicator (%) for a parameter string that is longer than a single line.

```
NATUNLD
SAVED, *, FROM, OLDLIB, WHERE, DBID, 100, FNR, 160, %
FMDATE, 00-05-01, TO, NEWLIB
END
```

In this example, all programming objects are unloaded from the library OLDLIB - together with their corresponding cross-reference data which are located on an FDIC file with File Number 20 on Database 10. The objects are to be loaded into the library NEWLIB.

```
NATUNLD
ALL, *, FROM, OLDLIB, DIC, (10/20//), TO, NEWLIB, %
WITH, XREF, Y
END
```

This example shows the specification of multiple unloading instructions in one command.

```
NATUNLD
ALL, PROG1, FROM, TESTORD, DBID, 1, FNR, 6, PSW, PASSWD, TO, ORDERSMAIN
CATALOGED, PGM*, FROM, TESTLIB, TO, PRODLIB, WITH, TYPE, PNS
SAVED, PROG1, TO, NEWLIB
END
```

# NATLOAD Direct Command Syntax

This section covers the following topics:

- General Information
  - NATLOAD Syntax Diagrams
  - Examples of Direct Commands for NATLOAD in Batch Mode
- 

## General Information

The individual items within a "with-...-clause" or "where-...-clause" can be specified in any sequence.

The values following the keyword DIC in the "object-with-clause" and "ddm-with-clause" must be separated by commas. The commas must also be specified for omitted values; for example: DIC (10,,SECRET,12). If the session parameter ID has been set to comma, use slashes (/) instead of commas as separators.

For error messages, the parameters FROM LIBRARY and NEWLIBRARY are only valid for user-defined messages, but not for Natural system messages.

For error messages, the default range of numbers is 1 to 9999.

For better readability, synonymous keywords are omitted from the syntax diagrams below. An underlined portion of a keyword represents an acceptable abbreviation.

Valid synonyms are:

Keyword	Synonym
DDM	<u>V</u> IEW
END	STOP
	QUIT
	FIN
	.
FROM	FM
PASSWORD	PSW

## NATLOAD Syntax Diagrams

Below is information on:

- Loading/Scanning the Entire Work File
- Loading/Scanning Programming Objects only
- Loading/Scanning DDMs only
- Loading/Scanning Error Messages only
- Loading/Scanning Programming Objects, DDMs and Error Messages
- object-where-clause
- object-with-clause
- ddm-where-clause
- ddm-with-clause

- message-where-clause
- message-with-clause

### Loading/Scanning the Entire Work File

```
NATLOAD { LOAD
          SCAN } [ REPLACE ] [ DELETE-ALLOWED ] [ PC ]
```

### Loading/Scanning Programming Objects only

```
NATLOAD [ LOAD
          SCAN ] [ EXCEPT ] { ALL
                              SAVED
                              CATALOGED }
      [ object-name [ FROM [ LIBRARY ] library-name ]
      [ object-where-clause ] [ object-with-clause ] ]
```

### Loading/Scanning DDMs only

```
NATLOAD [ LOAD
          SCAN ] [ EXCEPT ] DDM
      [ ddm-name
      [ ddm-where-clause ] [ ddm-with-clause ] ]
```

### Loading/Scanning Error Messages only

```
NATLOAD [ LOAD
          SCAN ] [ EXCEPT ] ERROR { ALL
                                    USER
                                    NATURAL } [ BOTH
                                                SHORT
                                                LONG
                                                HELP ]
      [ start-error-number ] [ TOERR end-error-number ]
      [ FROM [ LIBRARY ] library-name ]
      [ message-where-clause ] [ message-with-clause ]
```

### Loading/Scanning Programming Objects, DDMs and Error Messages

<p><b>NATLOAD</b></p> <p>[ <b>LOAD</b> ] [ <b>SCAN</b> ]</p> <p>[ <b>EXCEPT</b> ]</p> <p>{ <b>ALL</b> <b>SAVED</b> <b>CATALOGED</b> }</p> <p><i>object-name</i></p> <p>[ <b>FROM</b> [ <b>LIBRARY</b> ] <i>library-name</i> ]</p> <p>[ <i>object-where-clause</i> ] [ <i>object-with-clause</i> ]</p> <p>[ <b>DDM</b> <i>dmm-name</i> [ <b>WHERE</b> ] [ <b>DDMDBID</b> <i>dbid</i> ] [ <b>DDMFNR</b> <i>fnr</i> ]</p> <p>[ <b>ERROR</b> { <b>ALL</b> <b>USER</b> <b>NATURAL</b> } { <b>BOTH</b> <b>SHORT</b> <b>LONG</b> <b>HELP</b> }</p> <p>[ <i>start-error-number</i> ] [ <b>TOERR</b> <i>end-error-number</i> ]</p> <p>[ <b>WHERE</b> ] [ <b>LANGUAGE</b> <i>language-codes</i> ]</p>
---

### object-where-clause

<p>[ <b>WHERE</b> ] [ <b>DBID</b> <i>dbid</i> ] [ <b>FNR</b> <i>fnr</i> ] [ <b>NAME</b> <i>vsam-name</i> ] [ <b>CIPHER</b> <i>cipher-key</i> ]</p> <p>[ <b>PASSWORD</b> <i>password</i> ] [ <b>NEWLIBRARY</b> <i>library-name</i> ]</p>
---

### object-with-clause

<p>[ <b>WITH</b> ]</p> <p>[ <b>DELETE-ALLOWED</b> ] [ <b>DIC</b> ( <i>dbid, fnr, password, cipher-key</i> ) ] [ <b>NUMBER</b> <i>number</i> ]</p> <p>[ <b>FMDATE</b> <i>date</i> ] [ <b>TODATE</b> <i>date</i> ] [ <b>FMTIME</b> <i>time</i> ] [ <b>TOTIME</b> <i>time</i> ]</p> <p>[ <b>TID</b> <i>terminal-id</i> ] [ <b>TYPE</b> <i>object-types</i> ] [ <b>USER-ID</b> <i>user-id</i> ] [ <b>XREF</b> <i>xref-setting</i> ] [ <b>PC</b> ]</p> <p>[ <b>REPLACE</b> ] [ <b>CHECK-DATE</b> ]</p> <p>[ <b>VERSIONCHECK</b> ]</p>
--

### dmm-where-clause

<p>[ <b>WHERE</b> ] [ <b>DDMDBID</b> <i>dbid</i> ] [ <b>DDMFNR</b> <i>fnr</i> ] [ <b>DBID</b> <i>dbid</i> ] [ <b>FNR</b> <i>fnr</i> ]</p>
---

**ddm-with-clause**

```
[ WITH ] [ DELETE-ALLOWED ]
[ DIC ( dbid,fnr,password,cipher-key ) ]
[ NUMBER number ] [ PC ] [ REPLACE ]
```

**message-where-clause**

```
[ WHERE ]
[ DBID dbid ] [ FNR fnr ] [ NAME vsam-name ] [ CIPHER cipher-key ]
[ PASSWORD password ] [ LANGUAGE language-codes ]
[ NEWLIBRARY library-name ]
```

**message-with-clause**

```
[ WITH ]
[ DELETE-ALLOWED ] [ PC ] [ REPLACE ]
```

**Examples of Direct Commands for NATLOAD in Batch Mode**

In this example, all programming objects with target library LIB1 are loaded from the work file.

```
NATLOAD ALL * FM LIB LIB1
```

In this example, NATLOAD is placed in a separate line, and commas instead of blanks are used as separator characters. Apart from that, the example is identical to the previous one.

```
NATLOAD
ALL,*,FROM,LIB,LIB1
```

In this example, all programming objects whose names begin with MAINEX and with target library SYSEXT are loaded from the work file and into File Number 10 on Database 32. This example also shows the use of the continuation indicator (%) for a parameter string that is longer than a single line.

```
NATLOAD
ALL,MAINEX*,FROM,LIBRARY,SYSEXT,WHERE%
FNR,10,DBID,32
```

In this example, all programming objects whose names begin with A and all DDMs are loaded from the work file. Older programming objects/DDMs of the same names will be replaced in the target system files.

```
NATLOAD LOAD ALL A* REPLACE CHECK-DATE DDM *
```

In this example, all DDMs on the work file are scanned.

```
NATLOAD SCAN DDM
```

# NATUNLD/NATLOAD Commands for Reports

In online mode, the utilities NATUNLD and NATLOAD provide the following commands for displaying the reports of the objects processed:

<b>Online Command</b>	<b>Function</b>
<u>B</u> ATCHREPORT	Displays reports in batch-mode.
<u>N</u> OBATCHREPORT	Displays reports in online-mode.
<u>N</u> OREPORT	Deactivates the display mode.
<u>R</u> EPORT	Reactivates the display mode.

# Object Handler - Overview

The Natural Object Handler is designed to process Natural and non-Natural objects for distribution in Natural environments. This is done by unloading the objects in the source environment into work files and loading them from work files into the target environment.

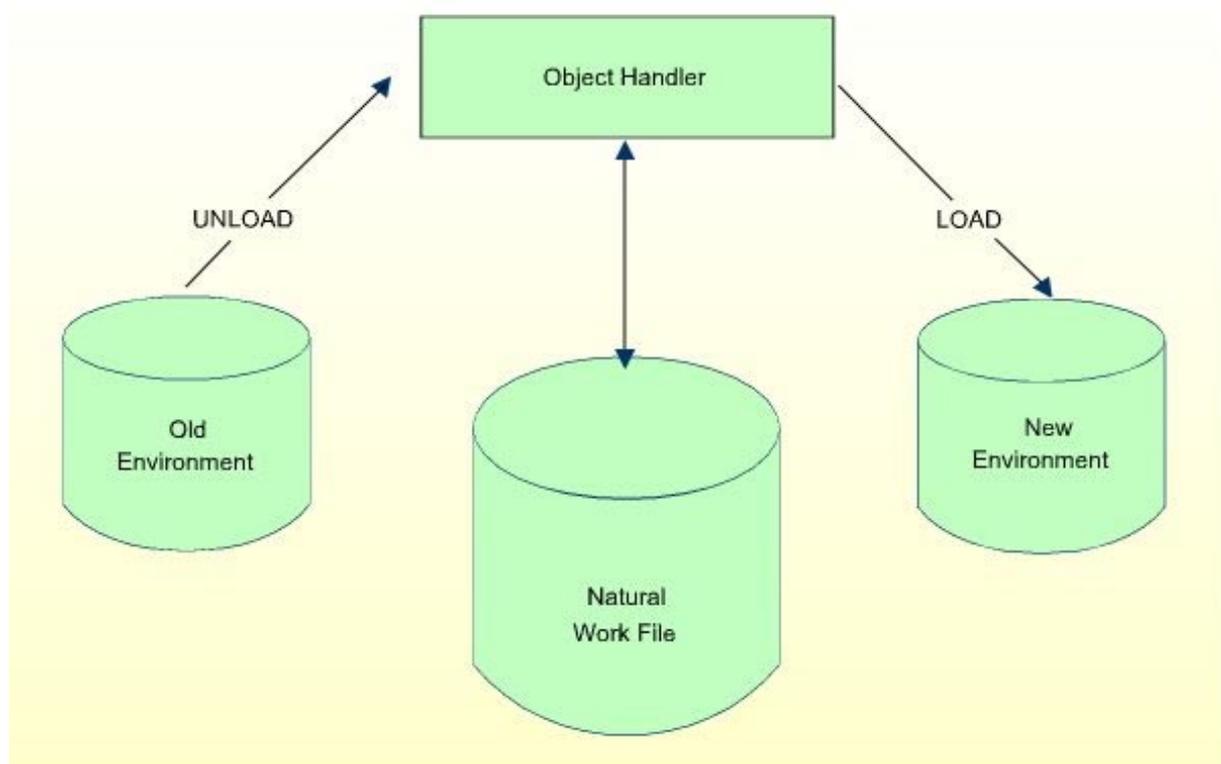
The Object Handler documentation covers the following topics:

●	General Information	Invoking the Object Handler, using batch or online mode and applying Natural Security.
●	Functions	Using the functions provided for processing objects: unload, load, restart load, scan, view, find and administration.
●	Object Specification	Specifying objects to be processed.
●	Settings	Specifying option and parameter settings for Object handler functions.
●	Workplans	Creating standard procedures for executing Object Handler functions.
●	Name, Date and Time Specification	Specifying names, dates, times and ranges.
●	Work Files	Work files used by the Object Handler.
●	Commands	Direct commands and syntax for executing functions and Workplans of the type PROCEDURE.
●	Batch Condition Codes and User Exit Routines	Condition codes and user exit routines provided in batch mode.
●	Tools	Displaying status information and setting trace and report options.
●	SYSOBJH Profile	Setting up a profile for defining individual defaults and standard procedures.

# Object Handler - General Information

The Object Handler consists of the utility SYSOBJH which is located in the library SYSOBJH, and the direct command interface. Additionally, the programming interface OBJHAPI is provided for executing Object Handler functions from a Natural program.

The following graphic shows how the Object Handler works when transferring objects from one platform to another:



This section covers the following topics:

- Principles of Object Transfer
- Invoking the Object Handler
- Batch or Direct Command Calls
- Issuing Object Handler Commands from a Natural Program
- Text Members for Reports, Restarts and Traces
- Natural Security
- Standard PF Keys

---

## Principles of Object Transfer

To process Natural and non-Natural objects, the Object Handler unloads objects from the source environment into work files and loads them from work files into the target environment.

Depending on the source and target environment, you can transfer data of binary or text format. Binary format can be used for Natural sources and cataloged objects, error messages, Natural command processor sources, Natural DDMs (data definition modules), Natural-related objects and Adabas FDTs (Field Definition Tables).

Text format applies to Natural source objects, command processor sources, error messages, DDMs and Adabas FDTs. Between mainframe and UNIX/Windows platforms, you can only transfer text data. Between identical platforms, you can transfer binary data.

The format of the work file to be transferred depends on the mode set for transferring binary or text data. For further details on the transfer modes provided by the Object Handler, see Work File Format in the section Work Files.

## Invoking the Object Handler

There are two ways of invoking the Object Handler for interactive usage:

### To invoke the Object Handler online from any Natural library

- Enter the system command SYSOBJH.  
The Natural Object Handler Main Menu is displayed.

The Natural Object Handler Main Menu offers the following functions:

- Unload
- Load
- Scan
- View
- Administration

### To select a menu item

- Enter any character in the input field next to the option that corresponds to the function desired.  
Or choose the PF key that corresponds to the function desired.  
Or, in the command line, enter a corresponding Object Handler command (see Commands).

See the section Functions for a description of the functions provided, and how to process the functions in advanced-user mode or by using wizards.

### To invoke the Object Handler in batch or direct command online mode

- Enter the system command SYSOBJH followed by a direct command (see also Batch or Direct Command Calls below and the section Commands).

After execution of a direct command, you can enter either another direct command or a period (.) to exit the Object Handler.

## Batch or Direct Command Calls

Several commands can be issued to the Object Handler. The last command in the command sequence must be a period (.), STOP, END, QUIT or FIN, where FIN ends the Natural session.

### Batch Mode

The commands to the Object Handler are read from standard input. Each command can be separated into a maximum of 20 command parts/strings by entering input delimiters (session parameter ID) after any keyword or keyword value. Each command part/string must not exceed 248 bytes.

If the command is longer than a single line, at the end of every line except the last that belongs to the command, enter the character defined with the session parameter CF (default is %) This indicates continuation on the next line. However, this is only possible if you specify the command SYSOBJH in a line by itself. That is, you cannot use CF, if you enter SYSOBJH in the same line where a multi-line command starts.

In addition, we recommend that you set the LS profile parameter to 250.

**Example (assuming ID is set to ","):**

```
SYSOBJH
UNLOAD PROG* LIBRARY MYLIB1, OBJTYPE N,%
WITH NEWLIBRARY MYLIB2%
WHERE REPORT TRANSFER
STOP
```

**Related Topics in the Section Object Handler - Commands:**

- Direct and PROCEDURE Workplan Commands
- Direct and PROCEDURE Workplan Syntax

## Online Mode

The command to the Object Handler in the Natural command line can consist of up to 20 command parts.

**Example:**

```
SYSOBJH UNLOAD * LIB EXAMPLE WHERE TRANSFER
```

# Issuing Object Handler Commands from a Natural Program

You can issue commands to the Object Handler with a Natural program by using the OBJHAPI programming interface. For the parameters required and examples, see the Natural program DOC-API supplied in the Natural system library SYSOBJH.

## Text Member for Reports, Restarts and Traces

Report, restart and trace data created by the Object Handler are stored as Natural text members in the Workplan library. The Object Handler generates names for text members that have not been explicitly specified in the Options window. The names generated are a combination of the weekday and the time.

For example: a member with the name 21415568 was created on Tuesday (the second day of the week) at 14:15:56,8.

## Natural Security

If Natural Security is installed, the Object Handler checks the SYSOBJH utility profile in Natural Security to find out whether the requested function is allowed.

Should a Natural Security error occur during the load function, the following applies:

- If the option Write Report is set, the error message is written into the report file and loading continues.
- If the option Write Report is not set, an error message is issued and the load function is terminated.

## Standard PF Keys

The following PF keys are available on all full-screen maps:

PF Key	Explanation
PF1	Invokes the Help function for the field at which the cursor is positioned.
PF1	Invokes the Help function for the field at which the cursor is positioned.
PF3	Exits the current screen and returns to the previous screen.
PF6	Goes to the top of a list.
PF7	Returns to the previous page. On wizard screens: goes back one screen/step.
PF8	Goes to the next page. On wizard screens: goes to the next screen/step.
PF9	Goes to the bottom of a list.
PF10	Invokes the Commands menu to select commands for navigation purpose and to specify special settings. See also Commands for CUIs in the section Commands.
PF12	Cancel the current function.
PF20	Lists all active programs of the Object Handler (for support purposes).

# Object Handler - Functions - Overview

This section describes the main functions provided by the Object Handler. To guide you through the steps required to execute the unload, load and scan functions, you can take advantage of the Object Handler wizards. The wizards are activated by default. If you prefer the unload, load or scan mode for the experienced user instead, on the Natural Object Handler Main Menu, mark the field next to "Advanced user" or, in the SYSOBJH profile, set Advanced-Mode to **Y** (Yes). See also SYSOBJH Profile.

To automate function processing, for the unload, load or scan function, the Object Handler provides the option to create standard procedures to define recurring settings and object specifications. See the section Workplans for further information.

This section covers the following topics:

- Wizards
- Advanced User
- Restart Load  
(section: Settings)
- View
- Find
- Administration
- Change Workplan Library  
(section: Administration)
- System File Selection
- Library Selection
- Workplan Selection  
(see List in the section Administration)
- System Error Message Selection
- Object Selection

# Object Handler Wizards

The Object Handler provides wizards that determine the processing sequence for

- Unloading data from the Natural system environment into Natural work files.
- Loading data from work files into the Natural system environment.
- Scanning the contents of Natural work files.

## To activate the wizards

- On the Natural Object Handler Main Menu, select the "Advanced user" field if required (the field is not selected by default).

The wizards provide the keys PF8 and PF7 to navigate between the screens (steps). Use PF12 to cancel the processing sequence.

The steps below describe the processing sequence performed with the unload, load or scan wizard:

- Step 1 - Start the Procedure
  - Step 2 - Unload/Load/Scan Object into/from Work Files
  - Step 3 - Set Parameters
  - Step 4 - Select Objects
  - Step 5 - Execute Processing
  - Step 6 - Continue Processing
- 

## Step 1 - Start the Procedure

### To start the unload, load or scan procedure

1. From the Natural Object Handler Main Menu, choose Unload, Load or Scan by entering any single character next to the function desired or by using the corresponding PF key.  
The Unload/Load/Scan Wizard main screen appears with the following options:
  1. Unload/Load/Scan objects into/from Natural work file(s)
  2. Start Object Handler command procedure
2. Choose any option:
  - If you choose "Unload/Load/Scan objects into/from Natural work file(s)", proceed with Step 2 below.
  - If you choose "Start Object Handler command procedure", proceed as follows:
    1. On the Unload/Load/Scan Wizard screen, choose "Start Object Handler command procedure".  
The Procedure screen appears.
    2. In the Name field, enter the name of a Workplan of the type PROCEDURE by using either option below:
      - Type in the name of a Workplan of the type PROCEDURE (see also Workplans) that should be used for the transaction.
      - Choose Select Workplan or choose PF5 to display a list of available Workplans of the type PROCEDURE.  
In the line next to the Workplan you want to select, enter the command **S** or **SE**.  
Press ENTER to execute the command and fill the Name field on the Procedure screen.
    3. Select List Workplan or choose PF4 if you want to display the Workplan specified.

See also List Workplan in the section Administration.

3. Press ENTER to continue.
4. Proceed with Step 5 and Step 6 below.

## Step 2 - Unload/Load/Scan Objects into/from Work Files

### To unload, load or scan objects into/from Natural work files

1. On the Unload/Load/Scan Wizard screen, choose "Unload/Load/Scan objects into/from Natural work file(s)".
2. Press ENTER or choose PF8/Next to continue.  
The Options screen of the wizard appears with the following fields, commands and alternative PF keys:

Field	PF Key	Explanation
Transfer format		<p>Only valid if "Use default options" (this is the default) has been selected.</p> <p>If selected, the data to be processed is written in Transfer format into/from the work file. See also Work File Format in the section Work Files.</p> <p>Unload function: The data to be unloaded is written in Transfer format into the work file. Note that if you want to change the setting of this field for a subsequent unload, you need to return to the Natural Object Handler Main Menu or enter the command GO UNLOAD END (see Commands for CUIs) and restart the unload function.</p> <p>Load and scan functions: The data to be loaded or scanned are expected to be in Transfer format.</p>
Use PC work file		<p>Only applies if Entire Connection is installed.</p> <p>If selected, the data to be processed is written into an Entire Connection work file.</p>
PC File		<p>Only valid if "Use PC work file" has been selected.</p> <p>The name of the path and the Entire Connection work file to be used.</p>
Use default options		<p>Default options are (this is the default). For the options available, see SYSOBJH Profile and Set Additional Options in the section Settings.</p>
Set additional options	PF4	<p>Only valid if "Use default options" has been selected.</p> <p>Invokes the Options screen of the wizard where you can modify the default settings and enter additional options for the processing sequence. See Set Additional Options in the section Settings.</p>
Use Option Workplan		<p>If selected, a Workplan of the type OPTION is used (see Workplans).</p>
Name		<p>Only valid if Use Option Workplan has been selected.</p> <p>The name of a Workplan of the type OPTION to be used.</p>
List Option Workplan	PF6	<p>Only valid if Use Option Workplan has been selected.</p> <p>Displays the contents of the Workplan specified in the field Name.</p>
Select Option Workplan	PF5	<p>Only valid if Use Option Workplan has been selected.</p> <p>Displays a selection list of available Workplans of the type OPTION (see also List Workplan in the section Administration).</p>

3. Select any of the options provided and (if necessary) complete the fields to be used for the processing sequence.
4. Press ENTER or choose PF8/Next to continue.  
The Parameters screen of the wizard appears.

## Step 3 - Set Parameters

### ▶ To set parameters for the processing procedure

1. On the Parameters screen, select any of the following options listed below and (if necessary) complete the fields to be used for the processing sequence:

Field	PF Key	Explanation
Do not use parameters		If selected (default), no parameters are set.
Use global parameters		If selected, global parameters are used. See Set Global Parameters in the section Settings.
Set global parameters	PF4	Only valid if "Use global parameters" has been selected.  If selected, the Parameters screen is invoked. See Set Global Parameters (Settings) and Parameter-Setting (Direct and PROCEDURE Workplan Syntax, Commands) for a description of keywords and valid values.
Use Parameter Workplan		If selected, a Workplan of the type PARAMETER is used (see Workplans).
Name		Only valid if Use Parameter Workplan has been selected.  The name of a Workplan of the type PARAMETER to be used.
List Parameter Workplan	PF6	Only valid if Use Parameter Workplan has been selected.  If selected, the contents of the Workplan specified in the field Name is displayed.
Select Parameter Workplan	PF5	Only valid if the field Use Parameter Workplan has been selected.  If selected, a selection list of available Workplans of the type PARAMETER is displayed (see List Workplan in the section Administration).

2. Press ENTER or PF8/Next to continue.  
The Select Unload/Load/Scan Type screen appears.

## Step 4 - Select Objects

### ▶ To select the type of object you want to process

- On the Select Unload/Load/Scan Type screen, use any of the three options below. Note that the first option only applies to the load and scan functions. For the keywords and valid values that apply to each object type, see the relevant explanation in the section Object Specification.
  1. Select "Load/Scan all objects" to process all objects from the work file.
  2. Select a particular type of object:
    - Natural library objects
    - Natural system error messages
    - Natural command processor sources
    - Natural-related objects
    - DDMs
    - FDTs

Press ENTER or choose PF8/Next to continue.

Depending on the type of object selected, a screen appears where you can specify selection criteria for the objects to be processed.

Specify the objects and choose Details (if available) for further object specification, if required. For information on Details, see the relevant explanation in the section Object Specification.

3. Select "Use Selection or List Workplan" to use a Workplan of the type SELECTION or LIST. See also Workplans.

Press ENTER or choose PF8/Next to continue.

The Selection or List screen appears.

In the Name field, enter the name of a Workplan of the type SELECTION or LIST by using either option:

- Type in the name of a Workplan.
- Or, choose Select Workplan or PF5/SelWP to display a list of all Workplans available.  
In the line next to the Workplan you want to select, enter the command S or SE. Press ENTER to execute the command and fill the Name field on the Selection or List screen. See also List Workplan in the section Administration.

Choose List Workplan or PF4/Li-WP if you want to display the contents of the Workplan entered in the Name field.

- Press ENTER or choose PF8/Next to continue.  
The wizard displays the processing command generated from the input data.  
You can save the command displayed as a Workplan of the type PROCEDURE (see also Workplans), by entering the command SAVE or by choosing PF5/Save.

## Step 5 - Execute Processing

### To execute the processing procedure

1. On the command execution screen, press ENTER or choose PF8/Next to confirm the settings and to process the objects specified.  
If required, choose PF7/Back and modify the processing settings before you confirm the command execution.  
The Object Handler performs the function and displays a confirmation message.
2. Press ENTER to continue.  
A report screen appears with a list of the objects processed.
3. Choose PF3/Exit again to leave the report screen or choose PF12/Canc to terminate the function.  
A window appears where you choose whether to continue processing data.
4. Choose No and press ENTER to terminate the function.  
Or choose PF12 to terminate the function.

The Natural Object Handler Main Menu appears.

## Step 6 - Continue Processing

### To continue processing

1. On the report screen, choose PF3/Exit.  
A window appears where you choose whether to proceed with the next processing step.
2. Choose Yes.

A screen appears with the option to reuse or change previous settings.

# Object Handler - Advanced User

This section describes how to invoke advanced-user mode and perform the unload, load and scan functions.

- Activating Advanced User
  - Processing Objects
- 

## Activating Advanced User

### ▶ To activate advanced-user mode

- On the Natural Object Handler Main Menu, select the "Advanced user" field (the field is not selected by default).  
Or, in the SYSOBJH profile, set Advanced-Mode to **Y** (Yes). See also SYSOBJH Profile.

## Processing Objects

### ▶ To process objects in advanced-user mode

1. On the Natural Object Handler Main Menu, check the "Advanced user" field and select Unload, Load or Scan.
2. Press ENTER to continue.  
The Unload/Load/Scan Settings screen appears with the sections Options and Parameters.
3. Set the options and parameters as described in the section Settings.
4. Press ENTER to continue.  
The Select Unload/Load/Scan Type screen appears.
5. Select the objects you want to process: see also the section Object Specification.
6. Choose Details to specify additional selection criteria: see the relevant sections in Object Specification.
7. Press ENTER to continue.
  - If the parameter "Display-Cmd-in-Advanced-Mode" is set to N (No) in the SYSOBJH profile (this is the default), or if no such profile exists, the command generated from the input data is executed immediately after you have specified the selection data. See SYSOBJH profile.  
The Display Unload/Load/Scan Report screen appears with a list of the objects processed if the parameter Write Report was selected (this is the default). See also Work File Options in the section Settings.
  - If the parameter "Display-Cmd-in-Advanced-Mode" is set to Y (Yes) in the SYSOBJH profile (see the relevant section), or if the internal command SET ADVANCEDCMD ON (see Commands for CUIs) was executed earlier, a screen appears displaying the command generated from the input data.  
You can save the command displayed as a Workplan of the type PROCEDURE (see also Workplans), by entering the command SAVE or by choosing PF5/Save.  
Press ENTER to confirm the command execution or choose PF3/Exit to modify the processing settings before confirming the command execution.  
The Display Unload/Load/Scan Report screen appears with a list of the objects processed if the parameter Write Report was selected (this is the default). See also Work File Options in the section Settings.

# Object Handler - View

This function is used to display the objects currently located in your Natural system environment.

For information on the columns and fields that appear on the screens generated by the view function, refer to the section Object Specification.

## To invoke the view function

- On the Natural Object Handler Main Menu, choose View.  
Or, on any other Object Handler screen, enter the direct command GO VIEW (see also Commands for CUIs).  
The Select View Type screen appears with a list of the types of object available for selection:
  - Natural library objects
  - Natural system error messages
  - Natural command processor sources
  - FDTs
  - Natural-related objects
  - DDMs

The section below describes how to view the object types listed above. In addition, information is provided on the columns and commands available on the screens generated by the view function:

- Columns and Commands
- 

## Natural Library Objects

### To view Natural programming objects and user-defined error messages

1. On the Select View Type screen, select "Natural library objects".
2. Press ENTER.  
The View System Files screen appears with a list of all system files available.
3. In the Cmd column, enter any single character next to the system file you want to select (default is the current FUSER/FNAT): see also System File Selection.
4. Press ENTER.  
The View Libraries screen appears with a list of all libraries available.
5. In the Cmd column, next to the library you want to select, enter the line command SE or LI: see also Library Selection.
6. Press ENTER.  
The View Library Objects screen appears with a list of all objects available.
7. In the Cmd column, next to the object(s) you want to select, enter LI: see also Object Selection.
8. Press ENTER.  
The object(s) selected is displayed on the screen.

## Natural System Error Messages

### To view Natural system error messages

1. On the Select View Type screen, select "Natural system error messages".
2. Press ENTER.  
The View System Error Messages screen appears.

3. Specify a range of error numbers (for valid values, see Name in Name, Date and Time Specification), the database ID (DBID) and file number (FNR) of the FNAT system file where the system error messages are stored.  
See also: Natural System Error Message Selection.
4. Press ENTER.  
The Object Handler generates a list of the system error messages specified and displays them on the screen.

## Natural Command Processor Sources

### To view Natural command processor sources stored in an Adabas file

1. On the Select View Type screen, select "Natural command processor sources".
2. Press ENTER.  
The View Natural Command Processors screen appears.
3. Enter the database ID (DBID), the number of the Adabas file (FNR), the Adabas password and cipher code of the Adabas file where the Natural command processor libraries are stored. As default value, the current setting of the FUSER system file is used.
4. Press ENTER.  
The View Libraries screen appears with a list of all libraries where Natural command processor sources are stored.
5. In the Cmd column, next to the library you want to select, enter the line command SE or LI: see also Library Selection.
6. Press ENTER.  
The View Command Processor Sources screen appears with a list of all command processor sources available.

## FDTs

### To view the FDTs (Field Definition Tables) available in an Adabas database

- On the Select View Type screen, select FDTs.
- Press ENTER.  
The View FDTs screen appears.
- Enter the database ID (DBID) of the Adabas database where you want the FDTs to be displayed, and the range of file numbers (FNRs).
- Press ENTER.  
The "View FDTs for DBID" screen appears with a list of all FDTs available in this database.

## Natural-related Objects

### To view Natural-related objects

1. On the Select View Type screen, select "Natural-related objects".
2. Press ENTER.  
The Select Natural-related Type screen appears.
3. Select the Natural-related type you want to view.
4. Press ENTER.  
A screen appears where you can specify the database ID (DBID) and the number (FNR), password and cipher code of the Adabas file where the Natural-related objects are stored. As default value, the current setting of the FNAT system file (for profiles), the FUSER system file (for debug environments) or the FDIC system file (for DL/I subfiles) is used.
5. Press ENTER.

The View Profiles/Debug Environments/DL/I Subfiles screen appears.

- Specify the database ID (DBID) and the number (FNR), password and cipher code of the Adabas file where the Natural-related objects are stored.  
Or, press ENTER to list all Natural-related objects available.

A screen appears that lists all objects specified.

- In the Cmd column, next to the object(s) you want to select, enter LI: see also Object Selection.
- Press ENTER.  
The Natural-related object(s) selected is displayed on the screen.

## DDMs

### To view Natural DDM (data definition module) objects

- On the Select View Type screen, select DDMs.
- Press ENTER.  
A screen appears where you can specify the database ID (DBID) and the number (FNR), password and cipher code of the Adabas file where the DDMs are stored. As default value, the current setting of the FDIC system file is used.
- Press ENTER.  
The View DDMs screen appears that lists the specified DDMs in alphabetical order.
- In the Cmd column, next to the DDM(s) you want to select, enter LI: see also Object Selection.
- Press ENTER.  
The DDM(s) selected is displayed on the screen.

## Columns and Commands

Except for FDTs where no input is possible, in the Cmd column of the screens generated by the view function you can enter the following line commands:

Line Command	Explanation
DE	Deletes the list item(s) selected.
LI	Not applicable to Natural command processor sources.
L	Lists the source of the object selected.

# Object Handler - Find

This function is used to locate objects in your Natural environment and generate a list of the objects found.

 **To invoke the find function**

- On any Object Handler screen, in the command line, enter GO FIND.

For information on

- The columns that appear on the report screen generated by the find function, refer to the section Object Specification.
- The subcommands provided with GO FIND, refer to the relevant section in Object Handler Commands for CUIs.

# Object Handler - Administration

This function is used to maintain Object Handler Workplans.

For information on Workplans and the syntax that applies, refer to the sections Workplans and Commands.

**Note:**

You can set the default library for Workplans in the SYSOBJH profile by using the option Workplan-Library (see SYSOBJH Profile).

The Administration screen provides the following options:

- List the available Workplans in the Workplan library
  - Create a new Workplan
  - Change the Workplan library
- 

## List the available Workplans in the Workplan Library

This function is used to list all Natural objects of the type Text that are available in the Workplan library and to select all Workplans available.

The instructions below show how to invoke and execute the function:

### To list Workplans

- On the Administration screen, select "List the available Workplans in the Workplan library" and press ENTER, or choose the corresponding PF key.

The List Workplans screen appears with a list of all Workplans available.

If the Natural object of the type Text is a Workplan, the type of Workplan and the first 50 bytes of the Workplan description are listed. Choose PF5 to display additional information.

- Select a Workplan.

The section below provides information on:

- Columns and Commands

## Columns and Commands

The List Workplans screen provides the following columns and commands:

Column	PF Key	Explanation
Cmd		<p>The following line commands can be entered in the input field next to the Workplan desired:</p> <p>C        Only applies to Workplans of the types PROCEDURE, SELECTION, CH        PARAMETER and OPTION. Checks the syntax.</p> <p>DE       Deletes the Workplan.</p> <p>ED       Edits the Workplan.</p> <p>EX       Only applies to Workplans of the type PROCEDURE. Executes the Workplan.</p> <p>L        Lists the Workplan.</p> <p>LI</p> <p>S        Only applies to the Select Workplan function.</p> <p>SE       Selects the Workplan.</p>
Name		Lists all Natural objects of the type Text and provides the option to enter a range of object names in the input fields of the first row: see also (*) Name.
Type		<p>The type(s) of Workplan. Valid entries are:</p> <p>PROCEDURE        or P</p> <p>SELECTION         or S</p> <p>LIST                or L</p> <p>PARAMETER        or A</p> <p>OPTION             or O</p> <p>TEXT                or T</p> <p>*                    all types</p> <p>or any combination of the short types, for example, SL.</p>
Description		Description range of the Workplans listed: see (*) Name.
User ID		<p>Only displayed with PF5.</p> <p>The ID of the user who created the Workplan. Specify a single user ID or a range of user IDs: see also (*) Name.</p>
Date		<p>Only displayed with PF5.</p> <p>A date or a range of dates: see (*) Date.</p>
Time		<p>Only displayed with PF5.</p> <p>A time or a range of times: (*) Time.</p>
	PF4	Switches from the additional information display (PF5) to the standard display.
	PF5	Displays additional information: user ID, date and time (see the relevant explanation above).

**Note:**

\* The notation "see (\*)" in the table above regarding name, date and time parameters refers to the section Name, Date and Time Specification.

## Create a new Workplan

This function invokes the "Create a new Workplan" screen where you can specify the type of the new Workplan and choose the option Free Format Editing (see also the direct commands SET FREE ON and SET FREE OFF in Commands for CUIs).

With Free Format Editing set to OFF (not activated), for Workplans of the types OPTION, PARAMETER and SELECTION, screens with input fields are provided.

If the option Free Format Editing set to ON, or if you create a Workplan of another type, you will get a map with an edit area. Enter the contents of the Workplan; see the syntax as described in the section Commands.

## Change the Workplan Library

This function is used to change the Workplan library. All Workplans must be stored in a Workplan library.

The Change Workplan Library screen provides the following fields:

Library	The name of the Workplan library. Default is the library WORKPLAN.
Select library	Displays a list of all Workplan libraries available: see also Library Selection.
PF4/SeLib	
DBID/FNR	Specifies the database ID (DBID) and file number (FNR) where the Workplan library is located. If no values are specified, the current FUSER or FNAT system file is used.
Passw./Ciph.	The password and cipher code of the Adabas file where the Workplan library is located.

# Object Handler - System File Selection

This function is used to generate a selection screen with the system files available in your Natural environment.

The instructions below shows how to invoke and execute the function by using advanced-user mode (see the relevant section):

## To list and select system files

1. On the Unload Natural Library Objects screen, choose Select DBID/FNR and press ENTER, or choose PF5/DBIDs.  
The Select System File window appears with a list of the database IDs (DBID) and the file numbers (FNR) of the current FNAT and FUSER system files. Additionally, fields appear for input of user-specified values.
2. Select the system file you want to view by entering any single character next to the file desired in the input field of the Sel column.  
Or, in the input fields next to "User-defined", enter a DBID, FNR and an Adabas password/VSAM name and cipher code, if relevant.

# Object Handler - Library Selection

This function is used to list and select the Natural libraries available in the system file and specified by the database ID (DBID) and the file number (FNR).

The instructions below show how to invoke and execute the function by using advanced-user mode (see the relevant section):

## To list and select libraries

- On the Unload Natural Library Objects screen, choose "Select library" and press ENTER, or choose PF4/SeLib.  
The Select Library window appears with a list of all libraries available and the corresponding DBIDs and FNRs.
- Select a library or specify a range of libraries (see the table below).
- Press ENTER

The Select Library screen provides the following columns:

Select	Select a library by entering any single character in the input field next to the library desired.
Library	Lists all Natural libraries available and provides the option to enter a range of libraries in the input fields of the first row: see also Name in the section Name, Date and Time Specification.
DBID/FNR	<p>The DBID and FNR of the Natural system file where the libraries are stored.</p> <p>If no values (or 0) are specified, the current FUSER or FNAT system file is used. These fields cannot be changed if:</p> <ul style="list-style-type: none"> <li>• The view function (see the relevant section) is used,</li> <li>• A library is selected for the load or scan function,</li> <li>• Natural Library Objects Exceptions are specified (see the relevant section in Object Specification).</li> </ul>

# Natural System Error Message Selection

This function is used to list and select Natural system error messages.

The instructions below show how to invoke and execute the function by using advanced-user mode (see the relevant section):

## To list and select Natural System error messages

- On the Unload Natural System Error Messages screen, choose "Select system error messages" and press ENTER.  
The List System Error Messages screen appears with a list of all system error messages for the Natural system file specified.
- If you want to unload one or more objects, select a system error message or specify a range: see Columns and Commands below.
- Press ENTER.  
The object(s) has been added to the list of unload items.
- Choose PF2/Unloa to execute the unload function (see also Columns and Commands below).

## Columns and Commands

The List System Error Messages screen provides the following columns and commands:

Column	PF Key	Explanation
Cmd		<p>The following line commands can be entered in the input field next to the object desired:</p> <p><b>L or LI</b>                Lists the error message.</p> <p><b>S, SE, U or UL</b>        Only applies to the unload function. Selects the error message for subsequent unloading.</p> <p><b>DE</b>                        Deletes the error message.</p> <p><b>DL</b>                        Deletes the long text of the error message only.</p>
Number		<p>Lists all Natural system error messages available and provides the option to enter a range of error messages in the input fields of the first row. Valid ranges for selection are:</p> <p><i>value</i>*                All items with names that begin with value.</p> <p><i>value</i>&gt;                All items with names greater than or equal to value. Example: 10&gt;</p> <p><i>value</i>&lt;                All items with names less than or equal to value. Example: 100&lt;</p>

Column	PF Key	Explanation
S/L-Kind		The kind of error message text:  <b>S</b> Short texts of error messages.  <b>L</b> Long texts.  <b>A</b> All short and/or long texts.
Language		The language code(s) of the error messages to be selected: Up to 8 valid Natural language codes. Asterisk (*) selects all language codes.
Error Message Text		Displays the short text of the error message.
	PF2	Only applies to the unload function.  Starts unloading the object(s) selected for processing.  As an alternative, in the command line, enter either of the following direct commands:  UNLOAD  UNLD
	PF6	Goes to the top of the list.
	PF7	Returns to the previous page.
	PF8	Goes to the next page.
	PF9	Goes to bottom of the list.
	PF11	Selects all objects of the range(s) specified.  As an alternative, in the command line, enter the following direct command: <u>SELECT ALL.</u>

# Object Handler - Object Selection

This function is used to list and select the Natural objects or Natural command processor sources available in the library specified.

The instructions below show how to invoke and execute the function by using advanced-user mode (see the relevant section):

## To list and select Natural objects or Natural command processor sources

- On the Unload Natural Library Objects screen, in the field Library, enter the name of a library and choose Select Objects and press ENTER.  
The List Library Objects screen appears with a list of all objects available.
- If you want to unload one or more objects, select the object or specify a range of objects: see Columns and Commands below.
- Press ENTER.  
The object has been added to the list of unload items.
- Choose PF2/Unloa to execute the unload (see also Columns and Commands below).

## Columns and Commands

The List Library Objects screen provides the following columns and commands:

Column	PF Key	Explanation
Cmd		The following line commands can be entered in the input field next to the object desired:  <b>L or LI</b> Lists the object (not possible for Natural command processor sources).  <b>S, SE, U or</b> Only applies to the unload function. <b>UL</b> Selects the object for subsequent unloading.  <b>DE</b> Delete the object.
Name		Lists all Natural library objects available and provides the option to enter a range of library objects in the input fields of the first row: see also (*) Name.
Type		Up to 11 valid Natural object types (for example, <b>P</b> for program) or asterisk (*) to list all objects.
S/C		The SC-Kind of the Natural programming objects listed:  <b>*</b> All saved and/or cataloged objects. This is the default.  <b>S</b> Source objects only.  <b>C</b> Cataloged objects only.  <b>S/C</b> Both source and cataloged objects if both exist.  <b>W</b> Both source and cataloged objects if both exist with identical date and time.

Column	PF Key	Explanation
M		The programming mode of the objects:  * Any mode, structured and/or reporting. <b>S</b> Structured mode only. <b>R</b> Reporting mode only.
Version		The Natural version under which the objects were saved and/or cataloged. For valid ranges of versions, see (*) Name.
User ID		The ID of the user who saved or cataloged the object. Specify a single user ID or a range of user IDs: see also (*) Name.
Date		A date or a range of dates: see (*) Date.
Time		A time or a range of times: (*) Time.
	PF2	Only applies to the unload function.  Starts unloading the object(s) selected for processing.  As an alternative, in the command line, enter either of the following direct commands:  UNLOAD  UNLD
	PF6	Goes to the top of the list.
	PF7	Returns to the previous page.
	PF8	Goes to the next page.
	PF9	Goes to the bottom of the list.
	PF11	Only applies to the unload function.  Selects all objects of the range(s) specified.  As an alternative, in the command line, enter the following direct command: <u>SELECT ALL</u> .

\* The notation "see (\*)" in the table above regarding name, date and time parameters refers to the section Name, Date and Time Specification.

# Object Handler - Object Specification

The Object Handler provides the screen Select Unload/Load/Scan Type where you can select any of the following types of object for processing:

- All Objects on the Work File  
(load and scan only)
- Natural Library Objects
- Natural System Error Messages
- Natural Command Processor Sources
- Natural-related Objects
- DDMs
- FDTs
- Use Selection or List Workplan

For each type of object selected, you are provided object-specification screens. These screens are used to specify selection criteria for the objects to be processed. The field and options provided on these screen are explained in this section.

**Note:**

\* The notation "see (\*)" in the tables below regarding name, date and time parameters refers to the section Name, Date and Time Specification.

---

## All Objects on the Work File

This function is used to select all objects available in the work file for processing. From the Load/Scan All Objects screen, you can invoke the Settings screen where you can specify option and parameter settings. See the section Settings.

## Natural Library Objects

This function is used to select Natural library objects for processing. Natural library objects are programming objects and user-defined error messages.

For a description of keywords and valid values, see also Select-Clause in Direct and PROCEDURE Workplan Syntax in the section Commands.

The screen Unload/Load/Scan Natural Library Objects provides the following fields and PF keys:

Field	PF Key	Explanation
DBID/FNR		Only applies to the unload function.  The database ID (DBID) and file number (FNR) of the system file where the Natural libraries are stored. If no values (or <b>0</b> ) are specified, the current FUSER or FNAT system file is used.
Select DBID/FNR	PF5	Only applies to the unload function.  Displays a selection list of system files available.
Password/Cipher		Only applies to the unload function.  The password and cipher code of the Adabas file where the Natural libraries are stored.
Library	PF6	The name of a library or a range of names: see (*) Name.
Select library	PF4	Displays a selection list of all libraries available. See also the section Library Selection.
Object name		The name of a Natural programming object or a range of names: see (*) Name.  Only evaluated if the parameter Natural programming objects (default) is selected on the screen Natural Library Objects, Details. See also Natural Library Object Details.
Select objects		Only applies to the unload function.  If no library range is specified, a selection list of all Natural objects available is displayed (see also the section Object Selection).
Error number		A range of user-defined error messages delimited by the first and the last message number.  Only evaluated if the parameter Error Messages (default) is selected on the screen Natural Library Objects, Details. See also Natural Library Object Details.
Details		Invokes the screen Natural Library Objects, Details where you can enter more detailed object specifications. See Natural Library Object Details.
Settings	PF7	Invokes the Unload/Load/Scan Settings screen where you can specify option and parameter settings: see the section Settings.

The section below covers the following topics:

- Natural Library Object Details
- Natural Library Object Properties
- Natural Library Object Exceptions
- Natural Library Object Exception Properties

## Natural Library Object Details

This function is used to specify further selection criteria for Natural library objects.

For a description of keywords and valid values, see also Select-Clause in Direct and PROCEDURE Workplan Syntax in the section Commands.

The screen Unload/Load/Scan Natural Library Objects, Details provides the following fields and PF keys:

Field	PF Key	Explanation
Library		The name of a library or a range of names: see (*) Name.
Select (Library)	PF4	Displays a selection list of all libraries available. See also the section Library Selection.
DBID/FNR	PF5	See DBID/FNR in Natural Library Objects above.
Passw./Ciph.		Only applies to the unload function.  The password and cipher code of the Adabas file where the Natural libraries are stored.
Object Types: Natural programming objects		Natural programming objects.
Object Types: Error messages		User-defined Natural error messages.
Object name		See "Object name" in Natural Library Objects above.
Programming Object Options: S/C-Kind		The kind of Natural programming object:  <b>S</b> Source objects only. <b>C</b> Cataloged objects only. <b>A</b> All saved and/or cataloged objects. This is the default. * <b>W</b> All stowed objects: source and cataloged objects with identical date and time. <b>B</b> Both source and cataloged objects if both exist.  <b>Note:</b> <b>W</b> (stowed) and <b>B</b> (both) are valid for the unload function only. Though <b>W</b> and <b>B</b> can also be entered for the load or scan function, they are treated like <b>A</b> (all objects).
Programming Object Options: Natural types		The types of Natural programming object:  Up to 15 valid external Natural object types, such as <b>P</b> for program.  An asterisk (*) selects all types of Natural programming object (this is the default).
Select Natural types	PF6	Invokes a selection window where you can specify the type of Natural programming object.
Properties	PF7	Invokes an extra screen where you can specify additional properties of Natural programming objects: see Natural Library Object Properties.
Error Messages: Error number		See "Error number" in Natural Library Objects above.

Field	PF Key	Explanation
Error Messages: Language codes		The language code(s) of the error messages: Up to 8 valid Natural language codes. An asterisk (*) selects all Natural language codes.
Error Messages: S/L-Kind		The kind of error message text:  <b>S</b> Short texts of error messages. <b>L</b> Long texts. <b>A</b> All short and/or long texts. This is the default. <b>B</b> Short and long texts if both exist. (unload only)
Exceptions	PF8	Invokes an extra screen where you can specify exceptions to the selection of Natural programming objects: see Natural Library Object Exceptions.

## Natural Library Object Properties

This function is used to specify properties for the Natural library objects selected for processing.

For a description of keywords and valid values, see also Select-Clause in Direct and PROCEDURE Workplan Syntax in the section Commands.

The window Unload/Load/Scan Library Objects, Properties provides the following fields:

Field	Explanation
User ID	The ID of the user who saved or cataloged a Natural programming object. Specify a single user ID or a range of user IDs: see (*) Name.
Programming mode	The programming mode of the Natural programming objects:  <b>R</b> Reporting mode only. <b>S</b> Structured mode only. <b>A</b> No mode check performed. This is the default.
Natural version	The Natural version of the Natural programming objects.  Valid version format: <i>VRSM</i> where <i>V</i> is the 1-digit version, <i>R</i> the 1-digit release, and <i>SM</i> the 2-digit system maintenance level.  You can also specify a range of versions: see (*) Name.
Object Date: Select all objects (no date check)	Performs no check for the object date.
Object Date: Select objects modified between/and	Selects all objects with a save or catalog date and/or time within the range specified in these fields by entering a precise start date and/or time and/or an end date and/or time.  Special dates allowed are: TODAY, YESTERDAY, MONTH, YEAR. See also: (*) Date.  The time must be specified in the format <i>HH:II:SS</i> ( <i>HH</i> = hours, <i>II</i> = minutes, <i>SS</i> = seconds).
Object Date: Select objects modified on	Selects all objects with a save or catalog date and/or time that fits the date/time specified in these fields by entering a precise date and/or time.  Special dates allowed are: TODAY and YESTERDAY. See also: (*) Date.  Valid time format: <i>HH:II:SS</i> ( <i>HH</i> = hours, <i>II</i> = minutes, <i>SS</i> = seconds).
Object Size: Select all objects (no size check)	Performs no check for the object size.
Object Size: Select objects with size between/and	Selects all objects with a size within the range specified in these fields by entering a start size and/or an end size.
Object Size: Select objects with size	Selects all objects with a size that fits the size specified in this field.

## Natural Library Object Exceptions

This function is used to specify exceptions to the selection of Natural library objects.

All objects that match the selection criteria specified in Natural Library Objects, Natural Library Object Details and Natural Library Object Properties are checked against the specifications made on the screen Unload/Load/Scan Library Objects, Exceptions. Objects that match **all** specifications defined as exceptions, are exempted from processing.

For a description of keywords and valid values, see also Select-Clause in Direct and PROCEDURE Workplan Syntax in the section Commands.

The screen Unload/Load/Scan Library Objects, Exceptions is basically identical to the screen Unload/Load/Scan Natural Library Objects, Details. See the relevant section for an explanation of the fields, commands and alternative PF keys listed in the table below. The field "Add/change properties for selection" is used to specify additional properties for Natural programming object exceptions: see Natural Library Object Exception Properties.

## Natural Library Object Exception Properties

This function is used to specify exceptions to the properties of the Natural library objects selected for processing.

The screen Unload/Load/Scan Library Objects, Exceptions provides the following fields:

Field	Explanation
User ID	See User ID in Natural Library Object Properties.
Programming mode	See "Programming mode" in Natural Library Object Properties.
Natural version	See "Natural version" in Natural Library Object Properties.
Object Date: Ignore object date	Performs no check for the object date.
Object Date: Exclude objects modified between/and	Exempts from processing all objects with a save or catalog date and/or time within the range specified in these fields by entering a precise start date and/or time and/or an end date and/or time.  Special dates allowed are: TODAY, YESTERDAY, MONTH, YEAR. See also: (*) Date.  The time must be specified in the format <i>HH:II:SS</i> ( <i>HH</i> = hours, <i>II</i> = minutes, <i>SS</i> = seconds).
Object Date: Exclude objects modified on	Exempts from processing all objects with a save or catalog date and/or time that fits the date/time specified in these fields by entering a precise date and/or time.  Special dates allowed are: TODAY and YESTERDAY. See also: (*) Date.  Valid time format: <i>HH:II:SS</i> ( <i>HH</i> = hours, <i>II</i> = minutes, <i>SS</i> = seconds).
Object Size: Ignore object size	Performs no check for the object size.
Object Size: Exclude objects with size between/and	Exempts from processing all objects with a size within the range specified in these fields by entering a start size and/or an end size.
Object Size: Exclude objects with size	Exempts from processing all objects with a size that fits the size specified in this field.

## Natural System Error Messages

This function is used to select for processing Natural system error messages from the specified Natural system file.

For a description of keywords and valid values, see also Select-Clause in Direct and PROCEDURE Workplan Syntax in the section Commands.

The screen Unload/Load/Scan Natural System Error Messages provides the following fields and PF keys:

Field	PF Key	Explanation
Error number from/to		A range of Natural system error messages delimited by the first and the last message number. Select "Select system error messages" for a list of all system error messages available.
Details	PF6	Invokes the screen Unload/Load/Scan Natural Library Objects, Details where you can enter more detailed object specifications: see Natural System Error Message Details.
Settings	PF7	Invokes the screen Unload/Load/Scan Settings where you can specify option and parameter settings. See the section Settings.

The section below covers the following topics:

- Natural System Error Message Details
- Natural System Error Message Exceptions

### Natural System Error Message Details

This function is used to specify further selection criteria for Natural system error messages.

For a description of keywords and valid values, see also Select-Clause in the section Direct and PROCEDURE Workplan Syntax, Commands.

The screen Unload/Load/Scan System Error Messages, Details provides the following fields and PF keys:

Field	PF Key	Explanation
DBID/FNR		Only applies to the unload function.  The database ID (DBID) and the number of the Adabas (FNR) file where the Natural error messages are stored.
Password/Cipher		Only applies to the unload function.  The password and cipher code of the Adabas file where the Natural error message sources are stored.
Error number from/to		See "Error number" in Natural System Error Messages above.
Language codes		See "Language codes" in Natural Library Object Details.
S/L-Kind		See S/L-Kind in Natural Library Object Details.
Exceptions	PF8	Invokes an extra screen where you can specify exceptions to the selection of Natural system error messages: see Natural System Error Message Exceptions.

## Natural System Error Message Exceptions

This function is used to specify exceptions to the selection of Natural system error messages.

All Natural system error messages that match the selection criteria specified in Natural System Error Messages and Natural System Error Message Details are checked against the specifications made on the screen Unload/Load/Scan System Error Messages, Exceptions. Error messages that match **all** specifications defined as exceptions, are exempted from processing.

For an explanation of the fields provided on the exceptions screen, see Natural System Error Message Details above.

For a description of keywords and valid values, see also Select-Clause in Direct and PROCEDURE Workplan Syntax in the section Commands.

## Natural Command Processor Sources

This function is used to select Natural command processor sources for processing.

For a description of keywords and valid values, see also Select-Clause in Direct and PROCEDURE Workplan Syntax in the section Commands.

The screen Unload/Load/Scan Natural Command Processors provides the following fields and PF keys:

Field	PF Key	Explanation
Library		The name of a Natural command processor library or a range of names: see (*) Name.
Select library	PF4	Invokes a selection list of Natural command processor libraries available. See also Library Selection.
DBID/FNR		Only applies to the unload function.  The database ID (DBID) and the number of the Adabas (FNR) file where the Natural command processor sources are stored.
Password/Cipher		Only applies to the unload function.  The password and cipher code of the Adabas file where the Natural command processor sources are stored.
Object name		The name of a Natural command processor source or a range of names: see (*) Name.
Select objects		Only applies to the unload function.  If no library range has been specified and this field is selected, a selection list of Natural command processor sources available is displayed (see also the section Object Selection).
Exceptions	PF8	Invokes an extra screen where you can specify exceptions to the selection of Natural command processor sources: see Natural Command Processor Source Exceptions.
Settings	PF7	Invokes the Unload/Load/Scan Settings screen where you can specify option and parameter settings. See the section Settings.

## Natural Command Processor Source Exceptions

This function is used to specify exceptions to the selection of Natural command processor sources.

All objects that match the selection criteria specified in Natural Command Processor Sources are checked against the specifications made on the screen Unload/Load/Scan Natural Command Processors, Exceptions. Command Processor Sources that match **all** specifications defined as exceptions, are exempted from processing.

For an explanation of the fields provided in the exceptions window, see Natural Command Processor Sources above.

For a description of keywords and valid values, see also Select-Clause in Direct and PROCEDURE Workplan Syntax in the section Commands.

## Natural-related Objects

This function is used to select Natural-related objects for processing. Natural-related objects are profiles, debug environments and DL/I subfiles.

For a description of keywords and valid values, see also Select-Clause in Direct and PROCEDURE Workplan Syntax in the section Commands.

When you select Natural-related objects on the Unload/Load Type screen, the Select Related Type screen appears where you can specify the type of the Natural-related object: Natural profiles, debug environments or DL/I subfiles.

The section below covers the following topics:

- Natural Profiles
- Natural Profile Exceptions
- Natural Debug Environments
- Natural Debug Environment Exceptions
- Natural DL/I Subfiles
- Natural DL/I Subfile Exceptions

## Natural Profiles

The screen Unload/Load/Scan Natural Profiles provides the following fields and PF keys:

Field	PF Key	Explanation
DBID/FNR Password/Cipher		Only applies to the unload function.  The database ID (DBID), file number (FNR), password and cipher code of the Adabas file where the Natural profiles are stored.  If no values (or 0) are specified, the current FNAT system file is used.
Select (DBID/FNR)	PF5	Only applies to the unload function.  Invokes the Select System File window with a list of all system files available in your Natural environment: see also the section System File Selection.
Profile Types		The type(s) of profile to be processed:  Parameter, Editor, Map and/or Device.
Object name		The name of a profile or a range of names: see (*) Name.
Select (Object name)		Only applies to the unload function.  Invokes the List Profiles screen with a selection list of profiles available (see also the section Object Selection).
Exceptions	PF8	Invokes an extra screen where you can specify exceptions to the selection of profiles: see Natural Profile Exceptions.
Settings	PF7	Invokes the Unload/Load/Scan Settings screen where you can specify option and parameter settings: see the section Settings.

For a description of keywords and valid values, see also Select-Clause in Direct and PROCEDURE Workplan Syntax in the section Commands.

## Natural Profile Exceptions

This function is used to specify exceptions to the selection of Natural profiles.

All objects that match the selection criteria specified in Natural Profiles are checked against the specifications made on the screen Unload/Load/Scan Natural Profiles, Exceptions. Objects that match **all** specifications defined as exceptions, are exempted from processing.

For a description of keywords and valid values, see also Select-Clause in Direct and PROCEDURE Workplan Syntax in the section Commands.

The screen Unload/Load/Scan Natural Profiles, Exceptions provides the following fields:

Field	Explanation
Object name	The name of a profile or a range of names: see (*) Name.
Profile Types	The type(s) of profile to be processed:  Parameter, Editor, Map and/or Device.

## Natural Debug Environments

The screen Unload/Load/Scan Natural Debug Environments provides the following fields and PF keys:

Field	PF Key	Explanation
Library		The name of a library or a range of names: see (*) Name.
Select library	PF4	Displays a selection list of all libraries available. See also the section Library Selection.
DBID/FNR Password/Cipher		Only applies to the unload function.  The database ID (DBID), file number (FNR), password and cipher code of the Adabas file where the Natural debug environments are stored.  If no values (or 0) are specified, the current FNAT system file is used.
Object name		The name of a debug environment or a range of names: see (*) Name.
Select objects		Only applies to the unload function.  Displays a selection list of debug environments available (see also the section Object Selection).
Exceptions	PF8	Invokes an extra screen where you can specify exceptions to the selection of profiles: see Natural Debug Environment Exceptions.
Settings	PF7	Invokes the Unload/Load/Scan Settings screen where you can specify option and parameter settings: see the section Settings.

For a description of keywords and valid values, see also Select-Clause in Direct and PROCEDURE Workplan Syntax in the section Commands.

## Natural Debug Environment Exceptions

This function is used to specify exceptions to the selection of Natural debug environments.

All objects that match the selection criteria specified in Natural Debug Environments are checked against the specifications made on the screen Unload/Load/Scan Debug Environments, Exceptions. Objects that match **all** specifications defined as exceptions, are exempted from processing.

For a description of keywords and valid values, see also Select-Clause in Direct and PROCEDURE Workplan Syntax in the section Commands.

The screen Unload/Load/Scan Debug Environments, Exceptions provides the following fields:

Field	Explanation
Library	The name of a library or a range of names: see (*) Name.
Select (Library)	Displays a selection list of all libraries available. See also the section Library Selection.
Object name	The name of a debug environment or a range of names: see (*) Name.

## Natural DL/I Subfiles

The screen Unload/Load/Scan Natural DL/I Subfiles provides the following fields and PF keys:

Field	PF Key	Explanation
DBID/FNR Password/Cipher		Only applies to the unload function.  The database ID (DBID), file number (FNR), password and cipher code of the Adabas file where the Natural DL/I subfiles are stored.  If no values (or 0) are specified, the current FDIC system file is used.
Select (DBID/FNR)	PF5	Only applies to the unload function.  Invokes the Select System File window with a list of all system files available in your Natural environment: see also the section System File Selection.
Subfile Types		The type(s) of DL/I subfile to be processed: NSB and/or NDB.
Object name		The name of a DL/I subfile or a range of names: see (*) Name.
Select (Object name)		Only applies to the unload function.  Displays a selection list of DL/I subfiles available (see also the section Object Selection).
Exceptions	PF8	Invokes an extra screen where you can specify exceptions to the selection of profiles: see Natural DL/I Subfile Exceptions.
Settings	PF7	Invokes the Unload/Load/Scan Settings screen where you can specify option and parameter settings: see the section Settings.

## Natural DL/I Subfile Exceptions

This function is used to specify exceptions to the selection of Natural DL/I subfiles.

All objects that match the selection criteria specified in Natural DL/I Subfiles are checked against the specifications made on the screen Unload/Load/Scan Natural DL/I Subfiles, Exceptions. Objects that match **all** specifications defined as exceptions, are exempted from processing.

For a description of keywords and valid values, see also Select-Clause in Direct and PROCEDURE Workplan Syntax in the section Commands.

The screen Unload/Load/Scan Natural DL/I Subfiles, Exceptions provides the following fields:

Field	Explanation
Object name	The name of a DL/I subfile or a range of names: see (*) Name.
Subfile Types	The type(s) of DL/I subfile to be processed: NSB and/or NDB.

## DDMs

This function is used to select Natural DDMs (data definition modules) for processing.

For a description of keywords and valid values, see also Select-Clause in Direct and PROCEDURE Workplan Syntax in the section Commands.

The screen Unload/Load/Scan DDMs provides the following fields and PF keys:

Field	PF Key	Explanation
FDIC DBID/FNR FDIC Password/Cipher		Only applies to the unload function.  The database ID (DBID), file number (FNR), password and cipher code of the Adabas file where the DDMs are stored.  If no values (or <b>0</b> ) are specified, the current FDIC system file is used.
DDM name		Only applies to the unload function.  The name of a DDM or a range of names: see (*) Name.
Select objects		Only applies to the unload function.  Displays a selection list of DDMs available (see also the section Object Selection).
Properties	PF7	Invokes an extra screen where you can specify additional properties of DDMs: see DDM Properties.
Exceptions	PF8	Invokes an extra screen where you can specify exceptions to DDMs: see DDM Exceptions.
Settings	PF7	Invokes the Unload/Load/Scan Settings screen where you can specify option and parameter settings: see the section Settings.

The section below covers the following topics:

- DDM Properties
- DDM Exceptions

## DDM Properties

This function is used to specify properties for the DDMs selected for processing.

For a description of keywords and valid values, see also Select-Clause in Direct and PROCEDURE Workplan Syntax in the section Commands.

The window Unload/Load/Scan DDMs, Properties provides the following fields:

Field	Explanation
User ID	The ID of the user who saved or cataloged a Natural programming object. Specify a single user ID or a range of user IDs: see (*) Name.
DDM DBID	The database ID (DBID) of the DDMs. Valid entries are: 1 to 65535 or 0 (all DBIDs)
DDM FNR	The file number (FNR) of the DDMs: Valid entries are: 1 to 65535 or 0 (all FNRs).
Object Date: Select all objects (no date check)	Performs no check for the object date.
Object Date: Select objects modified between/and	See Object Date in Natural Library Object Properties.
Object Date: Select objects modified on	See Object Date in Natural Library Object Properties.
Object Size: Select all objects (no size check)	Performs no check for the object size.
Object Size: Select objects with size between/and	Selects all objects with a size within the range specified in these fields by entering a start size and/or an end size.
Object Size: Select objects with size	Selects all objects with a size that fits the size specified in this field.

## DDM Exceptions

This function is used to specify exceptions to the selection of DDMs.

All objects that match the selection criteria specified in DDMs and DDM Properties are checked against the specifications made on the screen Unload/Load/Scan DDM, Exceptions. Objects that match **all** specifications defined as exceptions, are exempted from processing.

For a description of keywords and valid values, see also Select-Clause in Direct and PROCEDURE Workplan Syntax in the section Commands.

The screen Unload/Load/Scan DDMs, Exceptions provides the following fields:

Field	Explanation
DDM name	The name of a DDM or a range of names: see (*) Name.
DDM DBID	See DDM DBID in DDM Properties.
DDM FNR	See DDM FNR in DDM Properties.
User ID	See User ID in DDM Properties.
Object Date: Ignore object date	Performs no check for the object date.
Object Date: Exclude objects modified between/and	See Object Date in Natural Library Object Exception Properties.
Object Date: Exclude objects modified on	See Object Date in Natural Library Object Exception Properties.
Object Size: Ignore object size	Performs no check for the object size.
Object Size: Exclude objects with size between/and	Exempts from processing all objects with a size within the range specified in these fields by entering a start size and/or an end size.
Object Size: Exclude objects with size	Exempts from processing all objects with a size that fits the size specified in this field.

## FDTs

This function is used to select Adabas FDTs (Field Definition Tables) for processing.

For a description of keywords and valid values, see also Select-Clause in Direct and PROCEDURE Workplan Syntax in the section Commands.

The screen Unload/Load/Scan FDTs provides the following fields and PF keys:

Field	PF Key	Explanation
DBID		The database ID (DBID) where the FDT is located.  Load and scan: A DBID or <b>0</b> for all DBIDs.
FNR		The file number (FNR) where the FDT is located.  Load and scan: A FNR or <b>0</b> for all FDTs.
Password/Cipher		Only applies to the unload and load functions.  The Adabas password and the cipher code of the Adabas file where the FDT is located.
Settings	PF7	Invokes the Unload/Load/Scan Settings screen where you can specify option and parameter settings. See the section Settings.

## Use Selection or List Workplan

This function is used to specify a Workplan of the type SELECTION or LIST. These Workplans specify selection criteria for the objects to be processed. See also the section Workplans.

The screen Unload/Load/Scan Selection or List provides the following fields and PF keys:

Field	PF Key	Explanation
Name	PF4	The name of the Workplan to be processed.
List Workplan		Displays the contents of the Workplan specified in the Name field.
Select Workplan	PF5	Displays a list of all Workplans available. See also List Workplan in the section Administration.
Settings	PF7	Invokes the Unload/Load/Scan Settings screen where you can specify option and parameter settings. See the section Settings.

# Object Handler - Settings

The settings option is used to specify option settings for the unload, load, find or scan function and parameter settings for the unload or load function.

## Note:

\* The notation "see (\*)" in the tables below regarding name, date and time parameters refers to the section Name, Date and Time Specification.

### To invoke the Unload/Load/Scan Settings screen

- On any of the unload, load or scan screens, enter the internal command SETTINGS (see also Commands for CUIs).  
Or, activate advanced-user mode, choose a function and press ENTER to start the processing procedure.  
Or, on the advanced-user screens, choose PF7/Setti.

Unless selected by default, to activate the options provided on the Unload/Load/Scan Settings screen described below, mark the corresponding input field with any single character.

The section below covers the following topics:

- Settings Screen Fields
- Set Additional Options
- Set Global Parameters

## Settings Screen Fields

The Unload/Load/Scan Settings screen provides the following fields and PF keys:

Field	PF Key	Explanation
Transfer format		<p>Only valid if "Use default options" (this is the default) has been selected.</p> <p>If selected, the data to be processed is written in Transfer format into/from the work file. See also Work File Format in the section Work Files.</p> <p>Unload function: The data to be unloaded is written in Transfer format into the work file. Note that if you want to change the setting of this field for a subsequent unload, you need to return to the Natural Object Handler Main Menu or enter the command GO UNLOAD END (see Commands for CUIs) and restart the unload function.</p> <p>Load and scan functions: The data to be loaded or scanned are expected to be in Transfer format.</p>
Use PC File		<p>Only applies if Entire Connection is installed.</p> <p>Writes data into an Entire Connection work file.</p>
PC File		<p>Only applies if Entire Connection is installed.</p> <p>The path and name of the Entire Connection work file.</p>
Use default options		<p>Default options are used (this is the default). See also SYSOBJH Profile and Set Additional Options below.</p>

Field	PF Key	Explanation
Set additional options	PF4	Only valid if "Use default options" has been selected.  Invokes the Options screen where you can modify the default settings and enter additional options for the processing sequence. For the options available, see Set Additional Options below.
Use Option Workplan		A Workplan of the type OPTION is used. See also Workplans.
Name (next to Use Option Workplan)		Only valid if Use Option Workplan has been selected.  The name of a Workplan of the type OPTION to be used.
List Option Workplan	PF6	Only valid if Use Option Workplan has been selected.  Displays the contents of the Workplan specified in the field Name next to Use Option Workplan.
Select Option Workplan	PF5	Only valid if Use Option Workplan has been selected.  Displays a selection list of available Workplans of the type OPTION (see also List Workplan in the section Administration).
Do not use parameters		If selected (default), no parameters are set.
Use global parameters		Global parameters are used. See Set Global Parameters below.
Set global parameters	PF7	Only valid if "Use global parameters" has been selected.  Invokes the global parameters screen. See Set Global Parameters and Parameter-Setting (Direct and PROCEDURE Workplan Syntax, Commands) for a description of keywords and valid values.
Use Parameter Workplan		A Workplan of the type PARAMETER is used. See also Workplans.
Name (next to Use Parameter Workplan)		Only valid if Use Parameter Workplan has been selected.  The name of a Workplan of the type PARAMETER to be used.
List Parameter Workplan	PF9	Only valid if Use Parameter Workplan has been selected.  Displays the contents of the Workplan specified in the field Name next to Use Parameter Workplan.
Select Parameter Workplan		Only valid if Use Parameter Workplan has been selected.  Displays a selection list of available Workplans of the type PARAMETER. See also List Workplan in the section Administration.

## Set Additional Options

The sections contained in the Options screen are described below. Note that not all of the sections may appear on the screen, because they depend on the function used, the settings defined and the products installed.

- Work File and Report Options
- XREF Options
- Version Check
- Transfer Options
- Replace Options
- Number to Process
- FDIC Settings
- FSEC Settings

For a description of keywords and valid values, see also Option-Setting in the section Direct and PROCEDURE Workplan Syntax, Commands.

## **Work File and Report Options**

The options provided for work files and reports are described below:

Field	Explanation
Use PC File	Only applies if Entire Connection is installed.  Writes data into an Entire Connection work file.
PC File	Only applies if Entire Connection is installed.  The path and name of the Entire Connection work file.
Write report	Writes a report of the objects processed to a text member in the Workplan library. "Write report" is the default setting for object processing.  To display the report, enter the internal command SHOW REPORT FILE (see also the section Commands for CUIs).
Start new Report	Only valid if "Write report" has been selected.  Deletes the contents of the text member in the Workplan library before a new report is written. Otherwise, a new report is appended to the existing one.
Report text member	Only valid if "Write report" has been selected.  The name of the text member stored in the Workplan library into which the report is written.
Select text member	Displays a list of all text members stored in the Workplan library. From this list, you can select a "Report text member".
Write restart information	Only applies to the load function.  With this option you can resume load functions that terminated abnormally.  The selection criteria, the options, the parameter settings and the objects already processed are written as restart information into a text member in the Workplan library (see "Restart text member" below). If the load function terminates before the work file has been processed completely, with the restart function you can continue from the point of termination. To invoke the restart function, enter the internal command GO RESTART (see also the section Commands for CUIs).
Restart text member	Only applies to the load function and if "Write restart information" has been selected.  The name of the text member in the Workplan library into which the restart information is written.  If you do not specify a name, the Object Handler generates a name and assigns it to the text member.
Select text member	Displays a list of all text members stored in the Workplan library. From this list, you can select a "Restart text member".

## XREF Options

XREF options are only available when unloading or loading data in internal format, that is, if the field "Transfer format" has **not** been selected. To process XREF data, Predict must be installed.

The XREF options provided and the functions to which they apply are described below:

Field	Explanation	Function
Yes (unload/load XREF data)	Unloads cataloged objects and their cross-reference data, if any. Loads cataloged objects and their cross-reference data if cross-references exist in the work file.	Unload Load
No (ignore XREF data)	No XREF data is processed.	Unload Load
Force	Loads cataloged objects and their cross-reference data only if cross-references exist in the work file and if Predict entries exist for the objects in the FDIC system file.	Load
Doc	Loads cataloged objects only if Predict entries exist for the objects in the FDIC system file.	Load
Special	Loads cataloged objects and their cross-reference data (if any).	Load

## Version Check

The version check option is only available when loading data in internal format, that is, if the field "Transfer format" has not been selected.

If "Version check" is marked, the Natural version under which the objects were cataloged and written into the work file is compared with the current Natural version. Objects cataloged under a Natural version higher than the current one will be rejected.

## Transfer Options

Transfer options are only available when processing data in Transfer format, that is, if the field "Transfer format" has been selected.

The transfer options provided and the functions to which they apply are described below:

Option	Explanation	Function
Substitute line references	Only applies if source-code line numbers are used for statement references. If line numbers are used as references in the source code, the line numbers of referenced lines and the line number references are replaced with labels. The sources are not modified in the database.	Unload
Include Line Numbers	If you choose this option, the line numbers will be transferred. ( By default, line numbers in Natural objects are not transferred.)	Unload
Incorporate free rules	If Predict is installed, Predict rules associated with a map are incorporated into the map source.	Unload
Use conversion table	Unload: Converts data into ASCII format by using the internal Natural conversion table (System table) or a conversion table defined by the user (User table).  Load: Converts data into EBCDIC format by using the internal Natural conversion table (System table) or a conversion table defined by the user (User table). Note that this only applies if the data in the work file is in ASCII format or if a conversion program is specified (see User table).	Unload Load

Option	Explanation	Function
System table	<p>Only valid if "Use conversion table" has been selected.</p> <p>Unload: Converts data into EBCDIC format by using the internal Natural conversion table.</p> <p>Load: Converts data into ASCII format by using the internal Natural conversion table.</p>	Unload Load
User table	<p>Only valid if "Use conversion table" has been selected.</p> <p>If the name of a conversion program has been entered in the field, data is converted into EBCDIC or ASCII format by using the conversion program defined. To specify an individual conversion program, the program must be located in the library SYSOBJH or one of its steplibs. See the example program OTNCONEA in the library SYSOBJH.</p>	Unload Load
Translate to upper case	Translates any source code to be loaded into upper case.	Load
Data area format	<p>Only applies to data areas.</p> <p>Specifies the format in which to unload or load data area sources. Possible values are:</p> <p>N     Converts data areas into new internal data area format.</p> <p>O     Converts data areas into old internal data area format.</p> <p>If one or more data area sources cannot be converted into old internal data area format, the Object Handler issues a corresponding message when unloading is complete. In addition, in the Status column of the unload report generated by the unload function, a corresponding remark appears next to the names of the data area sources affected.</p> <p>*     Does not convert data areas. This is the default.</p> <p>For further details, see Data Area Editor in the Natural Editors documentation.</p>	Unload Load

## Replace Options

The replace options described below only apply to the load function:

Do not replace	Does not replace any objects. This is the default.
Replace all	Replaces all objects.
Replace obsolete	Replaces objects with a date older than the date of the objects in the load file.
Replace except newer	Replaces all objects except those with a date newer than the date of the objects in the load file.

## Number to Process

Number to process only applies to the load and scan functions.

In the field "Number to process", enter a value with a maximum of 5 digits.

If a value greater than 0 is specified, the load or scan function stops after the specified number of objects has been processed.

### Note:

If a cataloged Natural object is processed directly after the source object of the same name, they are considered one object.

## FDIC Settings

FDIC settings are used to specify the Predict file (FDIC) to be used for processing XREF data (only applies if Predict is installed) or load DDMs:

DBID	The database ID (DBID) where the FDIC file is located.
FNR	The file number (FNR) where the FDIC file is located.
Password	Optional. The Adabas password of the Adabas file where the FDIC file is located.
Cipher	Optional. The cipher code of the Adabas file where the FDIC file is located.

## FSEC Settings

FSEC settings only apply if Natural Security is installed.

FSEC settings are used to specify the Natural Security data file (FSEC) to be used for security checks:

DBID	The database ID (DBID) where the FSEC file is located.
FNR	The file number (FNR) where the FSEC file is located.
Password	Optional. The Adabas password of the Adabas file where the FSEC file is located.
Cipher	Optional. The cipher code of the Adabas file where the FSEC file is located.

## Set Global Parameters

### Not applicable to the scan function.

Global parameters are used to change parameter settings for the objects to be processed with the load or unload function, and to change the target environment for the load function.

For valid parameter settings, see also Parameter-Setting in the section Direct and PROCEDURE Workplan Syntax, Commands.

If global parameters are specified during the unload function, the parameter settings affect the objects before they are written into the work file. If they are specified during the load function, the parameter settings affect the objects before they are written to the target environment.

The fields on the Parameters screen are described below. The values that can be specified to change parameter settings, are entered in the fields Check Value and New Value. Check Value and New Value do not apply to the field "Error number difference" and the fields in the section "System files for load".

If no value has been entered in Check Value, the value entered in New Value affects all objects to which the specific parameter setting applies. If a value has been entered in Check Value, the value entered in New Value only affects objects to which the specific parameter setting and the value entered in Check Value apply. If a Check Value or New Value is not relevant to the type of object to be processed, any value entered in either field will be ignored. For example: Natural system error messages have no library name. Therefore, when processing Natural system error messages, a value entered in Check Value or New Value for the Library field will be ignored.

See also Rules for New Values below.

The following data can be specified for the load and the unload functions:

Field	Explanation
Object name	<p>Check Value: A single object name or a range of names: see (*) Name.</p> <p>New Value: A single object name or a range of names: see (*) Name.</p> <p>See also Rules for New Values below.</p>
Library	<p>Check Value: A single library name or a range of names: see (*) Name.</p> <p>New Value: A single library name or a range of names: see (*) Name.</p> <p>See also Rules for New Values below.</p>
Date	<p>Check Value: A single date or a range of dates: see (*) Date and (*) Time.</p> <p>New Value: A single date or a range of dates: see (*) Date and (*) Time.</p> <p>See also Rules for New Values below.</p>
Time	<p>Check Value: A time.</p> <p>New Value: A time.</p>
User ID	<p>Check Value: A single user ID or a range of user IDs: see (*) Name.</p> <p>New Value: A single user ID or a range of user IDs: see (*) Name.</p> <p>See also Rules for New Values below.</p>

Field	Explanation
Terminal ID	<p>Check Value: A single terminal ID or a range of terminal IDs: see (*) Name.</p> <p>New Value: A single terminal ID or a range of terminal IDs: see (*) Name.</p> <p>See also Rules for New Values below.</p>
Lang. Codes	<p>Only applies when processing Natural system error messages or user-defined messages.</p> <p>Check Value: Up to 8 language codes.</p> <p>New Value: Up to 8 language codes.</p> <p>If more than 1 language code is specified, Check Value must contain the same number of language codes. In this case, the language code in Check Value is replaced by the language code in the corresponding New Value.</p>
Error number difference (+/-nnnn)	<p>Only applies when processing Natural system error messages or user-defined messages.</p> <p>A 4-digit positive or negative value. It can only be specified if start and end values are entered as selection criteria (Error number from/to). Otherwise, it is not possible to check if the result is valid (valid range: 1 to 9999).</p>
FDT DBID/FNR	<p>Check Value: A valid DBID and/or FNR of Adabas FDTs.</p> <p>New Value: A valid DBID and/or FNR of Adabas FDTs.</p>

The following data can be specified or Natural system files for the load function only in the "System files for load" section of the Parameters screen:

Field	Explanation
Load FNAT DBID/FNR Password Cipher	The database ID (DBID) and file number (FNR) of the target FNAT system file. This system file is used for all library objects whose library name starts with SYS, but not SYSTEM. Additionally, you can specify the Adabas password and cipher code.
Load FUSER DBID/FNR Password Cipher	The DBID and FNR of the target FUSER system file. This system file is used for all library objects whose library name does not start with SYS, and for the library SYSTEM. Additionally, you can specify the Adabas password and cipher code.
Select (FNAT/FUSER)	Invokes the Select System File window with a list of all system files available in your Natural environment: see System File Selection.
Load NCP DBID/FNR Password Cipher	The DBID and FNR of the target Adabas file into which the Natural command processor sources are to be loaded. Additionally, you can specify the Adabas password and cipher code.



## Rules for New Values

The following applies to New Value for "Object name", Library, Date/Time, User ID, Terminal ID and Lang. Codes.

If New Value contains a range with an asterisk (\*), such as ABC\*, the number of characters before the asterisk (\*) determines the number of characters to be replaced in Check Value. This is also valid if Check Value is shorter than the range specified in New Value (see the second example in Examples below).

### Examples:

1. Object name is ABCDEFG and New Value is set to ZYX\* the resulting object name is ZYXDEFG.
2. Object name is AB and New Value is set to ZYX\* the resulting object name is ZYX.
3. Object date is 2002-03-26 and New Value is set to 2003\* the resulting object date is 2003-03-26.

# Object Handler - Workplans

Workplans are used to define individual standard procedures to further automate the load and unload process. Workplans are Natural objects of the type Text. They are, by default, stored in the library WORKPLAN located in the current FUSER system file.

With the Administration function (see the relevant section) or Object Handler direct commands (see Direct and PROCEDURE Workplan Commands), you can create and select Workplans or change the default library for Workplans.

To change the default library, you can also use the SYSOBJH profile by setting the option Workplan-Library (see SYSOBJH Profile).

This section covers the following topics:

- Contents of Workplans
- Examples of Workplans
- Referencing Workplans

---

## Contents of Workplans

A Workplan consists of a header (generated by the Object Handler) and an associated instructional or textual part. Instructional parts contain Object Handler commands and parameter and/or option settings. Textual parts only contain plain text. Header and instructional or textual parts can contain comments (for example, the short description of the Workplan) that must start with the delimiter character `/*` and are restricted to one line.

There are six types of Workplan: PROCEDURE, SELECTION, LIST, PARAMETER , OPTION and TEXT.

The table below lists the valid headers (to be entered if creating a Workplan outside the Object Handler) for the corresponding types of Workplan and describes the contents of the instructional or textual part. Additionally, it provides cross references to the clauses that apply when specifying Object Handler commands. The Object Handler commands provided are explained in Direct and PROCEDURE Workplan Syntax (see the section Commands).

Valid Headers	Contents	Related Commands/Clauses
TYPE PROCEDURE	An Object Handler procedure. It can contain any combination of Object Handler commands available for PROCEDURE. Enter a sequence of commands separated by semicolons (;).	Direct and PROCEDURE Workplan Commands (section: Commands)
TYPE SELECTION	A selection criterion for objects. It can be used in Object Handler Workplan commands.	Select-Clause (section: Direct and PROCEDURE Workplan Syntax, Commands)
TYPE LIST	A list of objects. It can be used in used in Object Handler Workplan commands.	Object List (section: Direct and PROCEDURE Workplan Syntax, Commands)
TYPE PARAMETER	Parameters for the unload or load function, for example, old and new name. It can be used in Object Handler Workplan commands.	Parameter-Setting (section: Direct and PROCEDURE Workplan Syntax, Commands)
TYPE OPTION	Options for the unload or load function, for example, report settings. It can be used in Object Handler Workplan commands.	Option-Setting in the (section: Direct and PROCEDURE Workplan Syntax, Commands)
TYPE TEXT	Comments or any other text that can be used for documentation purpose.	Not applicable.

## Examples of Workplans

The following table contains examples of the instructional parts of Workplans.

Workplan Type	Instruction	Explanation
PROCEDURE	FINDLIB * LIB TEST	Check whether the library TEST exists.
PROCEDURE	UNLOAD A* LIB TEST	Unload from the library TEST into Work File 1 all Natural programming objects starting with A, and all user-defined error messages.
SELECTION	* LIB TEST	Process all objects from the library TEST.
TEXT	This is a Workplan comment.	Any text.

## Referencing Workplans

The following syntax applies when referencing an Object Handler Workplan in connection with the Object Handler direct commands described in Direct and PROCEDURE Workplan Commands:

```
( workplan-name
  [LIBRARY library-name]
  [DBID dbid [FNR fnr] ] [NAME vsam-name]
  [CIPHER cipher]
  [
    {
      PASSWORD
      PSW
    } password
  ]
)
```

### Keyword Explanation

The table below contains describes the keywords and values that apply to the syntax for referencing Workplans:

Keyword	Values	Default Value
<i>workplan-name</i>	A valid Natural object name.	No default
<b>LIBRARY</b>	The name of the Natural library where the Workplan is located.	WORKPLAN
<b>DBID</b>	The ID of the Adabas database (DBID) where the Workplan library is located.	0 (current FNAT/FUSER)
<b>FNR</b>	The number of the Adabas file (FNR) where the Workplan library is located.	0 (current FNAT/FUSER)
<b>NAME</b>	Only applies to objects on mainframes.  The name of a valid VSAM file where the Workplan library is located.	blank (current FNAT/FUSER)
<b>CIPHER</b>	Only applies to objects on mainframes.  An 8-digit cipher code.	blank (current FNAT/FUSER)
<b>PASSWORD</b>	Only applies to objects on mainframes.  An 8-character Adabas password.	blank (current FNAT/FUSER)

# Name, Date and Time Specification

You can use a name, a date, a time or a range of names, dates and times to select Natural library objects, Natural command processor sources, Natural-related objects or Natural DDMs (data definition modules).

This section contains information on the specification options provided for:

- Name
- Date
- Time

## Name

You can specify a name or a range of names.

In the list of options below, *value* is any combination of one or more characters:

	Input	Items Selected
	<i>value</i>	All items with names equal to <i>value</i> .
	*	All items.
	>	
	?	All items with any single character for each question mark (?) entered.
Leading Characters	<i>value</i> *	All items with names that start with <i>value</i> . Example: AB* Selected: AB, AB1, ABC, ABEZ Not selected: AA1, ACB
Wildcard	<i>value</i> ?	All items with names that start with <i>value</i> and end with any single character for each question mark (?) entered. Example: ABC? Selected: ABCA, ABCZ Not selected: AXC, ABCAA
	<i>value</i> ? <i>value</i> ?	All items that match <i>value</i> combined with asterisk (*) and question mark (?) in any order. Example: A?C*Z Selected: ABCZ, AXCBBBZ, ANCZ Not selected: ACBZ, ABDEZ, AXCBBBZA
	<i>value</i> * <i>value</i> ?	
	* <i>value</i> ? <i>value</i> *	
Start Value	<i>value</i> >	All items with names greater than or equal to <i>value</i> . Example: AB> Selected: AB, AB1, BBB, ZZZZZZ Not selected: AA1, AAB
End Value	<i>value</i> <	All items with names less than or equal to <i>value</i> . Example: AX< Selected: AB, AWW, AX Not selected: AXA, AY

**Note:**

The parameter specification option New Value only allows leading characters (asterisk (\*) notation). See Rules for New Values in Set Global Parameters (Settings).

## Date

All date values within the Object Handler are specified in international date format.

You can specify a date, a range of dates, a special date or a range of special dates. A date must be specified in the format *YYYY-MM-DD* (*YYYY* = year, *MM* = month, *DD* = day).

In the list of options below, *value* is any combination of one or more digits:

	<b>Input Value</b>	<b>Items Selected</b>
Date	<i>YYYY-MM-DD</i>	All items with a date equal to <i>YYYY-MM-DD</i> . Example: 2003-02-15
Leading characters	<i>value*</i>	All items with a date that starts with <i>value</i> . Example: 2002* Selected: 2002-01-01, 2002-12-31 Not selected: 2001-12-31, 2003-01-01
Start value	<i>value&gt;</i>	All items with a date greater than <i>value</i> . Example: 2002-05> Selected: 2002-05-01, 2002-12-31, 2003-01-01, 2003-12-31 Not selected: 2002-04-31, 2001-12-31  Special dates can be used as <i>value</i> (see below).
End value	<i>value&lt;</i>	All items with a date less than <i>value</i> . Example: 2003-02< Selected: 2002-05-01, 2002-12-31, 2003-01-01, 2003-01-31 Not selected: 2003-02-01, 2003-05-18  Special dates can be used as <i>value</i> (see below).
<b>Special Dates</b>		
<u>T</u> ODAY (+/- <i>nnnn</i> )		All items with the date of the current day.  The day can be followed by <b>+<i>nnnn</i></b> or <b>-<i>nnnn</i></b> where <i>nnnn</i> has a maximum of 4 digits. The resulting date is computed as the date of the current day plus or minus <i>nnnn</i> days. Example: the current date is 2003-03-01 TODAY +5 results in 2003-03-06
<u>Y</u> ESTERDAY		All items with the date of the day before the current day.

	Input Value	Items Selected
<u>MONTH</u>		<p>All items with the date range of the current month.</p> <p>Example: the current month is 2003-02. Selected: 2003-02-01, 2003-02-30. Not selected: 2003-03-01.</p> <hr/> <p>FMDATE: Starts with the first day of the current month.</p> <p>TODATE: Ends with the last day of the current month.</p> <p>If the values of FMDATE and TODATE are identical, the selection is restricted to one day.</p>
<u>YEAR</u>		<p>All items with the date range of the current year.</p> <p>Example: the current year is 2003. Selected: 2003-01-01, 2002-12-31. Not selected: 2002-31-12.</p> <hr/> <p>FMDATE: Starts with the first day of the current year.</p> <p>TODATE: Ends with the last day of the current year.</p> <p>If the values of FMDATE and TODATE are identical, the selection is restricted to one year.</p>

**Note:**

The parameter specification option New Value only allows leading characters (asterisk (\*) notation). See Rules for New Values in Set Global Parameters (Settings).

## Time

You can specify a time or a range of times. The time must be specified in the format *HH:II:SS* (*HH* = hours, *II* = minutes, *SS* = seconds).

In the list of options below, *value* is any combination of one or more digits:

	Input Value	Items Selected
Time	<i>HH:II:SS</i>	<p>All items with a time equal to <i>HH:II:SS</i>.</p> <p>Example: 14:15:16</p>
Leading characters	<i>value</i> *	<p>All items with a time that starts with <i>value</i>.</p> <p>Example: 13:*</p> <p>Selected: 13:00:00, 13:10:53, 13:59:59 Not selected: 12:59:59, 14:00:00</p>

# Object Handler - Work Files

This section describes work files and valid formats that apply to the unload, load and scan functions of the Object Handler.

See also Work File Options in the section Settings.

This section covers the following topics:

- Work File Assignment
- Work File Format

## Work File Assignment

The following table lists the work files used by the Object Handler:

File	Explanation
Work File 1	Used for the unload, load and scan functions.  Contains the data unloaded.
Work File 4	Used to record the execution of a command that contains a DIRECT clause (see Direct and PROCEDURE Workplan Commands).
Work File 7	Only used if Entire Connection is installed.  Work File 7 must be defined as Entire Connection work file to be used for the unload, load and scan functions when Use PC File is specified for the Unload/Load/Scan Settings.  Contains the data unloaded.
Work File 8	Only applies if Entire Connection is installed.  Work File 8 must be defined as Entire Connection work file to be used for the unload, load and scan functions when Use PC File is specified for the Unload/Load/Scan Settings.

## Work File Format

There are two file formats for unloading objects in the source environment into work files and for loading them from work files into the target environment: an internal format and the Transfer format. To transfer binary data, the work files must be of internal format. To transfer text data, the work files must be of Transfer format.

The section below covers the following topics:

- Internal Format
- Transfer Format

### Internal Format

This format is used to transfer Natural sources and cataloged objects, error messages, Natural command processor sources, Adabas FDTs (Field Definition Tables), Natural DDMs (data definition modules) and Natural-related objects from one environment to another. To achieve this, the Object Handler uses the internal format, an internal record layout for the work files.

With the internal format activated, Natural objects are read from the source environment and written into a Natural work file by using the unload function of the Object Handler. This work file can be transported to another environment with standard file transfer services. In the target environment, the objects can then be read from the work file and loaded into the local database system with the load function of the Object Handler.

To transfer objects between identical platforms, use work files of internal format.

**Note:**

Work files created with the Natural utility NATUNLD must be processed in internal format. Work files created by the Object Handler in internal format can be processed with the Natural utility NATLOAD. However, this only applies to objects that can be transferred with NATUNLD or NATLOAD: Natural programming objects, DDMs and Natural error messages. Other objects are ignored. The work files must be created on a server of the same platform where NATUNLD or NATLOAD is applied. See also NATUNLD/NATLOAD Utilities.

## Transfer Format

See also Transfer Format in the section Settings.

This format is used to transfer the sources of Natural objects, Natural command processor sources, error messages, DDMs and Adabas FDTs from one hardware platform to another. To achieve this, the Object Handler uses the Transfer format, a general record layout for work files containing load or unload data. The Transfer format is independent of any hardware platforms.

With the option "Transfer format" set, the unload function of the Object Handler reads Natural objects from a hardware platform and then restructures them.

Formatted records are written into a Natural work file that can be transported to another platform with standard file transfer services. On the target platform, the load function of the Object Handler then reads the objects from the work file and loads them into the local file or database system. The objects read from the work file are restructured according to the structure of the new hardware platform.

Use work files of Transfer format to transfer Natural objects between mainframe and UNIX/Windows platforms.

**Note:**

Use Transfer format to process work files created by the Natural utility SYSTRANS (see the relevant section). Work files created by the Object Handler in Transfer format can be processed with the utility SYSTRANS on all platforms.

# Object Handler - Commands - Overview

The Object Handler provides commands to directly execute the unload, load, scan or find function, maintain Workplans, define special settings and navigate through the screens.

This section provides information on the commands available and the command syntax:

- Direct and PROCEDURE Workplan Commands
- Direct and PROCEDURE Workplan Syntax
- Object Handler Commands for CUIs

# Direct and PROCEDURE Workplan Commands

This section describes the Object Handler direct commands provided for executing Object Handler functions and maintaining Workplans of the type PROCEDURE. See Direct and PROCEDURE Workplan Syntax for an explanation of the Object Handler clauses mentioned below.

**DIRECT**

The DIRECT clause can contain any of the commands used with the Natural utilities NATLOAD, NATUNLD and SYSTRANS. If you specify more than one direct command, enter a semicolon (;) between the commands to separate them from one another.

**Note:**

The report produced by the DIRECT clause (for example, when executing NATLOAD, NATUNLD or SYSTRANS commands) is written into Work File 4.

**EXECUTE** (*procedure-workplan*)

Executes a Workplan of the type PROCEDURE.

**UNLOAD** *select-clause* [*parameter-setting*] [*option-setting*]

Unloads the objects defined in the Select-Clause with the parameters defined in Parameter-Setting with the options defined in Option-Setting.

**LOAD** *select-clause* [*parameter-setting*] [*option-setting*]

Loads the objects defined in the Select-Clause with the parameters defined in Parameter-Setting with the options defined in Option-Setting.

**LOADALL** [*parameter-setting*] [*option-setting*]

Loads all objects from a work file with the parameters defined in Parameter-Setting with the options defined in Option-Setting.

**SCAN** *select-clause* [*option-setting*]

Scans a work file for the objects defined in the Select-Clause with the options defined in Option-Setting.

**SCANALL**    *[option-setting]*

Scans a work file for all objects with the options defined in Option-Setting.

**DELETE**    *select-clause*    *[option-setting]*

Deletes the objects defined in the Select-Clause with the options defined in Option-Setting.

**Restrictions:**

Delete FDT is not possible.

**FIND**    *select-clause*    *[option-setting]*

Finds the objects defined in the Select-Clause with the options defined in Option-Setting and writes a report of the objects found into a Natural text member stored in the Workplan library. In addition, a report of the objects found can be written into a specified report file as Natural text member in the Workplan library.

**FINDLIB**    *select-clause*    *[option-setting]*

Finds the libraries for Natural objects or Natural command processor sources defined in the Select-Clause with the options defined in Option-Setting and writes a report of the objects found into a Natural text member stored in the Workplan library. In addition, a report of the objects found can be written into a specified report file as Natural text member in the Workplan library.

**RESTART**    *[file-name]*

Continues an interrupted load function. This is only possible if information was written into a Natural text member in the Workplan library during the aborted load. See also RESTART under OPTION-Clause in Option-Setting (Direct and PROCEDURE Workplan Syntax, Commands).

# Direct and PROCEDURE Workplan Syntax

This section describes the syntax of Object Handler clauses that apply when executing Object Handler functions and maintaining Workplans. The symbols used in the diagrams are explained in the section System Command Syntax in the Natural System Command Reference documentation.

Depending on the type of Workplan (see the relevant section in Workplans), the following clauses apply:

The SELECTION Workplan contains the Select-Clause,  
LIST contains Object List,  
PARAMETER contains Parameter-Setting and  
PROCEDURE contains an Object Handler procedure.  
The TEXT Workplan only contains commentary text.

This section covers the following topics:

- Select-Clause
- Object List
- Parameter-Setting
- Option-Setting

# Object Handler - Select-Clause

The Select-Clause is used to define a Workplan of the type SELECTION or object selection criteria for the objects to be processed. For a description of the types of object, refer to the section Object Specification.

This section describes the syntax that applies to the Select-Clause.

**Note:**

\* The notation "see (\*)" in the tables below regarding name, date and time parameters refers to the section Name, Date and Time Specification.

The Select-Clause consists of one of the following:

- SELECTION Workplan
- Natural Object and DDM Selection
- Natural-related Debug Environment Selection
- Natural-related Profile Selection
- Natural-related DL/I Subfile Selection
- Natural System Error Selection
- Natural Command Processor Selection
- FDT Selection

**Natural DDM Selection:**

To select Natural DDMs (data definition modules), the same syntax applies as described for Natural Object Selection.

---

## SELECTION Workplan

A Workplan of the type SELECTION contains a selection from one of the following types of object or file: Natural objects, Natural-related objects, Natural system error messages, Natural command processor sources, DDMs or Adabas FDTs (Field Definition Tables). Alternatively, the Workplan can contain a selection list of objects as described in the section Object List. Such an object list can be used for the FIND and UNLOAD commands only.

For the syntax that applies to Workplans, see Referencing Workplans in the section Workplans.

## Natural Object and DDM Selection

This selection is used to select Natural objects for processing.

Natural DDMs (data definition modules) are processed like Natural objects, except for OBJTYPE, which must be set to **D**. However, as described below, not all of the keywords available apply to DDMs.

### Syntax

*object-name*

**LIBRARY** *library-name*

[ **DBID** *dbid* **FNR** *fnr* [**NAME** *vsam-name*] [**CIPHER** *cipher*] [ { **PASSWORD** **PSW** } *password* ] ]

[**OBJTYPE** *resource-type* ]

[**NATTYPE** *object-type* ]

[**SCKIND** *object-kind* ]

[**MODE** *object-mode* ]

[**FMNUM** *error-number-from* ]

[**TONUM** *error-number-to* ]

[**SLKIND** *message-type* ]

[**LANGUAGE** *languages* ]

[**DDMDBID** *dgm-dbid* ] [**DDMFNR** *dgm-fnr* ]

[**NATVERS** *natural-version* ]

[ { **DATE** *date* **FMDATE** *date-from* [**TODATE** *date-to* ] } ]

[ { **SIZE** *size* **FMSIZE** *size-from* [**TOSIZE** *size-to* ] } ]

[**USERID** *user-id* ]

[**TID** *terminal-id* ]

[**EXCEPT**- Clause]

**EXCEPT-Clause****EXCEPT***(object-name*[**LIBRARY** *library-name* ][**OBJTYPE** *resource-type* ][**SCKIND** *object-kind* ][**NATTYPE** *object-type* ][**MODE** *object-mode* ][**SLKIND** *message-type* ][**FMNUM** *error-number-from*] [**TONUM** *error-number-to* ][**LANGUAGE** *languages* ][**DDMDBID** *dsm-dbid* ] [**DDMFNR** *dsm-fnr* ][**NATVERS** *natural-version* ]

[	{	<b>DATE</b> <i>date</i>	}	]
		[ <b>FMDATE</b> <i>date-from</i> ] [ <b>TODATE</b> <i>date-to</i> ]		

[	{	<b>SIZE</b> <i>size</i>	}	]
		[ <b>FMSIZE</b> <i>size-from</i> ] [ <b>TOSIZE</b> <i>size-to</i> ]		

[**USERID** *user-id* ][**TID** *terminal-id* ]

)

**Note:**

For the command FINDLIB, only the following keywords are processed: **LIBRARY**, **DBID**, **FNR**, **NAME**, **CIPHER** and **PASSWORD/PSW**.

### Keyword Explanation - Natural Object Selection

The keywords and valid values for the objects to be processed are described below:

Keyword	Valid Values	Default Value
<i>object-name</i>	A valid Natural object name or a range of names. If <i>object-name</i> contains blank characters, it must be enclosed in double quotation marks (" ").  (* ) See also Name.	none
<u>LIBRARY</u>	A valid Natural library name or a range of names.  If OBJTYPE= <b>D</b> (see below), the library name is ignored.  (* ) See also Name.	none
DBID	Not valid for DDMs on mainframes (OBJTYPE <b>D</b> - see below).  A valid database ID (DBID).	0 (current FNAT/FUSER)
FNR	Not valid for DDMs on mainframes (OBJTYPE <b>D</b> - see below).  A valid file number (FNR).	0 (current FNAT/FUSER)
NAME	Only applies to objects on mainframes. Not valid for DDMs on mainframes (OBJTYPE <b>D</b> - see below). A valid VSAM name.	blank (current FNAT/FUSER)
CIPHER	Only applies to objects on mainframes. Not valid for DDMs on mainframes (OBJTYPE <b>D</b> - see below). The 8-digit cipher code of the Adabas file where the objects are stored.	blank (current FNAT/FUSER)
PASSWORD PSW	Only applies to objects on mainframes. An 8-character Adabas password.	blank (current FNAT/FUSER)
<u>OBJTYPE</u>	Types of object are:  <ul style="list-style-type: none"> <li><b>D</b> DDMs (objects on mainframes only)</li> <li><b>E</b> Natural error messages</li> <li><b>N</b> Natural programming objects</li> <li><b>*</b> Asterisk (all)</li> </ul> or a valid combination.  <b>Exception:</b> OBJTYPE <b>D</b> cannot be combined with any other type.	*
<u>NATTYPE</u>	Not applicable to OBJTYPE <b>D</b> (see above).  Up to 15 valid Natural object types, such as <b>P</b> for program and <b>V</b> for DDMs.  Asterisk (*) = all	*

Keyword	Valid Values	Default Value
<u>S</u> CKIND	<p>Not applicable if OBJTYPE <b>D</b> (see above).</p> <p>The kind of Natural programming objects. Valid letters:</p> <p><b>S</b> Source objects. If used in the EXCEPT-Clause: objects that are stored only in source form.</p> <p><b>C</b> Cataloged objects. If used in the EXCEPT-Clause: objects that are stored only in source form.</p> <p><b>A</b> All saved and cataloged objects.</p> <p><b>W</b> Source and cataloged objects if both have the same date and time (stowed).</p> <p><b>B</b> Source and cataloged objects if both exist.</p> <p><b>Note:</b> <b>W</b> (stowed) and <b>B</b> (both) are valid for the UNLOAD and FIND commands only. For LOAD and SCAN, <b>W</b> and <b>B</b> are valid entries, but they are treated like <b>A</b> (all objects). If data is processed in Transfer format, only <b>S</b> (source objects) and <b>A</b> applies.</p>	A
MODE	<p>Not applicable if OBJTYPE <b>D</b> (see above).</p> <p>The programming mode of the Natural programming objects. Valid letters:</p> <p><b>A</b> Any.</p> <p><b>R</b> All objects in reporting mode.</p> <p><b>S</b> All objects in structured mode.</p>	A
FMNUM	<p>A start number of Natural error messages. Valid range:1 to 9999.</p>	1
TONUM	<p>An end number of Natural error messages. Valid range: 1 to 9999. The value must be greater than or equal to the value of FMNUM, if specified.</p>	9999 or value of FMNUM (if specified)

Keyword	Valid Values	Default Value
<u>SLKIND</u>	The type of Natural error message text. Valid letters:  <b>S</b> Short text. Cannot be applied to the DELETE command (see Direct and PROCEDURE Workplan Commands).  <b>L</b> Long text.  <b>A</b> Short and long text, or the one that exists.  <b>B</b> Short and long text, but only if both exist.	A
<u>LANGUAGE</u>	Up to 8 valid language codes of Natural error messages. Asterisk (*) = all language codes.	*
<u>DDMDBID</u>	The valid database ID (1 to 65535) of a DDM. 0 = all database IDs (DBIDs).	0 (no check)
<u>DDMFNR</u>	The valid file number (1 to 65535) of a DDM. 0 = all file numbers (FNRs).	0 (no check)
<u>NATVERS</u>	The Natural version of Natural programming objects.  Valid version format: <i>VRSM</i> where <i>V</i> is the 1-digit version, <i>R</i> the 1-digit release, and <i>SM</i> the 2-digit system maintenance level. You can also specify a range of versions: see (*) Name.	blank (no check)
<u>DATE</u>	The save or catalog date of Natural programming objects. You can add a time by inserting a blank between date and time. For the format and ranges allowed, see (*) Date and (*) Time.  Special terms allowed are <u>YESTERDAY</u> and <u>TODAY</u> . See (*) Special Dates.	blank (no check)
<u>FMDATE</u>	A start value: The date on or after which Natural programming objects were cataloged or saved. The format is identical to DATE. See (*) Date.  Special terms allowed are <u>YEAR</u> , <u>MONTH</u> , <u>YESTERDAY</u> and <u>TODAY</u> . See (*) Special Dates.	blank (no check)
<u>TODATE</u>	An end value: The date on or before which Natural programming objects were cataloged or saved. The format is identical to DATE. See (*) Date.  Special terms allowed are <u>YEAR</u> , <u>MONTH</u> , <u>YESTERDAY</u> and <u>TODAY</u> . See (*) Special Dates.	blank (no check) or high value (if FMDATE specified)
<u>SIZE</u>	The size of Natural programming objects (up to 7 digits).	0 (no check)
<u>FMSIZE</u>	A start value: The minimum size of Natural programming objects (up to 7 digits).	0 (no check)

Keyword	Valid Values	Default Value
TOSIZE	An end value: The maximum size of Natural programming objects (up to 7 digits).	0 (no check) or high value (if FMSIZE specified)
USERID	The ID of the user who saved or cataloged the Natural programming objects  You can also specify a range of user IDs: see (*) Name.	blank (no check)
TID	Not applicable if OBJTYPE=D (see above).  The ID of the terminal where the Natural programming objects were saved or cataloged (provided by the Natural system variable *INIT-ID).  You can also specify a range of terminal IDs: see also (*) Name.	blank (no check)
EXCEPT	All objects that match the selection criteria entered before EXCEPT are checked against <b>all</b> parameters contained within the parentheses following the keyword EXCEPT. If they match all these parameters too, they are not processed.	not applicable

**Note:**

- Parameters that are irrelevant for OBJTYPE are ignored. For example: DATE, SIZE and USERID have no meaning for Natural error messages.
- DBID, FNR, NAME, CIPHER and PASSWORD/PSW are ignored by the LOAD and SCAN commands. These parameters must instead be specified in the Parameter-Setting clause as described for LOADFNAT... and LOADFUSER... in Keyword Explanation - Parameter-Setting.

## Natural-related Debug Environment Selection

This selection is used to select Natural-related debug environments for processing.

### Syntax

```

object-name

NATPATH DEBUG
[LIBRARY library-name]
[DBID dbid [FNR fnr] ] [NAME vsam-name]
[CIPHER cipher]
[
  {
    PASSWORD
    PSW
  } password
]
[ EXCEPT ( object-name [LIBRARY library-name ] ) ]
    
```

**Keyword Explanation - Natural-related Debug Environment Selection**

The keywords and valid values for the objects to be processed are described below:

<b>Keyword</b>	<b>Valid Values</b>	<b>Default Value</b>
<i>object-name</i>	A valid object name or a range of names. See also (*) Name.	none
LIBRARY	A valid Natural library name or a range of names. (*) See also Name.	none
DBID	A valid database ID (DBID).	0 (current FUSER)
FNR	A valid file number (FNR).	0 (current FUSER)
NAME	A valid VSAM name.	blank (current FUSER)
CIPHER	The 8-digit cipher code of the Adabas file where the objects are stored.	blank (current FUSER)
PASSWORD	An 8-character Adabas password.	blank (current FUSER)
PSW		
EXCEPT	See EXCEPT under Natural Object Selection.	not applicable

**Note:**

DBID, FNR, NAME, CIPHER and PASSWORD/PSW are ignored by the LOAD and SCAN commands. These parameters must instead be specified in the Parameter-Setting clause as described for LOADFNAT... and LOADFUSER... in Keyword Explanation - Parameter-Setting.

## Natural-related Profile Selection

This selection is used to select Natural-related profiles for processing.

### Syntax

<p><i>object-name</i></p> <p><b>NATPATH PROFILE</b></p> <p><b>[OBJTYPE <i>profile-type</i> ] [DBID <i>dbid</i> [FNR <i>fnr</i> ] ]</b></p> <p><b>[NAME <i>vsam-name</i> ] [CIPHER <i>cipher</i>]</b></p> <p><b>[ { <b>PASSWORD</b> <b>PSW</b> } <i>password</i> ]</b></p> <p><b>[ EXCEPT ( <i>object-name</i> [OBJTYPE <i>profile-type</i> ] ) ]</b></p>
--

**Keyword Explanation - Natural-related Profile Selection**

The keywords and valid values for the objects to be processed are described below:

<b>Keyword</b>	<b>Valid Values</b>	<b>Default Value</b>
<i>object-name</i>	A valid object name or a range of names.  See also (*) Name.	none
<b>OBJTYPE</b>	The type of profile:  <b>D</b> Device profile <b>E</b> Editor profile <b>M</b> Map profile <b>P</b> Parameter profile * Asterisk (all profile types)  or any combination.	*
<b>DBID</b>	A valid database ID (DBID).	0 (current FNAT)
<b>FNR</b>	A valid file number (FNR).	0 (current FNAT)
<b>NAME</b>	A valid VSAM name.	blank (current FNAT)
<b>CIPHER</b>	The 8-digit cipher code of the Adabas file where the objects are stored.	blank (current FNAT)
<b>PASSWORD</b>	An 8-character Adabas password.	blank (current FNAT)
<b>PSW</b>		
<b>EXCEPT</b>	See EXCEPT under Natural Object Selection.	not applicable

**Note:**

DBID, FNR, NAME, CIPHER and PASSWORD/PSW are ignored by the LOAD and SCAN commands. These parameters must instead be specified in the Parameter-Setting clause as described for LOADFNAT... and LOADFUSER... in Keyword Explanation - Parameter-Setting.

**Natural-related DL/I Subfile Selection**

This selection is used to select Natural-related DL/I subfiles for processing.

**Syntax**

```

object-name

NATPATH SUBFILE

[OBJTYPE subfile-type] [DBID dbid [FNR fnr ]
[NAME vsam-name] [CIPHER cipher]

[
{
    PASSWORD
    PSW
}
    password
]

[ EXCEPT ( object-name [OBJTYPE subfile-type ] ) ]

```

### Keyword Explanation - Natural-related DL/I Selection

The keywords and valid values for the objects to be processed are described below:

Keyword	Valid Values	Default Value
<i>object-name</i>	A valid object name or a range of names. See also (*) Name.	none
<b>OBJTYPE</b>	The type of DL/I subfile:  <b>D</b> NDB <b>P</b> NSB *   Asterisk (both subfile types)	*
DBID	A valid database ID (DBID).	0 (current FDIC)
FNR	A valid file number (FNR).	0 (current FDIC)
NAME	A valid VSAM name.	blank (current FDIC)
CIPHER	The 8-digit cipher code of the Adabas file where the objects are stored.	blank (current FDIC)
PASSWORD	An 8-character Adabas password.	blank (current FDIC)
PSW		
EXCEPT	See EXCEPT under Natural Object Selection.	not applicable

#### Note:

DBID, FNR, NAME, CIPHER and PASSWORD/PSW are ignored by the LOAD and SCAN commands. These parameters must instead be specified in the Parameter-Setting clause as described for LOADFNAT... and LOADFUSER... in Keyword Explanation - Parameter-Setting.

# Natural System Error Selection

This selection is used to select Natural system error messages for processing.

## Syntax

```

ERROR NATERROR
    [
        DBID dbid FNR fnr [NAME vsam-name] [CIPHER cipher]
        [ { PASSWORD } password ]
        [ PSW password ]
        [FMNUM error-number-from] [TONUM error-number-to]
        [SLKIND message-type]
        [LANGUAGE languages]
        [
            EXCEPT (
                [FMNUM error-number-from] [TONUM error-number-to]
                [SLKIND message-type]
                [LANGUAGE languages]
            )
        ]
    ]
    
```

## Keyword Explanation - Natural System Error Selection

The keywords and valid values for the Natural system error messages to be processed are described below:

Keyword	Valid Values	Default Value
<b>DBID</b>	Only applies to objects on mainframes. A valid database ID (DBID).	0 (current FNAT)
<b>FNR</b>	Only applies to objects on mainframes. A valid file number (FNR).	0 (current FNAT)
<b>NAME</b>	Only applies to objects on mainframes. A valid VSAM name.	blank (current FNAT)
<b>CIPHER</b>	Only applies to objects on mainframes. The 8-digit cipher code of the Adabas file where the Natural system error messages are stored.	blank (current FNAT)
<b>PASSWORD</b>	Only applies to objects on mainframes. An 8-character Adabas password.	blank (current FNAT)
<b>PSW</b>		
<b>FMNUM</b>	A start number of Natural system error messages. Valid range: 1 to 9999.	1
<b>TONUM</b>	An end number of Natural system error messages. Valid range: 1 to 9999. The value must be greater than or equal to the value of FMNUM if specified.	9999 or value of FMNUM (if specified)
<b>SLKIND</b>	See SLKIND under Natural Object Selection.	A
<b>LANGUAGE</b>	Up to 8 valid language codes of Natural system error messages. Asterisk (*) = all language codes.	*
<b>EXCEPT</b>	See EXCEPT under Natural Object Selection.	

**Note:**

DBID, FNR, NAME, CIPHER and PASSWORD/PSW are ignored by the LOAD and SCAN commands. These parameters must instead be specified in the Parameter-Setting clause as described for LOADFNAT... in Keyword Explanation - Parameter-Setting.

## Natural Command Processor Selection

This selection is used to select Natural command processor sources for processing.

### Syntax

<i>object-name</i> <b>PROCESSOR</b> <i>ncp-library-name</i>																									
[	<table border="0"> <tr> <td><b>DBID</b></td> <td><i>ncp-dbid</i> <b>FNR</b> <i>ncp-fnr</i></td> <td>[</td> <td>{</td> <td><b>PASSWORD</b></td> <td>}</td> <td><i>ncp-password</i></td> <td>]</td> </tr> <tr> <td></td> <td>[<b>NAME</b> <i>ncp-vsam-name</i> ]</td> <td></td> <td></td> <td><b>PSW</b></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td>[<b>CIPHER</b> <i>ncp-cipher</i> ]</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </table>	<b>DBID</b>	<i>ncp-dbid</i> <b>FNR</b> <i>ncp-fnr</i>	[	{	<b>PASSWORD</b>	}	<i>ncp-password</i>	]		[ <b>NAME</b> <i>ncp-vsam-name</i> ]			<b>PSW</b>					[ <b>CIPHER</b> <i>ncp-cipher</i> ]						
<b>DBID</b>	<i>ncp-dbid</i> <b>FNR</b> <i>ncp-fnr</i>	[	{	<b>PASSWORD</b>	}	<i>ncp-password</i>	]																		
	[ <b>NAME</b> <i>ncp-vsam-name</i> ]			<b>PSW</b>																					
	[ <b>CIPHER</b> <i>ncp-cipher</i> ]																								
[	<table border="0"> <tr> <td><b>EXCEPT</b></td> <td>(</td> </tr> <tr> <td></td> <td><i>object-name</i></td> </tr> <tr> <td></td> <td>[<b>LIBRARY</b> <i>ncp-library-name</i>]</td> </tr> <tr> <td></td> <td>)</td> </tr> </table>	<b>EXCEPT</b>	(		<i>object-name</i>		[ <b>LIBRARY</b> <i>ncp-library-name</i> ]		)																
<b>EXCEPT</b>	(																								
	<i>object-name</i>																								
	[ <b>LIBRARY</b> <i>ncp-library-name</i> ]																								
	)																								

**Note:**

For the command FINDLIB only the following keywords are processed: PROCESSOR, DBID, FNR, NAME, CIPHER and {PASSWORD/PSW}.

### Keyword Explanation - Natural Command Processor Selection

The keywords and valid values for the Natural command processor sources to be processed are described below:

Keyword	Valid Values	Default Value
<i>object-name</i>	The name of a valid Natural command processor source or a range of names.  See also (*) Name.	none
<b>PROCESSOR</b>	A valid Natural library name or a range of names.  See also (*) Name.	none
<b>DBID</b>	The valid database ID (DBID) of the Adabas file where the Natural command processor sources are stored.	0 (current FNAT/FUSER)
<b>FNR</b>	The valid file number (FNR) of the Adabas file where the Natural command processor sources are stored.	0 (current FNAT/FUSER)
<b>NAME</b>	Only applies to objects on mainframes. A valid VSAM name.	blank
<b>CIPHER</b>	The 8-digit cipher code of the Adabas file where the Natural command processor sources are stored.	blank
<b>PASSWORD</b> <b>PSW</b>	The 8-character Adabas password of the Adabas file where the Natural command processor sources are stored.	blank
<b>EXCEPT</b>	See EXCEPT under Natural Object Specification.	

**Note:**

DBID, FNR, NAME, CIPHER and PASSWORD/PSW are ignored by the LOAD and SCAN commands. These parameters must instead be specified in the Parameter-Setting clause as described for LOADNCP... in Keyword Explanation - Parameter-Setting.

## FDT Selection

This selection is used to select Adabas FDTs (Field Definition Tables) for processing.

For loading FDTs, see also FDTs in the section Object Specification.

The following syntax applies when processing FDTs:

### Syntax

```

FDT
DBID dbid
    {
        FNR fnr [CIPHER cipher]
        FMFNR fnr-start TOFNR fnr-end
    }
    [
        {
            PASSWORD
            PSW
        } password
    ]

```

**Keyword Explanation - FDT Selection**

The keywords and valid values for the FDTs to be processed are described below:

<b>Keyword</b>	<b>Valid Values</b>	<b>Default Value</b>
DBID	The database ID (DBID) of the FDT.	none
FNR	The file number (FNR) of the FDT.	none
CIPHER	The 8-digit Adabas cipher code of the FDT.	none
PASSWORD	The 8-character Adabas password of the FDT.	none
PSW		
FMFNR	Only applies to the FIND or UNLOAD command.  A start value: The file number (FNR) of an FDT.	none
TOFNR	Only applies to the FIND or UNLOAD command.  An end value: The file number (FNR) of an FDT.	none

# Object Handler - Object List

An object list consists of object specifications which are to be processed by the Object Handler.

The following syntax applies to an object list:

## TYPE LIST

```
[ object-type-and-location ( object-name-description ... ) ] ...
```

### Note:

Each item (except for the ones enclosed in parentheses) must start on a new line and end on the same line.

This section covers the following topics:

- Syntax of Object-type-and-location
- Syntax of Object-name-description

## Syntax of Object-type-and-location

### Natural Objects and DDMs

#### **LIBRARY** *library-name*

```
[
  DBID dbid FNR fnr [NAME vsam-name ] [CIPHER cipher ] [ { PASSWORD
  PSW } password ] ]
  [OBJTYPE resource-type ]
```

### Note:

For DDMs on mainframe platforms, OBJTYPE must be set to **D**.

No ranges are allowed for *library-name*.

### Natural System Error Messages

#### **ERROR NATERROR**

```
[
  DBID dbid FNR fnr [NAME vsam-name ] [CIPHER cipher ] [ { PASSWORD
  PSW } password ] ]
```

### Natural Command Processor Sources

#### **PROCESSOR** *ncp-library-name*

```
[
  DBID dbid FNR fnr [NAME vsam-name ] [CIPHER cipher ] [ { PASSWORD
  PSW } password ] ]
```

**Note:**

No ranges are allowed for *ncp-library-name*.

**Natural-related Debug Environment Objects**

<p><b>NATPATH DEBUG</b>  <b>LIBRARY</b> <i>library-name</i></p> <p>[ [DBID <i>dbid</i> [FNR <i>fnr</i>] ] [NAME <i>vsam-name</i>] [CIPHER <i>cipher</i>] [ { PASSWORD PSW } <i>password</i> ] ] ]</p>
---

**Natural-related Profile Objects**

<p><b>NATPATH PROFILE</b></p> <p>[ [DBID <i>dbid</i> [FNR <i>fnr</i>] ] [NAME <i>vsam-name</i>] [CIPHER <i>cipher</i>] [ { PASSWORD PSW } <i>password</i> ] ] ]</p>
---

**Natural-related DL/I Subfile Objects**

<p><b>NATPATH SUBFILE</b></p> <p>[ [DBID <i>dbid</i> [FNR <i>fnr</i>] ] [NAME <i>vsam-name</i>] [CIPHER <i>cipher</i>] [ { PASSWORD PSW } <i>password</i> ] ] ]</p>
---

**FDTs**

<b>FDT</b>
------------

**Syntax of Object-Name-Description****Natural Objects**

<p>{ <i>object-name</i> [SCKIND <i>object-kind</i> ]</p> <p>{ <i>error-number</i> [SLKIND <i>message-type</i>] [LANGUAGE <i>languages</i>]  FMNUM <i>error-number-from</i> TONUM <i>error-number-to</i> [SLKIND <i>message-type</i>] [LANGUAGE <i>languages</i>]</p>
--

**DDMs**

<i>object-name</i>
--------------------

**Natural System Error Messages**

$\left\{ \begin{array}{l} \text{error-number } [\underline{\text{SLKIND}} \text{ message-type}] \underline{\text{LANGUAGE}} \text{ languages} \\ \text{FMNUM error-number-from TONUM error-number-to } [\underline{\text{SLKIND}} \text{ message-type}] [\underline{\text{LANGUAGE}} \text{ languages}] \end{array} \right\}$
---

**Natural Command Processor Sources**

$object\text{-name}$
----------------------

**Natural-related Debug Environment Objects**

$object\text{-name}$
----------------------

**Natural-related Profile Objects**

$object\text{-name } [\underline{\text{OBJTYPE}} \text{ profile-type } ]$
---

**Natural-related DL/I Objects**

$object\text{-name } [\underline{\text{OBJTYPE}} \text{ subfile-type } ]$
---

**FDTs**

$\left[ \text{DBID } dbid \text{ FNR } fnr [\underline{\text{CIPHER}} \text{ cipher } ] \left[ \left\{ \begin{array}{l} \text{PASSWORD} \\ \text{PSW} \end{array} \right\} \text{ password } \right] \right]$
---

**Example:**

```
TYPE LIST
LIBRARY LIB-1 OBJTYPE N      /* process Natural objects from library 'LIB-1'
( A* SCKIND S                /* all sources whose names start with 'A'
B1                            /* source and/or cataloged of object 'B1'
CDE> SCKIND C )             /* all cataloged whose names are greater/equal 'CDE'
/*                            /* Comment line
LIBRARY LIB-2                /* process Natural objects from library 'LIB-2'
/*                            /* including error messages and shared resources
( *                           /* source and/or cataloged of all objects
/*                            /* including shared resources
FMNUM 1 TONUM 100           /* error messages from 1 to 100
)
```

# Object Handler - Parameter-Setting

The Parameter-Setting clause is used to change attributes for the LOAD or UNLOAD command for the objects to be processed and to define target destinations for the LOAD command (for example, FNAT).

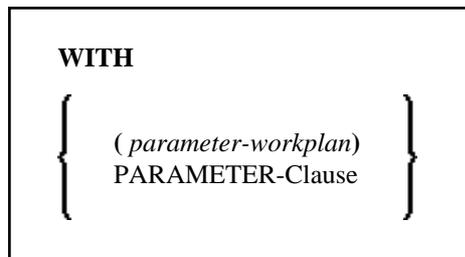
**Note:**

\* The notation "see (\*)" in the tables below regarding name, date and time parameters refers to the section Name, Date and Time Specification.

This section covers the following topic:

- Syntax of Parameter-Setting
  - Keyword Explanation - Parameter-Setting
- 

## Syntax of Parameter-Setting



## PARAMETER-Clause

[[**NAME** *old-name*] **NEWNAME** *new-name*]

[[**LIBRARY** *old-library-name*] **NEWLIBRARY** *new-library-name*]

[**LOADFNATDBID** *fnat-dbid* **LOADFNATFNR** *fnat-fnr* [**LOADFNATNAME** *vsam-name*]

[**LOADFNATCIPHER** *fnat-cipher*]

[ { **LOADFNATPASSWORD**  
**LOADFNATPSW** } *fnat-password* ] ]

[**LOADFUSERDBID** *fuser-dbid* **LOADFUSERFNR** *fuser-fnr*

[**LOADFUSERNAME** *fuser-vsam-name*] [**LOADFUSERCIPHER** *fuser-cipher*]

[ { **LOADFUSERPASSWORD**  
**LOADFUSERPSW** } *fuser-password* ] ]

[**LOADNCPDBID** *ncp-file-dbid* **LOADNCPFNR** *ncp-file-fnr*

[**LOADNCPNAME** *ncp-file-vsam-name*] [**LOADNCPCIPHER** *ncp-file-cipher*]

[ { **LOADNCPPASSWORD**  
**LOADNCPPSW** } *ncp-file-password* ] ]

[[**FDTDBID** *old-fdt-dbid* **FDTFNR** *old-fdt-fnr*] **NEWFDTDBID** *new-fdt-dbid* **NEWFDTFNR** *new-fdt-fnr* ]

[**ERRNUMDIFF** *modification-of-error-message-range*]

[[**LANGUAGE** *old-language*] **NEWLANGUAGE** *new-language*]

[[**DATE** *old-date*] **NEWDATE** *new-date*]

[[**USERID** *old-userid*] **NEWUSERID** *new-userid*]

[[**TID** *old-terminal-id*] **NEWTID** *new-terminal-id*]

[[**PATH** *old-external-path-name*] **NEWPATH** *new-external-path-name*]

## Keyword Explanation - Parameter-Setting

Keyword	Values	Restricted to Command
NAME	The name to be checked if NEWNAME is specified.	
NEWNAME	A new object name.	
LIBRARY	The name to be checked if NEWLIBRARY is specified.	
NEWLIBRARY	A new library name.	
LOADFNATDBID	The database ID (DBID) of FNAT libraries.	LOAD
LOADFNATFNR	The file number (FNR) of FNAT libraries.	LOAD
LOADFNATNAME	Only applies to objects on mainframes. An FNAT VSAM file name.	LOAD
LOADFNATCIPHER	Only applies to objects on mainframes. An FNAT cipher code.	LOAD
LOADFNATPASSWORD	Only applies to objects on mainframes. An FNAT Adabas password.	LOAD
LOADFUSERDBID	The DBID of FUSER libraries.	LOAD
LOADFUSERFNR	The FNR of FUSER libraries.	LOAD
LOADFUSERNAME	Only applies to objects on mainframes. An FUSER VSAM file name.	LOAD
LOADFUSERCIPHER	Only applies to objects on mainframes. An FUSER cipher code.	LOAD
LOADFUSERPASSWORD	Only applies to objects on mainframes. An FUSER Adabas password.	LOAD
LOADNCPDBID	The DBID of the Adabas file for Natural command processor sources.	LOAD
LOADNCPFNR	The FNR of the Adabas file for Natural command processor sources.	LOAD
LOADNCPNAME	Only applies to objects on mainframes. The VSAM name of the Adabas file for Natural command processor sources.	LOAD
LOADNCPCIPHER	Only applies to objects on mainframes. The cipher code of the Adabas file for Natural command processor sources.	LOAD
LOADNCPPASSWORD	Only applies to objects on mainframes. The Adabas password of the Adabas file for Natural command processor sources.	LOAD
FDTDBID	The DBID of the Adabas FDT ((Field Definition Table) to be checked if NEWFDTDBID is specified.	
NEWFDTDBID	A new DBID of the FDT.	
FDTFNR	The DBID of the FDT to be checked if NEWFDTFNR is specified.	
NEWFDTFNR	A new FNR of the FDT.	

Keyword	Values	Restricted to Command
ERRNUMDIFF	A number (positive or negative) that is to be added to the Natural error messages during the UNLOAD or LOAD command.  ERRNUMDIFF can only be specified if FMNUM and TONUM (see the section Select-Clause) have been specified as selection criteria. Otherwise, it is not possible to check for valid results.	
<u>LANGUAGE</u>	Up to 8 valid language codes of Natural error messages to be checked if NEWLANGUAGE (see below) is specified.  If <i>language</i> contains more than one language code, <i>new-language</i> must contain the same numbers of language codes. Each <i>language</i> language code is replaced by the language code in the corresponding position of <i>new-language</i> . If <i>language</i> is not specified, <i>new-language</i> must not contain more than one language code.	
<u>NEWLANGUAGE</u>	Up to 8 valid language codes for new Natural error messages.  See also LANGUAGE above.	
DATE	An object date. You can add a time by inserting a blank between date and time. For the format and ranges allowed, see (*) Date and (*) Time.	
NEWDATE	A new object date. NEWDATE can be a date followed by a time value. You can add a time by inserting a blank between date and time. See also (*) Date and (*) Time.	
<u>USERID</u>	The user ID to be checked if NEWUSERID is specified.	
<u>NEWUSERID</u>	A new user ID.	
TID	Only applies to objects on mainframes. The terminal ID to be checked if NEWTID is specified.	
NEWTID	Only applies to objects on mainframes. A new terminal ID.	
PATH	The path name to be checked if NEWPATH is specified.	
NEWPATH	A new path name.	

**Note:**

- Parameters not applicable to the selection criterion processed are ignored.
- LOADFNAT, LOADFUSER... and LOADNCP... are used for the LOAD command only, and ignored otherwise.
- LOADFNAT... is used for libraries starting with SYS (except SYSTEM).
- LOADFUSER... is used for libraries not starting with SYS (but including SYSTEM).
- LOADNCP... is used for Natural command processor sources.

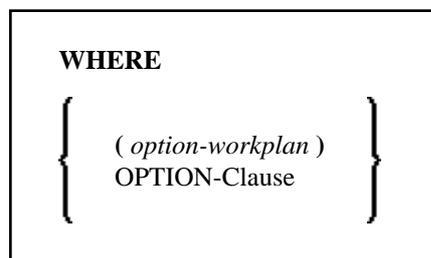
# Object Handler - Option-Setting

The Option-Setting clause is used to change the default values of Object Handler command options.

This section covers the following topics:

- Syntax of Option-Setting
  - Keyword Explanation - Option-Setting
- 

## Syntax of Option-Setting





## Keyword Explanation - Option-Setting

Option	Explanation	Restricted to Command
<u>REPLACE</u>	Replaces existing objects according to the option specified:  ALL            All objects (default).  OBSOLETE    All objects with a date older than the date of the object in the load file.  EXCEPT     All objects except those with a date newer than the date of the object in the load file.	LOAD LOADALL
<u>NOREPORT</u>	Specifies the report file setting: No data is recorded into a report file. This is the default setting for the FIND and FINDLIB commands.	
<u>NEWREPORT</u>	Specifies the report file setting: Report data is recorded and written into a Natural text member stored in the Workplan library. An existing file will be overwritten.	
<u>REPORT</u>	Specifies the report file setting: Report data is recorded and written into a Natural text member stored in the Workplan library. This is the default setting for the commands UNLOAD, LOAD, LOADALL, SCAN, SCANALL and DELETE.	
<u>NORESTART</u>	No restart information is written into a file.	LOAD
<u>RESTART</u>	Restart information is written into a Natural text member stored in the Workplan library.	LOAD
<u>TRANSFER</u>	Set Transfer mode. The data is read and written in Transfer format.	UNLOAD LOAD SCAN
<u>NUMBERPROCESS</u>	Specifies the number of objects to be processed. The LOAD or SCAN command stops execution after the number specified.	LOAD SCAN
<u>FIXEDLENGTH</u>	Sets the format of the unload work file to a maximum record length of fixed size. Every data record contains 256 bytes if written in internal format, or 100 bytes in Transfer format.	UNLOAD

Option	Explanation	Restricted to Command
<u>CONVERSION-TABLE</u>	<p>Converts data processed in Transfer format by using either of the following conversion tables:</p> <p><u>SYSTEM-TABLE</u>:</p> <p>The internal Natural conversion table.</p> <p><u>USER-TABLE</u>:</p> <p>A user-defined conversion table if <i>conversion-program</i> has been specified. This program must be stored in the library SYSOBJH or one of its steplib; see the example programs OTNCONAE and OTNCONEA in the library SYSOBJH.</p>	UNLOAD LOAD SCAN
<u>SUBSTITUTE</u>	<p>Replaces line references by labels during the unload in Transfer format.</p> <p>This option only applies if your source-code line numbers are used for statement references. If so, the line numbers of referenced lines and the line number references are replaced by labels. The sources are not modified in the database.</p>	UNLOAD
<u>INCLUDE-LINE-NUMBERS</u>	<p>Transfers line numbers during the unload in Transfer format. By default, line numbers in Natural objects are <b>not</b> unloaded.</p>	UNLOAD
<u>UPPERCASE-TRANSLATION</u>	<p>Translates any source code into upper case during the load in Transfer format.</p> <p>By default, source code in Natural objects is <b>not</b> translated.</p>	LOAD
<u>INCORPORATE-FREE-RULES</u>	<p>Incorporates source text of Predict free rules associated with a map into a map source during the unload in Transfer format if Predict is installed.</p>	UNLOAD
<u>DA-FORMAT</u>	<p>Specifies format conversion of data area sources: see Data area format in Transfer Options, Set Additional Options, in the section Settings.</p>	UNLOAD LOAD
XREF	<p>Loads or unloads XREF data of cataloged Natural objects: see XREF below.</p>	UNLOAD LOAD
VERSIONCHECK	<p>Checks the Natural version of the cataloged object to be loaded.</p> <p>See also Version Check in Set Additional Options in the section Settings.</p>	LOAD
FDIC	<p>Specifies the Natural system file FDIC to be used for processing.</p>	UNLOAD LOAD DELETE
FSEC	<p>Specifies the Natural system file FSEC to be used.</p>	UNLOAD LOAD DELETE

Option	Explanation	Restricted to Command
<u>PC</u>	<p>Only applies if Entire Connection is installed.</p> <p>Writes data to or reads data from an Entire Connection work file.</p> <p>If you do not specify <i>file-name</i>, Entire Connection prompts you for the name of a work file.</p> <p>See also Work File Assignment in the section Work Files.</p>	UNLOAD LOAD SCAN

The section below contains further information on:

- XREF

## XREF

Only applies if Predict is installed.

Loads or unloads XREF data of cataloged Natural objects.

The following values can be specified:

Value	Explanation
ON	UNLOAD: Unloads cataloged objects and their cross-reference data, if any.  LOAD: Loads cataloged objects and their cross-reference data if cross-references exist in the work file.
OFF	No XREF data is processed. This is the default.
DOC	Only applies to LOAD.  Loads cataloged objects only if Predict entries exist for the objects in the FDIC system file.
FORCE	Only applies to LOAD.  Loads cataloged objects and their cross-reference data only if cross-references exist in the work file and if Predict entries exist for the objects in the FDIC system file.
SPECIAL	Only applies to LOAD.  Loads cataloged objects and their cross-reference data (if any).

# Object Handler Commands for CUIs

The Object Handler commands provided in CUI (character user interface) environments are mainly provided for navigation purpose and special settings, such as specifying trace files.

An Object Handler commands is entered in the command line of any Object Handler screen. To execute a Natural system command, enter two slashes (//) before the command. Note that any Natural system command terminates the Object Handler.

## To invoke the Commands menu of the Object Handler

- Choose PF10/Cmds.

The internal commands are listed below. An underlined portion of a keyword represents an acceptable abbreviation, Sub denotes subcommand.

Command	Sub 1	Sub 2	Explanation
CANCEL			Cancels the current function and returns to the Natural Object Handler Main Menu.
<u>C</u> HANGE	<u>W</u> ORKPLAN	<u>L</u> IBRARY	Go to the Change Workplan Library screen.
<u>C</u> LEAR			Reset the current contents of the input fields in the map to the default values.
CMD COMMAND			Invoke the Commands screen.
BYE			Leave the Object Handler.
EXIT			
QUIT			
.			
FIN			Leave the Object Handler and end the Natural session.
<u>G</u> O	<u>H</u> OME		Go to the Natural Object Handler Main Menu.
<u>G</u> O	<u>U</u> NLOAD		Go to the unload screen.
<u>G</u> O	<u>U</u> NLOAD	<u>E</u> ND	End the current unload function.
		<u>E</u> RROR	Go to the Unload Natural System Error Messages screen.
		<u>D</u> DM	Go to the Unload DDMs screen.
		<u>F</u> DT	Go to the Unload FDTs screen.
		<u>L</u> IBRARY	Go to the Unload Natural Library Objects screen.
		<u>N</u> CP	Go to the Unload Natural Command Processors screen.
		<u>R</u> ELATED	Go to the Unload Natural-related Objects screen.
		<u>S</u> ELECTION <u>L</u> IST	Go to the Unload Selection or List Workplan screen.
<u>G</u> O	<u>L</u> OAD		Go to the load screen.
<u>G</u> O	<u>L</u> OAD		Go to the load screen.

Command	Sub 1	Sub 2	Explanation
<u>GO</u>	<u>LOAD</u>	<u>ALL</u>	Go to the Load all Objects screen.
		<u>END</u>	End the current load function.
		<u>ERROR</u>	Go to the Load Natural System Error Messages screen.
		<u>DDM</u>	Go to the Load DDMs screen.
		<u>FDT</u>	Go to the Load FDTs screen.
		<u>LIBRARY</u>	Go to the Load Natural Library Objects screen.
		<u>NCP</u>	Go to the Load Natural Command Processors screen.
		<u>RELATED</u>	Go to the Load Natural-related Objects screen.
		<u>SELECTION LIST</u>	Go to the Load Selection or List Workplan screen.
<u>GO</u>	<u>RESTART</u>		Start the Restart Options screen.
<u>GO</u>	<u>SCAN</u>		Go to the scan screen.
<u>GO</u>	<u>SCAN</u>	<u>ALL</u>	Go to the Scan all Objects screen.
		<u>END</u>	End the current scan function.
		<u>ERROR</u>	Go to the Scan Natural System Error Messages screen.
		<u>DDM</u>	Go to the Scan DDMs screen.
		<u>FDT</u>	Go to the Scan FDTs screen.
		<u>LIBRARY</u>	Go to the Scan Natural Library Objects screen.
		<u>NCP</u>	Go to the Scan Natural Command Processors screen.
		<u>RELATED</u>	Go to the Scan Natural-related Objects screen.
		<u>SELECTION LIST</u>	Go to the Scan Selection or List Workplan screen.
<u>GO</u>	<u>ADMIN</u>		Go to the Administration screen.
<u>GO</u>	<u>ADMIN</u>	<u>CHANGE</u>	Go to the "Change the Workplan Library" administration function.
		<u>CREATE</u>	Go to the "Create a new Workplan" administration function.
		<u>LIST</u>	Go to the List Workplans administration function.
<u>GO</u>	<u>VIEW</u>		Go to the View menu.
<u>GO</u>	<u>VIEW</u>	<u>DDM</u>	Go to the View DDMs screen.
		<u>ERROR</u>	Go to the View Natural System Error Messages screen.
		<u>FDT</u>	Go to the View FDTs screen.
		<u>LIBRARY</u>	Go to the View Natural Library Objects screen.
		<u>NCP</u>	Go to the View Natural Command Processors screen.
		<u>RELATED</u>	Go to the Select Natural-Related Type screen.
<u>GO</u>	<u>FIND</u>		Go to the Find screen.

Command	Sub 1	Sub 2	Explanation
<u>G</u> O	<u>F</u> IND	<u>D</u> DM	Go to the Find DDMs screen.
		<u>E</u> RROR	Go to the Find Natural System Error Messages screen.
		<u>F</u> DT	Go to the Find FDTs screen.
		<u>L</u> IBRARY	Go to the Find Natural Library Objects screen.
		<u>N</u> CP	Go to the Find Natural Command Processors screen.
		<u>R</u> ELATED	Go to the Select Natural-Related Type screen.
		<u>S</u> ELECTION <u>L</u> IST	Go to the Find Selection or List Workplan screen.
<u>H</u> ELP			Invoke the Object Handler Help function.
INIT			Reinitialize the Object Handler utility.
READ	<u>P</u> ROFILE		Read the data from the SYSOBJH profile.
SET	<u>A</u> DVANCEDCMD	ON	Activate the display of commands generated by the Object Handler in advanced mode.
		OFF	Deactivate the display of commands generated by the Object Handler in advanced mode.
	<u>E</u> XECUTIONMSG	ON	Activate the window that displays the processing status.
		OFF	Deactivate the window that displays the processing status.
	FREE	ON	Activate free format editing.
		OFF	Deactivate free format editing.
	TRACE	ON	Activate the trace mode (trace output to screen).
		OFF	Deactivate the trace mode.
		<u>W</u> ORKFILE	Activate the trace mode (trace output to the Natural text member stored in the Workplan library).
	<u>T</u> RACEFILE		
SETTINGS			Display or change the unload, load or scan settings.
<u>S</u> HOW <u>D</u> ISPLAY	<u>L</u> AST	<u>M</u> ESSAGE	Display the last interface return code and message issued by the processing interface of the Object Handler.
		<u>R</u> ESULT	Display the last result issued by the processing interface of the Object Handler.
	<u>P</u> ROFILE		Display the text object PROFILES (if available).
	<u>R</u> EPORT		Display the report created last.
	<u>S</u> TATISTICS		Display statistics information about the objects processed.
	<u>S</u> TATUS		Display the current Object Handler status (contents of global variables).
	TRACE	FILE	

# Batch Condition Codes and User Exit Routines

This section provides a description of condition codes and user exit routines available for processing Object Handler functions in batch mode.

The sources of Object Handler user exit routines are named SRC-EX $nn$ , where  $nn$  denotes the number of the user exit routine. The user exit routines are delivered in the Natural system library SYSOBJH.

## To make a user exit routine available

- Catalog it under the name OBJHEX $nn$  in the Natural system library SYSOBJH.  
The name of each user exit source is different from the name of the corresponding cataloged object. This guarantees that the object is not affected if the user exit source is overwritten by an installation update.

Object Handler processing in batch mode terminates with one of the following condition codes:

Condition Code	Explanation
0	Object Handler process terminated successfully.
30	An internal Object Handler error occurred.
40	An error was detected in the Object Handler command.
50	An error occurred during Object Handler processing.
60	A Natural Security error occurred during Object Handler processing.
99	A Natural error occurred during Object Handler processing.

Whenever a condition code is set to a value greater than 0 (zero), if available, the user exit routine OBJHEX01 will be invoked before the Object Handler stops processing. With this user exit routine, you can specify whether to continue or terminate Object Handler processing. In the case of termination, you can change the condition code. For further details, see the source of the user exit routine SRC-EX01 in the Natural system library SYSOBJH.

If the Object Handler load function was executed successfully (with Condition Code 0), but one or more objects were rejected during loading (for example, not replaced), before the Object Handler stops processing, the user exit routine OBJHEX02 is invoked, if available. With OBJHEX02, you can specify whether to continue or terminate Object Handler processing. In the case of termination, you can set a condition code. For further details, see the source of the user exit routine SRC-EX02 in the Natural system library SYSOBJH.

# Object Handler - Tools

The Object Handler provides special features to display status information and reports and to check or modify trace settings.

This section covers the following topics:

- Status
  - Last Result
  - Traces
  - Reports
- 

## Status

Displays the Object Handler functions currently used, the user environment, the Workplan library and the setting of the trace option described below. To display the status, enter the command `SHOW STATUS`. See also the section `Commands for CUIs`.

## Last Result

Displays the last internal command issued by the processing interface of the Object Handler and possible return codes and messages. To display the last result, enter the command `SHOW LAST RESULT`. See also the section `Commands for CUIs`.

## Traces

Activates or deactivates the trace function. Traces record internal Object Handler program flows to provide control information for error diagnoses.

The trace option is set to `OFF` by default. To change the setting, see the command `SET TRACE` as described in `Commands for CUIs`.

## Reports

Lists the objects loaded, unloaded or scanned, and records errors that may interrupt processing. See also `Work File Options` in the section `Settings`.

The report option is set to `ON` by default and is displayed after the unload, load or scan function has been executed. To display the contents of the latest report file, enter the command `SHOW REPORT` as described in the section `Commands for CUIs`.

# SYSOBJH Profile

You can define an individual profile for your Object Handler utility. For this purpose, Natural provides the text object PROFILE in the library SYSOBJH. In PROFILES, you can enter general or user-specific profiles with corresponding default settings (see the listing below). These defaults are then displayed when you enter the relevant function screens.

## To activate the values defined in PROFILE

1. Save the text object PROFILE under the name PROFILES in the library SYSOBJH.
2. Invoke the Object Handler to activate PROFILES.

## Text Object PROFILE

```

0010*****
0020* Application: SYSOBJH (Object Handler)
0030* Object:      PROFILE
0040*****
0050* Function:    Source of the SYSOBJH Profile PROFILES
0060*
0070*      To activate save as 'PROFILES' in the library SYSOBJH.
0080*
0090*      Used to set defaults for SYSOBJH flags and options.
0100*
0110*      For possible values see the '[General-Start]' to '[General-End]'
0120*      block. It defines the options for all users.
0130*
0140*      In the '[User-Start uid]' to '[User-End uid]' blocks (where 'uid'
0150*      is the user ID as contained in the Natural system variable '*USER')
0160*      it is possible to define options for single users.
0170*      See the example in the '[User-Start UID-EXAM]' to
0180*      '[User-End UID-EXAM]' block.
0190*      Notes:
0200*      - Empty lines or lines starting with '*' or '/' are ignored.
0210*      - A text after '/' is ignored.
0220*      - The line length must not exceed 250 bytes.
0230*      In mainframe environments only 90 bytes are allowed.
0240*****
0250
0260[General-Start]
0270   Advanced-Mode                N                /* Y/N
0280   Display-Cmd-in-Advanced-Mode  N                /* Y/N
0290   Display-ExecutionMsg          N                /* Y/N
0300   Workplan-Library              WORKPLAN        /* Library name
0310   Workplan-Library-DBID         0                /* 0=FNAT/FUSER
0320   Workplan-Library-FNR          0                /* 0=FNAT/FUSER
0330   TRACE                         N                /* Y/N
0340   TRACE-TARGET                  S                /* Screen/Work File
0350   Option-Replace                 N                /* N/Y/O/E
0360   Option-TRANSFER-FORMAT        N                /* Y/N
0370   Option-TR-INCLUDE-LINE-NUMBERS N                /* Y/N
0380   Option-TR-SUBSTITUTE           N                /* Y/N
0390   Option-TR-TRANSLATE-TO-UPPER  N                /* Y/N
0400   Option-TR-USE-CONVERSION-TABLE N                /* N/S/U
0410   Option-TR-CONV-TABLE-NAME-LOAD OTNCONEA
0420   Option-TR-CONV-TABLE-NAME-UNLD OTNCONAE
0430   Option-Write-Report            Y                /* Y/N
0440   Option-Write-Restart-Info      N                /* Y/N

```

**SYSOBJH Profile****Text Object PROFILE**

```
0450 USE-OPTION-WORKPLAN N /* Y/N
0460 OPTION-WORKPLAN-Name OPTIONWP /* Name of Workplan
0470 USE-PARAMETER-WORKPLAN N /* Y/N
0480 PARAMETER-WORKPLAN-Name PARAWPLN /* Name of Workplan
0490 WORK-FILE-1-Name C:\TEMP\W1.SAG /* Work File 1
0500 Report-File-Name C:\TEMP\W4.DAT /* Work File 4
0510 Restart-File-Name C:\TEMP\W6.DAT /* Work File 6
0520 Trace-File-Name C:\TEMP\W10.DAT /* Work File 10
0530[General-End]
0540
0550[User-Start UID-EXAM]
0560 Advanced-Mode Y /* Y/N
0570[User-End UID-EXAM]
0580
```

# Natural Recording Utility

With the Natural Recording utility, you can record a Natural session and later play back the recorded session.

Related Documentation: Terminal Commands, Natural Terminal Commands documentation.

The Natural Recording utility documentation covers the following topics:

- Purpose of Recording
  - Data and Functions Recorded
  - Recording a Session
  - Playing Back a Recording
  - Manipulating a Recording
- 

## Purpose of Recording

The Natural Recording utility can be used for the following purposes:

- **Demonstration**  
Instead of having to type in several commands, such as input data by hand, you can play back a recorded sequence of keyboard actions to demonstrate a standard procedure.
- **Application development**  
When applying the same modifications to several objects (for example, programs or maps), you can use a recording to reduce the amount of work involved and at the same time ensure that the modifications are actually the same for all objects affected.
- **Testing**  
You can execute a standard testing procedure by simply playing back a recording.
- **Quality control**  
Before and after making changes to an application, you can play back a recording and compare the results of the two runs to make sure that certain things were not affected by the changes.
- **User training**  
You can incorporate the playback of recordings into training programs for users, to show them specific procedures. Also, you can record users' keyboard actions in a session and then inform them of any errors they make or of ways to carry out actions more efficiently. The recording of users' actions can also help you to detect any flaws in an application's user interface.

## Data and Functions Recorded

The Natural Recording utility records:

- All input data and commands (including terminal commands) entered on the screen,
- Any function keys pressed.
- The current cursor position as contained in the Natural system variable \*CURSOR.

## Recording a Session

Below are the steps required to activate and deactivate a recording:

- Specifying Libraries
- Activating a Recording
- Deactivating a Recording

### Specifying Libraries

#### ▶ To specify the library in which all subsequent recordings are to be stored

- Enter the terminal command `%B=library-name`.

If you activate the recording process without having specified *library-name*, the name of the library in which the recording is stored is the same as the value of the system variable `*INIT-USER` at the time when the recording process is activated.

When you log on to another library during a session being recorded, the library in which the recording is being stored remains the same (that is, either the one specified with `%B=` or the `*INIT-USER` library); this means that one recording can record keyboard actions across multiple applications.

### Activating a Recording

#### ▶ To activate a recording

- Enter the terminal command `%Bname`.

All subsequent keyboard actions are recorded.

*Name* denotes the name under which the data recorded are saved in source form as a Natural object of the type Recording. You can treat this source as any other Natural source (for example, delete it, copy it), except that you must not edit it: recordings contain binary data an editor will destroy.

*Name* can only be specified once. If a recording object of the same name already exists in the library specified for recording, Natural returns the message "Error in recording activation".

#### **Attention:**

Any situation that leads to a backout transaction or rollback (for example, a non-activity timeout) while a recording is in progress, will delete part of the recording thus making the entire recording useless.

Terminal Command `%Aname` included in a recording should be followed by Terminal Command `%B` as described in Recording %A below.

### Deactivating a Recording

#### ▶ To deactivate a recording

- Enter the terminal command `%B`.

The recording has terminated.

## Playing Back a Recording

When a recording is played back, the sequence of, for example, commands and function keys is actually executed again.

The recording is independent of the terminal type; that is, a session recorded on one terminal can be played back on a terminal of another type. You can also play back a recording in batch mode; a recorded online session may, of course, react differently when played back in batch.

Below is information on:

- Step Mode and Background Mode
- Activating a Playback
- Interrupting a Playback

### Step Mode and Background Mode

A recording can be played back in two modes: background mode and step mode.

In background mode, the entire recording is played back invisibly; that is, all keyboard actions of the recording are carried out without anything being displayed to you on the terminal screen during the execution of the recording. You cannot interrupt a recording that is played back in background mode, unless the recording contains the terminal command `%R` as explained below.

In step mode, a recording is played back step by step and all keyboard actions are displayed on the screen. By choosing any function key, you proceed from one step to the next. In step mode, it is also possible for you to interrupt the recording by pressing `CLEAR` as explained below.

By default, a recording is played back in background mode.

#### To set modes

- Enter the terminal command `%GON` to activate step mode.
- Enter the terminal command `%GOFF` to deactivate step mode and activate background mode.
- Enter the terminal command `%G` to toggle between step and background mode.

### Activating a Playback

#### To play back a recording

- Enter the Natural terminal command `%Aname`.

The recording saved under the specified name is executed again.

#### **Recording `%Aname`**

If you issue the command `%Aname` while a session is being recorded, the recording specified with `%Aname` is not executed but the command `%Aname` is included into the object source that is being recorded. Thus, you can execute a recording from within another recording and concatenate a series of recording to one another. However, you cannot have nested recordings; the execution of the recording that contains the `%Aname` command stops after this command and is not resumed when the execution of `name` finishes. As a result, the data recorded after `%Aname` will never be played back. To avoid this, you should enter `%B` immediately after you have entered `%Aname` in a recording.

## Interrupting a Playback

### To interrupt a recording that is played back in step mode

- Press CLEAR.

Once you have interrupted a recording, you have the following options:

- You can continue your session normally from the point where you stopped the recording.
- You can insert additional keyboard actions into the recording: after you have pressed CLEAR, enter the command **%B** and all actions you perform are inserted into the source of the recording until you enter **%B** again. Then, the execution of the recording is resumed.
- You can alter the next step in the recording: after you have pressed CLEAR, enter the command **%R**, then enter the input data for the next step. The newly entered input data overwrite the input data for this step in the recorded source. When you press ENTER, this step is executed with the new input data and subsequently the execution of the recording is resumed.
- You can execute any helproutine: after you have pressed CLEAR, enter the command **%J** directly followed by the name of the desired helproutine. The helproutine is invoked and the execution of the recording is continued as soon as the execution of the helproutine ends.

## Manipulating a Recording

By recording the terminal command **%R**, you can manipulate a single step in a recording when it is played back. This applies in step mode and in background mode. In background mode, **%R** is the only way to interact with a recording that is being played back. Interaction, for example, may be required to provide an input option for sensitive data, such as passwords which are unknown at the time of the recording.

If the terminal command **%R** (redisplay last screen) has been recorded, the subsequent screen is open for user input when the recording is played back; that is, the input data for this screen are not taken from the recording but from what the user enters. Subsequently, the execution of the recording is continued.

# SYSBPM Utility - Buffer Pool Management

The utility SYSBPM is used for managing buffer pools of the types "Natural buffer pool", "DL/I buffer pool" or "sort buffer pool"; see also keyword subparameter TYPE of the NTBPI macro.

SYSBPM provides statistical information on the current status of the buffer pool including the buffer pool cache and on the Natural programming objects loaded in the buffer pool (BP) and the BP cache. SYSBPM also provides administration functions.

For a general description of the Natural buffer pool, refer to the Natural Operations for Mainframes documentation.

The buffer pool is defined with the macro NTBPI in the Natural parameter module, or with the corresponding dynamic profile parameter BPI, both of which are described in the Natural Parameter Reference documentation.

## Note:

In the SYSBPM documentation, buffer pool is also referred to as BP.

The SYSBPM utility documentation covers the following topics:

- Invoking and Operating SYSBPM      Invoking the SYSBPM utility and selecting functions with the SYSBPM Main Menu.  
Using SYSBPM in a Sysplex environment.
- List Objects      Displaying information on the objects loaded in the buffer pool and/or the BP cache.  
Specifying objects or a range of objects to be displayed.
- Delete Objects      Deleting objects from the buffer pool and the BP cache.
- Directory Information      Displaying the full directory of an object loaded in the buffer pool or the BP cache.
- Hexadecimal Display      Displaying an object loaded in the buffer pool in hexadecimal format.
- Write to Work File      Writing to a local file or a PC text file the directory information of the objects loaded in the buffer pool and/or the BP cache.
- Display Sorted Extract      Displaying a sorted and limited list of the objects loaded in the buffer pool or the BP cache.
- Buffer Pool Statistics      Displaying object-independent statistics on the buffer pool including hash table statistics.
- BP Cache Statistics      Displaying statistics on the BP cache including general BP cache statistics, BP cache call statistics and BP cache hash statistics.
- Select Buffer Pool      Displaying, selecting and resetting buffer pools defined in your Natural environment.
- Blacklist Maintenance      Maintaining a blacklist of objects which are not to be executed and loaded into the buffer pool and, if they are already in the buffer pool, which are to be deleted.
- Preload List Maintenance      Maintaining a preload list where you can specify the names of objects to be loaded into the buffer pool on its initialization.
- SYSBPM Direct Commands      Using direct commands for executing SYSBPM functions.
- Batch Processing      Executing SYSBPM in batch mode.
- Appl. Programming Interfaces      Using application programming interfaces for handling the objects loaded in the buffer pool.

# Invoking and Operating SYSBPM

The functions of the SYSBPM utility always refer to Natural programming objects currently loaded in the Natural buffer pool and/or BP cache. Note that the buffer pool or BP cache only contains executed Natural programming objects that have been stowed or cataloged. Objects executed with the RUN command are not loaded into the buffer pool or BP cache.

The SYSBPM utility is menu-driven. You can use a function code, SYSBPM direct commands (see the relevant section) and a PF key to perform a specific function.

This section describes how to invoke the SYSBPM utility, obtain online help text and select functions by using the SYSBPM Main Menu. The functions are described in detail in the relevant sections of the SYSBPM documentation.

In addition, information is provided on the use of SYSBPM in a Sysplex environment.

The following topics are covered below:

- Invoking SYSBPM
- Online Help
- SYSBPM Main Menu - Fields and Functions
- SYSBPM in a Sysplex Environment

## Invoking SYSBPM

### To invoke the SYSBPM utility

- In the command line, enter the system command SYSBPM.

The SYSBPM Main Menu is displayed:

```

16:12:23          ***** NATURAL SYSBPM UTILITY *****          2002-08-27
BPNAME QA41GBP          - Main Menu -          Type Global Nat
BPPROP OFF          Loc DAEF QA41
          Preload QA41GBPL

          Object Functions          Object Pool Statistics

          L List Objects          A Buffer Pool
          D Delete Object          C BP Cache

          I Directory Information          Other Functions
          H Hexadecimal Display
          W Write to work file
          X Display sorted extract
          ? Help
          . Exit
          S Select Buffer Pool
          B Blacklist Maintenance
          P Preload List Maintenance

Code .. _   Library ... * _____
            Object .... * _____
            DBID ..... 0_____ FNR .. 0_____ Object Pool ... B (B,C,*)

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
          Help          Exit Last          Flip          Canc
    
```

- On the SYSBPM Main Menu, specify the executed object(s) loaded in the buffer pool or BP cache by choosing either of the options below:
  - Complete the input fields as described in SYSBPM Main Menu - Fields and Functions below.  
Or, in the command line, enter a SYSBPM direct command as described in SYSBPM Direct Commands.

## Online Help

The online help function of SYSBPM provides information on all input fields that appear in any SYSBPM screen.

### To invoke the online help function for a SYSBPM input field

- On any SYSBPM screen, position the cursor in any input field and press PF1 or enter a question mark (?).

The Help window appears for the relevant field with a list of all valid input values.

## SYSBPM Main Menu - Fields and Functions

Below is information on:

- Fields
- Functions

### Fields

Listed below are the fields displayed on the SYSBPM Main Menu:

Code	Fields	Explanation						
	BPNAME	The name of the global buffer pool as specified with the profile parameter BPNAME. For a local buffer pool, no name but a blank field is displayed for BPNAME.  See also BPNAME in the Natural Parameter Reference documentation.						
	BPPROP	The setting of the profile parameter BPPROP to control the propagation of changes to an object in a buffer pool.  See also BPPROP in the Natural Parameter Reference documentation.						
	Type	The type of buffer pool, such as Global Nat, Local Nat, Global Sort or Global DL/I.						
	Loc	The location. Displays the host ID (in the example screen above: DAEF) and the subsystem ID (in the example screen above: QA41).						
	Preload	The name of a preload list, if loaded.  See also Preload List Maintenance.						
	Library	The name of the library where the executed object is stored. You can specify a name or use asterisk (*) notation.  The default, asterisk (*), selects all libraries.						
	Object	The name of the executed object loaded in the buffer pool. You can specify a name or use asterisk (*) notation.  The default, asterisk (*), selects all objects.						
	DBID/FNR	The database ID (DBID) and file number (FNR) of the system file FNAT or FUSER where the executed object is stored and from where it is loaded.  If you specify <b>0</b> (this is the default) as DBID or FNR, the specified object(s) will be selected regardless of their DBID and FNR. Any value other than <b>0</b> represents a particular DBID or FNR specification.						
	Object Pool	Applies to the functions <b>L</b> , <b>I</b> , <b>W</b> or <b>X</b> as described below.  Selects the type of object pool(s) for applying any of the functions listed above:  <table style="margin-left: 40px;"> <tr> <td style="padding-right: 20px;">B</td> <td>Buffer pool. This is the default.</td> </tr> <tr> <td>C</td> <td>BP cache.</td> </tr> <tr> <td>*</td> <td>Both buffer pool and BP cache.</td> </tr> </table>	B	Buffer pool. This is the default.	C	BP cache.	*	Both buffer pool and BP cache.
B	Buffer pool. This is the default.							
C	BP cache.							
*	Both buffer pool and BP cache.							

## Functions

The functions provided on the SYSBPM Main Menu are organized in three sections:

- The Object Functions section contains functions for displaying or manipulating objects in the buffer pool or BP cache.
- The Object Pool Statistics section contains functions for obtaining object-independent statistical data on the buffer pool or BP cache including hash tables. Object-independent data do not include any individual information on the object, such as object name, size or addresses.
- The Other Functions section contains functions for selecting a buffer pool and for specifying the objects to be loaded into the buffer pool or BP cache.

The individual functions are listed below. You invoke a function by entering in the Code field the code that corresponds to the function desired:

Code	Function	Explanation
L	List Objects	Displays information on the objects loaded in the buffer pool and/or the BP cache (if used). Each list item can be accessed individually and various functions can be performed for each object.
D	Delete Objects	Deletes one or more objects from the buffer pool and the BP cache.
I	Directory Information	Displays the full directory information of a specified object loaded in the buffer pool or the BP cache.
H	Hexadecimal Display	Displays in hexadecimal format a specified object loaded in the buffer pool.
W	Write to Work File	Writes to a local file or a PC text file the object directory information located in the buffer pool and/or BP cache.
X	Display Sorted Extract	Displays a sorted list of 50 object directories located in the buffer pool or BP cache. The list items can be arranged by using any of the sort criteria provided.
A	Buffer Pool Statistics	Invokes the BP Statistics Main Menu. From this menu, you can invoke object-independent statistics functions for the buffer pool including hash table statistics.
C	BP Cache Statistics	BP cache required. Invokes the BP Cache Statistics Main Menu. From this menu, you can invoke object-independent statistics functions for the BP cache including hash table statistics.
S	Select Buffer Pool	Displays a selection list of all available buffer pools.
B	Blacklist Maintenance	Invokes the Blacklist Maintenance menu which is used to maintain a blacklist of objects which are <b>not</b> to be executed.
P	Preload List Maintenance	Invokes the Preload List Maintenance menu. In a preload list, you can specify the names of objects which are to be loaded into the buffer pool when the buffer pool is initialized.

## SYSBPM in a Sysplex Environment

Whenever Natural switches to another operating system image (host), Natural also switches buffer pools. A switch of buffer pools is indicated by a different host ID which is displayed in the Loc field of the SYSBPM screen.

Switching can take place after each terminal I/O, that is, after choosing any function key or by choosing ENTER. After switching buffer pools, browsing and positioning functions will not be executed (top, bottom, +, -, left, right). Instead, the list starts from the top of the new buffer pool.

If the BPPROP profile parameter (see the relevant section in the Natural Parameter Reference documentation) is set to PLEX or to GPLEX, SYSBPM commands, such as manipulating blacklists, deleting objects or initializing the buffer pool are first executed as usual, and then propagated to other buffer pools available on the same subsystem. If a BP switch caused a function to be aborted or propagated, a corresponding message is issued. A corresponding message also appears if Natural has successfully switched to another host and changed buffer pools.

# SYSBPM - List Objects

This function invokes the List Objects screen where you can obtain statistical data on the directories of Natural programming objects currently loaded in the buffer pool (BP) or the BP cache (if used).

Below is information on the statistical data displayed on the List Objects screen and the commands and functions available for selecting an object or a range of objects, manipulating their current status or navigating in the List Objects screen.

See also the function Display Sorted Extract for a sorted display of objects.

- Invoking List Objects
  - Columns and Selection Options
  - Line Commands
  - Navigation
- 

## Invoking List Objects

### To invoke the List Objects screen

- On the SYSBPM Main Menu, in the Code field, enter Function Code **L** and specify the object(s): see the valid field input values as described in the section Invoking and Operating SYSBPM.

Or, go directly to the list of objects in the buffer pool:

In the command line, enter

`DISPLAY LIST library-name object-name dbid fnr.`

Or, go directly to the list of objects in the BP cache:

In the command line, enter

`DISPLAY CLIST library-name object-name dbid fnr.`

The List Objects screen appears:

```

17:13:17          ***** NATURAL SYSBPM UTILITY *****          2002-09-16
BPNAME QA41GBP          - List Objects -          Type Global Nat
BPPROP OFF          Loc DAEF QA41
C  Library  Object    DBID  FNR  Loc  RLD  Use  Max  Reuse  TotalUC  ObjSize  Sto
*  *
__ SYSBPM  BPMCALL    10  1640  B          1  1          5  8,516  12
__ SYSBPM  BPMNSC     10   410  B          1          4  3,380   4
__ SYSDLINP PCNDL02   255  253  B  R          1          19   292   4
__ SYSLIBS  NAT00017    10   410  B          1          1  5,000   8
__ SYSLIB  ATEST     10  1640  B  R          1  340  16,148  16
__ SYSLIB  CATAL10   10   410  B          1          1  4,256   8
__ SYSBPM  BPM141-M    10  1640  B          1          1  5,944   8
__ SYSDLINS U246005   255  253  B  R          1          14    52   4
__ SYSBPM  MENU       10  1640  B          1          5  10,392  12
__ SYSLIBS  NAT00040    10   410  B          1          1  2,816   4
__ SYSLIBS  NAT00034    10   410  B          1          1  2,672   4
__ SYSDLIND DNDL01   255  253  B  R          3          42   552   4
__ SYSLIB  ACATAL1   10   410  B          1          3  55,728  56
__ SYSDLINS U246004   255  253  B  R          2          28   172   4

Top of List
Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      Exit  Last  Cache --  -  +  ++>  Canc

```

The List Objects screen lists all individual objects

1. currently loaded in the buffer pool (first part of the display) and
2. currently loaded in the BP cache (second part of the display).

The statistics displayed are snapshots of the contents of the buffer pool which are refreshed every time you press ENTER.

**Note for GDA objects loaded in the buffer pool:**

On the List Objects screen, two entries may be displayed for a GDA (global data area): one entry contains data on the GDA itself and the other entry contains the internal Natural symbol table for this GDA. This may happen if a program has been cataloged that references a GDA.

## Columns and Selection Options

Below is a description of the columns and statistical data displayed on the List Objects screen, and the input field and values provided for specifying object selection criteria.

 **To specify selection criteria for the objects to be displayed**

- In the input fields underneath the column titles, enter a valid value or range as described for the relevant fields.  
The default value is a blank character or asterisk (\*) which selects all objects.

Column	Explanation
C	In this column, enter a line command to perform a function for the object. See also Line Commands below.
Library	The library from which the object was loaded.  To specify selection criteria, see Name and Range Specification below.

Column	Explanation
Object	<p>The name of the object.</p> <p>To specify selection criteria, see Name and Range Specification below.</p>
DBID	<p>The database ID (DBID) of the Natural system file from which the object was loaded.</p> <p>To select objects of a specific database, enter a valid numeric value.</p>
FNR	<p>The file number (FNR) of the Natural system file from which the object was loaded.</p> <p>To select objects of a specific FNR, enter a valid numeric value.</p>
Loc	<p>Location of the object :</p> <p>B        Buffer pool.</p> <p>B/C      Buffer pool and BP cache.</p> <p>C        BP cache.</p> <p>C/B      BP cache and buffer pool.</p> <p>If <b>B</b> is listed in the first position, the statistical data derive from the buffer pool. If <b>C</b> is listed first, the data derive from the BP cache.</p> <p>Additionally, depending on this positioning, different line commands apply to the fields on the statistics screen (see also Line Commands below).</p> <p>To specify the object location(s), enter any of the values below:</p> <p>B        Selects all objects loaded in the buffer pool only.</p> <p>B/C      Selects all objects loaded in the buffer pool as well as in the BP cache.</p> <p>B*       Selects all objects loaded in the buffer pool or in both the buffer pool and BP cache (B* = B + B/C).</p> <p>C        Selects all objects loaded in the BP cache only.</p> <p>C/B      Selects all objects loaded in the BP cache as well as in the buffer pool.</p> <p>C*       Selects all objects loaded in the BP cache or in both the BP cache and buffer pool (C* = C + C/B).</p>

Column	Explanation
RLD	<p>Current status of the object in the buffer pool or the BP cache. A BP cache status only refers to object locking and, therefore, is only indicated underneath the L (Locked) of the RLD column.</p> <p>Buffer pool:</p> <p>R Marked as resident. Resident means that the object is not deleted from the buffer pool, not even if the relevant value in the Use column is set to 0.</p> <p>L Locked while load function is ongoing.</p> <p>D A Delete call for the object is pending. The object will be deleted from the buffer pool as soon as the value in the Use column is reset to 0.</p> <p>BP cache:</p> <p>L Locked while load function is ongoing.</p> <p>D Locked for delete.</p> <p>To select all objects of a specific status, as described above, enter the code R, L or D.</p>
Use	<p>Buffer pool only.</p> <p>The number of Natural applications that are currently executing the object.</p> <p>To select objects, enter:</p> <p><i>value</i> A numeric value. Selects all objects with this number. Example: 10</p> <p>&gt;<i>value</i> A numeric start value (&gt;). Selects all objects with a number greater than <i>value</i>. Example: &gt;10 Selected: 11, 21 Not selected: 10</p> <p>&lt;<i>value</i> A numeric end value(&lt;). Selects all objects with a number less than <i>value</i>. Example: &lt;10 Selected: 9, 8 Not selected: 10</p>
Max	<p>Buffer pool only.</p> <p>The maximum number of applications that have executed the object since it was loaded into the buffer pool.</p> <p>To select objects, see the valid input values in Use above.</p>

Column	Explanation
Reuse	<p>BP cache only.</p> <p>Indicates how many time the object has been loaded (reused) from the BP cache into the buffer pool.</p> <p>To select objects, see the valid input values in Use above.</p>
TotalUC	<p>Total Use Count: The total number of Locate calls of the object since it was loaded into the buffer pool.</p> <p>If a BP cache is used, this value is not lost if the object is removed from the buffer pool and saved to the BP cache. Therefore, this value indicates the number of times the object has been used since it was loaded from the system file.</p> <p>For buffer pool objects, this value is updated regularly. For BP cache objects, this value is only updated after the object was removed from the buffer pool and saved in the BP cache.</p> <p>To select objects, enter:</p> <p><i>value</i>      A numeric value or a numeric start value (&gt;). or              Selects all objects with a number greater than or equal to <i>value</i>. &gt;<i>value</i>        Example: &gt;10                     Selected: 10, 11, 21                     Not selected: 9</p> <p>&lt;<i>value</i>        A numeric end value(&lt;).                     Selects all objects with a number less than <i>value</i> or equal to <i>value</i>.                     Example: &lt;10                     Selected: 10, 9, 8                     Not selected: 11</p>
ObjSize	<p>The size of the object.</p> <p>To select objects, see the valid input values in TotalUC above.</p>
Sto	<p>Storage that has to be allocated for the object in the buffer pool or BP cache. The text record size of the buffer pool is defined at buffer pool initialization.</p> <p>To select objects, see the valid input values in Use above.</p>
(*) BP Load Time	<p>The date and time when the object was first loaded into the buffer pool.</p> <p>This date and time will be kept until the object has been removed from both the buffer pool and BP cache (deletion from the BP cache only will not remove the display of date and time).</p> <p>To select objects, see Date Specification and Time Specification below.</p>
(*) BP Last Action	<p>Buffer pool only.</p> <p>The date and time when the object was last used by an application.</p> <p>To select objects, see Date Specification and Time Specification below.</p>

Column	Explanation
(*) BPC Load Time	<p>BP cache (BPC) only.</p> <p>The date and time when the object was first loaded into the BP cache.</p> <p>This date and time will be kept until the object has been removed from the BP cache.</p> <p>To select objects, see Date Specification and Time Specification below.</p>
(*) BPC Last Get	<p>BP cache (BPC) only.</p> <p>The date and time when the object was last swapped from the buffer pool into the BP cache. This time stamp is also updated if the object was already available in the BP cache and had, therefore, not been written to the BP cache again.</p> <p>To select objects, see Date Specification and Time Specification below.</p>
(*) BPC Last Put	<p>BP cache (BPC) only.</p> <p>The date and time when the object was last loaded from the BP cache into the buffer pool.</p> <p>To select objects, see Date Specification and Time Specification below.</p>
(*) 1.BPperiod	<p>BP cache (BPC) only.</p> <p>The time frame the object has been available in the buffer pool starting with the time the object was first loaded and ending with the time the object was first swapped from the buffer pool into the BP cache. After 24 hours, the display of the time frame is canceled and replaced by this sign: **:**:**:**.*</p>

\* To display these columns, press PF11 as described in Navigation below.

## Name and Range Specification

To specify the objects to be displayed on the List Objects screen, in the input fields for Library or Object, enter a name or a range of names.

In the list of options below, *value* is any combination of one or more characters:

Input Value	Selected Libraries/Objects
*	All libraries/objects.  This is the default.
<i>value</i>	All libraries/objects with a name equal to <i>value</i> .
<i>value</i> *	All libraries/objects with a name that starts with <i>value</i> .  Example: AB* Selected: AB, AB1, ABC, ABEZ Not selected: AA1, ACB
<i>value</i> ?	All libraries/objects with a name that starts with <i>value</i> and ends with any single character for each question mark (?) entered.  Example: ABC? Selected: ABCA, ABCZ Not selected: AXC, ABCAA
<i>value</i> ? <i>value</i> ?	All items that match <i>value</i> combined with asterisk (*) and question mark (?) in any order.
<i>value</i> * <i>value</i> ?	Example: A?C*Z Selected: ABCZ, AXCBBBZ, ANCZ
* <i>value</i> ? <i>value</i> *	Not selected: ACBZ, ABDEZ, AXCBBBZA
<i>value</i> >	All libraries/objects with a name greater than or equal to <i>value</i> .  Example: AB> Selected: AB, AB1, BBB, ZZZZZZZ Not selected: AA1, AAB
<i>value</i> <	All libraries/objects with a name less than or equal to <i>value</i> .  Example: AX< Selected: AB, AWW, AX Not selected: AXA, AY

## Date Specification

To specify the objects to be displayed on the List Objects screen, in the input fields for dates, enter a date, a range of dates, a special date or a range of special dates.

A date must be specified in the format *YYYYMMDD* (*YYYY* = year, *MM* = month, *DD* = day).

In the list of options below, *value* is any combination of one or more digits:

Input Value	Selected Objects
<i>YYYYMMDD</i>	All objects with a date equal to <i>YYYYMMDD</i> . Example: 20020831
<i>value*</i>	All objects with a date that starts with <i>value</i> . Example: 2002* Selected: 20020101, 20021231 Not selected: 20011231, 20030101
<i>value&gt;</i>	All objects with a date greater than or equal to <i>value</i> . Example: 2001> Selected: 20010101, 20021231, 20020101 Not selected: 20001231
<i>value&lt;</i>	All objects with a date less than <i>value</i> . Example: 2001< Selected: 19990101, 20001231 Not selected: 20010101, 20011231
<b>Special Dates</b>	
<u>TODAY</u> (+/- <i>nnnn</i> )	All objects with the date of the current day.  The day can be followed by <b>+<i>nnnn</i></b> or <b>-<i>nnnn</i></b> where <i>nnnn</i> has a maximum of 4 digits. The resulting date is computed as the date of the current day plus or minus <i>nnnn</i> days. Example: the current date is 20020301 TODAY +5 results in 20020306
<u>YESTERDAY</u>	All objects with the date of the day before the current day.
<u>MONTH</u>	All objects with the date range of the current month.  Example: the current month is 200209. Selected: 20020901, 20020930. Not selected: 20021001.
<u>YEAR</u>	All objects with the date range of the current year.  Example: the current year is 2002. Selected: 20020101, 20021231. Not selected: 20031001.

## Time Specification

To specify the objects to be displayed on the List Objects screen, in the input fields for times, enter a time or a range of times.

The time must be specified in the format *HH:II:SS* (*HH* = hours, *II*= minutes, *SS* = seconds).

In the list of options below, *value* can be any combination of one or more digits:

Input Value	Selected Objects
<i>HH:II:SS</i>	All objects with a time equal to <i>HH:II:SS</i> . Example: 14:15:16
<i>value*</i>	All objects with a time that starts with <i>value</i> . Example: 13* Selected: 13:00:00, 13:10:53, 13:59:59 Not selected: 12:59:59, 14:00:00
<i>value&gt;</i>	All objects with a time greater than or equal to <i>value</i> . Example: 12:30> Selected: 12:30:00, 12:30:01, 16:34:01 Not selected: 12:29:59
<i>value&lt;</i>	All objects with a time less than <i>value</i> . Example: 12:30< Selected: 12:29:59 Not selected: 12:30:00

## Screen Navigation

To scroll in the List Objects screen, use the following PF keys and direct commands:

PF Key	Command	Function
PF1		Help. If chosen with the cursor on Column C: lists all commands and functions available.  If chosen with the cursor on the input fields underneath the column titles: lists all possible input values for object selection.
PF5	CACHE	Only applicable if BP cache data exist.  Scrolls to the top of the list with statistical data on BP cache objects.
PF6	--	Scrolls to the top of the list with statistical data on buffer pool objects.
PF7	-	Scrolls up one page.
PF8	+	Scrolls down one page.
PF9	++	Scrolls to the end of the list.
PF10	<	Scrolls left in the list.
	LEFT	Press PF11 to scroll to the right.
PF11	>	Scrolls right in the list and displays the additional screen columns: BP Load Time, BP Last Action, BPC Load Time, BPC Last Get, PBC Last Put and 1.BPperiod.
	RIGHT	Press PF10 to scroll to the left.

## Line Commands

On the List Objects screen, in Column C, for each object displayed, you can enter any of the line commands listed below:

Command	Function
CL	Buffer pool only. Releases an object marked as resident.
DE	Marks an object to be deleted from the buffer pool or BP cache. The object is deleted as soon as the relevant Use Count is set to <b>0</b> . If issued for a buffer pool object, the object will be deleted from both the buffer pool and the BP cache. If issued for a BP cache object, the object will be deleted from the BP cache only.
HD	Buffer pool only. Displays in hexadecimal format the directory information of an object.
HE	Buffer pool only. Corresponds to the function Hexadecimal Display as described in the relevant section.
FO	Buffer pool only. Deletes an object immediately from the buffer pool, regardless of the relevant Use Count.
LD	Corresponds to the function Directory Information as described in the relevant section.
RE	Buffer pool only. Marks an object as resident.
ZO	For the object specified, zooms in the fields Object, Use, Max, Reuse, TotalUC, ObjSize and Sto and displays them in full length. To zoom out, press ENTER.

For each command entered, a confirmation message is displayed for the relevant line overwriting text of rows displayed on the screen. Possible messages are:

- Failed (in response to any function that has not been executed successfully),
- Deleted (in response to the command DE or FO),
- Released (in response to the command CL) and
- Resident (in response to the command RE).

# SYSBPM - Delete Objects

This function deletes one object or more from the buffer pool (BP) and/or the BP cache. To specify the object pool from which the object(s) is to be deleted, use the DELETE command as described in SYSBPM Direct Commands.

Objects that have a Current Use Count (see Directory Information) of **0** are deleted immediately. Objects of a Current Use Count greater than **0** are marked for deletion and deleted as soon as their Current Use Count is reset to **0**.

## To invoke Delete Objects

- On the SYSBPM Main Menu, enter Function Code **D** and specify the object(s) to be deleted: see the valid field input values as described in Invoking and Operating SYSBPM.

Or, in the command line, enter  
DELETE *library-name object-name dbid fnr*.

# SYSBPM - Directory Information

This function is used to display the full directory of a Natural programming object currently loaded in the buffer pool or BP cache.

## To invoke Directory Information

- On the SYSBPM Main Menu:
  - In the Code field, enter Function Code **I**.
  - In the fields Object, Library, DBID and FNR, specify the object for which to display the directory: the valid input values are described in Invoking and Operating SYSBPM.
  - In the Object Pool field, enter the code that corresponds to the object pool from which to read the object directory information:

B or asterisk (*)	Buffer pool
C	BP cache

Or, in the command line, enter:

- DISPLAY DIRECTORY *library-name object-name dbid fnr*  
(directories of the specified objects loaded in the buffer pool).
- DISPLAY CDIRECTORY *library-name object-name dbid fnr*  
(directories of the specified objects loaded in the BP cache).

The Directory Information screen is displayed. Depending on the object pool (buffer pool or BP cache) selected, the screen provides different fields as described below.

Below is information on the Directory Information screen:

- Fields for Buffer Pool Objects
- Field for BP Cache Objects
- Commands

---

## Fields for Buffer Pool Objects

For directories of objects loaded in the buffer pool, the Directory Information screen provides the following fields and information on a specified object:

Field	Explanation
Directory of	The type (for example, map) and name of the object.
Loaded from Library	The name of the library from which the object was loaded into the buffer pool.
Loaded on DBID/FNR	The database ID and file number of the system file FNAT or FUSER from which the object was loaded into the buffer pool.
Loaded on <i>date/time</i>	The date and time when the object was loaded into the buffer pool.
Loaded by User	The ID of the user who executed the object.
Last Action on	The date and time when a user last executed the object.
BP Directory at Address	The address of the directory of the object in the buffer pool.
Object at Address	The address of the object in the buffer pool.
Allocated Size (KB)	The size that has been allocated in the buffer pool for the object.
Object Size	The size of the object.
Status (RLD)	<p>R The object is resident in the buffer pool. Resident means that the object is not deleted from the buffer pool, not even if its Use Count is set to 0.</p> <p>L The object is currently locked.</p> <p>D The Delete call for the object is pending. The object will be deleted from the buffer pool as soon as its Current Use Count (see below) is set to 0.</p>
Current Use Count	The number of applications currently executing the object.
Maximum Use Count	The maximum number of applications that have executed the object since it was loaded into the buffer pool.
BP Total Use	<p>The total number of times an object has been executed since it was loaded from the system file into the buffer pool.</p> <p>If a BP cache is used, the number in BP Total Use is also kept if the object is removed from the buffer pool and saved to the BP cache. BP Total Use is only reset if the object is deleted from the buffer pool <b>and</b> the BP cache.</p>
Cataloged	The information displayed in the Cataloged section of the Directory Information screen is identical to the information provided with the Natural system command LIST DIRECTORY as described in the relevant section in the Natural System Command Reference documentation.

## Fields for BP Cache Objects

For directories of objects loaded in the BP cache, the Directory Information screen provides the following fields and information on a specified object:

Field	Explanation
<i>object-type</i>	The type (for example, map) and name of the object.
Library	The name of the library from which the object was loaded into the buffer pool.
DBID/FNR	The database ID and file number of the system file FNAT or FUSER from which the object was initially loaded into the buffer pool.
Last Put	The date and time when the object was last loaded from the BP cache into the buffer pool.
BP Load Time	The date and time when the object was first loaded into the buffer pool.
Cache Load Time	The date and time when the object was first loaded into the BP cache.
Last Get	The date and time when the object was last swapped from the buffer pool into the BP cache.
Position Index	Serially numbered internal Natural position index of the objects in the BP cache.
First Data Block Offset	The address of the directory of the object in the BP cache.
Allocated Size (KB)	The size that has been allocated in the BP cache for the object.
Object Size	The size of the object.
Status	L Locked while load function is ongoing. D Locked for delete.
Reuse	Indicates how many times the object has been returned from the BP cache to the buffer pool.
BP Total Use	The total number of times an object has been executed since it was initially loaded from the system file into the buffer pool and then into the BP cache.
Cataloged	The information displayed in the Cataloged section of the Directory Information screen is identical to the information provided with the Natural system command LIST DIRECTORY as described in the relevant section in the Natural Command Reference documentation.

## Commands

In addition to the commands listed in SYSBPM Direct Commands, in the command line of the Directory Information screen, you can enter any of the SYSBPM direct commands listed below:

Command	Function
<u>F</u> DELETE	Buffer pool only. Deletes an object immediately from the buffer pool, regardless of its Use Count.
<u>R</u> ESIDENT	Buffer pool only. Marks an object as resident. Resident means that the object is not deleted from the buffer pool, not even if its Use Count is set to <b>0</b> .
<u>C</u> LEAR	Buffer pool only. Releases an object marked as resident.
<u>D</u> ELETE	Marks an object for deletion. See Status <b>D</b> of the buffer pool or BP cache above.
<u>N</u> EXT	Only applies if a range of objects was selected: Displays one object after the other and then redisplay the screen on which NEXT was entered.

# SYSBPM - Hexadecimal Display

This function is used to display in hexadecimal format the code of a Natural programming object currently loaded in the buffer pool.

## To invoke Hexadecimal Display

- On the SYSBPM Main Menu, enter Function Code **H** and specify an object:  
see the valid field input values as described in the section Invoking and Operating SYSBPM.

Or, in the command line, enter  
`DISPLAY HEX library-name object-name dbid fnr.`

The Hexadecimal Display screen appears with the object code displayed in hexadecimal format.

## Commands

Within the object displayed on the screen, you can move to a specific location by entering either an absolute hexadecimal address or a hexadecimal offset relative to your current position.

Listed below are the SYSBPM direct commands and PF keys available on the Hexadecimal Display screen. For further commands, see SYSBPM Direct Commands.

Command	PF Key	Function
	PF7	Scrolls up one page.
	PF8	Scrolls down one page.
GP		Displays the generated program.
KST		Displays the constant table.
MPT		Displays the multiple-purpose table.
<u>N</u> EXT		Only applies if a range of objects was selected: Displays one object after the other and then redisplay the screen on which NEXT was entered.

# SYSBPM - Write to Work File

This function writes to a work file the directory information of Natural programming objects currently loaded in the buffer pool and/or BP cache.

## To invoke Write to Work File

- On the SYSBPM Main Menu:
  - In the Code field, enter Function Code **W**.
  - In the Object Pool field, enter the code that corresponds to the object pool from which to read the object directory information:

B	Buffer pool
C	BP cache
asterisk (*)	Both buffer pool and BP cache

Or, in the command line, enter:

WRITE ALL  
(saves buffer pool and BP cache data)

WRITE BP  
(saves buffer pool data only)

WRITE BPC  
(saves buffer pool data only)

The window Work File Selection appears.

- In the Work File Selection window:
  - Select the target work file:
    - Enter N (No, this is the default) to output the data on Natural Work File 1.
    - Or, if Entire Connection is installed, enter Y (Yes) to output the data on a PC text file by using Natural Work File 7.
  - Specify the delimiter character, for example a semi-colon (;) to be used for separating the columns in the work file. The default is a blank character.

The statistical data written to the work file are snapshots of the list generated by the List Objects function. Refer to List Objects for an explanation of the columns.

The PC text file can be used as the basis for spreadsheet calculation.

# SYSBPM - Display Sorted Extract

This function generates a sorted excerpt of 50 directory entries of Natural programming objects currently loaded in the buffer pool. This list can be used for evaluation purposes, such as determining the objects to be marked as resident or to be included in a preload list as described in the relevant section.

## To invoke Display Sorted Extract

- On the SYSBPM Main Menu, enter Function Code **X** and, in the Object Pool field, specify the type of pool by choosing **B** (buffer pool) or **C** (BP cache).

The Specify Sort Criteria window appears:

- Enter any of the codes below to choose a column and the order by which to sort the statistics columns:

- O = Object Size (ObjSize). This is the default.
- T = Total Use Count (TotalUC).
- L = BP Last Action (only applicable to the buffer pool).

and

- D = Descending order. This is the default.
- A = Ascending order.

Or, use the SYSBPM direct command SORT or SORT BPC as described in the relevant section.

The BP Extract screen appears which indicates the sort criteria specified.

The BP Extract screen is similar to the List Objects screen. For an explanation of the columns and the commands that apply, refer to List Objects.

# SYSBPM - Buffer Pool Statistics

This function invokes the Buffer Pool Statistics main menu, which is used to obtain buffer-pool-related statistics (including hash table statistics) that are independent of Natural programming objects.

## To invoke Buffer Pools Statistics

- On the SYSBPM Main Menu, enter Function Code **A**.  
Or, in the command line, enter `DISPLAY STATISTICS`.

The Buffer Pool Statistics main menu is displayed.

The Buffer Pool Statistics main menu provides the following functions:

- General Buffer Pool Statistics
- Buffer Pool Load/Locate Statistics
- Buffer Pool Fragmentation
- Internal Function Usage
- Buffer Pool Hash Table Statistics

## General Buffer Pool Statistics

This function is used to monitor the performance of the buffer pool, and displays statistics regarding the activity of the buffer pool.

### To invoke General Buffer Pool Statistics

- On the Buffer Pool Statistics main menu, enter Function Code **G**.  
Or, in the command line, enter `DISPLAY GENERAL`.

The General Buffer Pool Statistics screen is displayed.

The statistics displayed on the General Buffer Pool Statistics screen are snapshots of the buffer pool which are refreshed each time you press `ENTER`. The following information is displayed:

Field	Explanation
Buffer Pool Address	The address of the buffer pool.
Directory Section	The address of the buffer pool directory section relative to the beginning of the buffer pool.  Each object loaded in the buffer pool requires a directory entry that contains information on this object. The space for these directory entries is allocated in the buffer pool.
Text Record Section	The address of the text record section relative to the beginning of the buffer pool.  After the space used by the directory entries has been allocated, the remaining space is divided into blocks called text records (whose size, by default, is 4 KB). An object can occupy one or more text records, depending on its size.
Dataspace attached	The name of the dataspace (BP cache) attached to the buffer pool.

Field	Explanation
Buffer Pool Size (MB)	<p>The size of the whole buffer pool in MB.</p> <p>The buffer pool size can be specified with the NTBPI macro in the parameter module or with the BPI profile parameter as described in the Natural Parameter Reference documentation.</p>
Directory Entry Size	The size of a directory entry in bytes.
Text Record Size (KB)	<p>The size of a text record in KB. The text record size can be specified with the NTBPI macro in the parameter module or with the BPI profile parameter, as described in the Natural Parameter Reference documentation. You can change the text record size of an existing buffer pool if you reinitialize the buffer pool by using the INITIALIZE command.</p> <p>The default text record size is set to 4 KB. However, if you use applications that consist of many rather small objects, we recommend that you reduce it to 2 KB. This reduces the percentage of unused space in the buffer pool, although it can lead to Algorithm 2 (see METHOD=S in Buffer Pool Search Methods in the Natural Operations for Mainframes documentation) being invoked more frequently.</p>
Buffer Pool Start	The date and time when the buffer pool was originally started.
Last Initialization	<p>The date and time when the buffer pool was most recently initialized, and the ID of the user who performed the initialization.</p> <p>The buffer pool is initialized when:</p> <ul style="list-style-type: none"> <li>● originally starting the buffer pool,</li> <li>● executing the INITIALIZE command of the SYSBPM utility, or</li> <li>● executing the REFRESH function of the GBP operating program (see Global Buffer Pool Operating Functions in the section Natural Buffer Pools in the Natural Operations documentation).</li> </ul>
Text Records - Total	The total number of text records.
Text Records - Used	The number of text records currently used.
Text Records - Used in %	The percentage of text records currently used.
Text Records - Max Used	The maximum number of text records used.
Text Records - Total Size	<p>The total space used by all text records used, which is Text Records Used multiplied by the size of a single text record.</p> <p>The difference between the total text record size and the total object size shows the amount of unused size in the text record section and can also be an indicator for the system administrator of whether to modify the text record size or not.</p>
Text Records - Avg Usage %	<p>The average usage in percent of all text records used, which is Objects - Total Size divided by Text Records - Total Size.</p> <p>This value should not be significantly less than 75%. If the buffer pool is almost full, any value above 75% indicates good usage of the buffer pool. If the usage is significantly less than 75%, the text record size should be reduced.</p>

Field	Explanation
Space Used %	The actual usage in percent of the text record section, which is Objects - Total Size divided by the total size of the Text Records section.  Note: If the buffer pool is almost full (that is, the value in the field Text Records - Used is almost 100%), any value above 75% indicates good usage of the buffer pool. If the usage is significantly less than 75%, the text record size should be reduced.
Objects - Loaded	The number of objects currently loaded in the buffer pool.
Objects - Max Loaded	The maximum number of objects ever loaded simultaneously in the buffer pool since the buffer pool was started.
Objects - Total Size	The total size in bytes of the objects currently loaded.
Objects - Avg TR Used	The average number of text records used by one object.
Objects - SumOfUseCounts	Totals the Use Counts of all objects currently loaded in the buffer pool.  The Use Count counts all applications currently executing an object. If an object is currently not in use, its Use Count returns to <b>0</b> (zero).
Objects - AvgLifetimeUsed (min)	The average life time (in minutes) of objects currently loaded in the buffer pool.
Objects - AvgLifetimeReplace (min)	The average life time (in minutes) of objects, which have already been replaced, that is deleted in the buffer pool.

## Buffer Pool Load/Locate Statistics

This function provides statistical information on the loading of objects into the buffer pool and the locating of objects in the buffer pool. This information also serves as an indicator of buffer pool performance.

### To invoke Buffer Pool Load/Locate Statistics

- On the Buffer Pool Statistics main menu, enter Function Code **L**.  
Or, in the command line, enter **DISPLAY LOAD**.

The Buffer Pool Load/Locate Statistics screen is displayed.

The statistics displayed on the Buffer Pool Load/Locate Statistics screen are snapshots of the buffer pool which are refreshed every time you press **ENTER**. The following information is displayed:

Field	Explanation
Total Locate Calls	The total number of object location calls; that is, the total number of times the Natural buffer pool manager was requested to search the buffer pool for an object. If the location is successful, the object has been found in the buffer pool or the BP cache and need not be loaded from a Natural system file thereby saving calls and I/Os.
Total Locate Calls - successful	The total number of successful Locate calls as an absolute number.

Field	Explanation
Total Locate Calls - failed	The total number of Locate calls that failed.
Quick Locate Calls	The total number of quick Locate calls.  Quick location means that the directory address of the last call of the requested object is still available. This is due to the fact that Natural maintains user-specific tables of internal directory entries which contain information on the objects used most recently by each Natural user. When a user invokes an object that has been used before in the Natural session, Natural passes this information to the buffer pool manager, which then bypasses the normal locate procedure. If the last call address cannot be found, a normal Locate call is automatically scheduled by the buffer pool manager.
Quick Locate Calls - successful	The number of quick Locate calls that have been successfully performed.
Quick Locate Calls - failed	The number of quick Locate calls that failed.  Failed quick Locate calls result in normal Locate calls.
Normal after Quick	The number of normal Locate calls that have been preceded by a quick Locate call.  For an explanation of normal calls, see the description of Quick Locate Calls above.
Normal after Quick - successful	The number of normal Locate calls that have been successful in locating the required object in the buffer pool or the BP cache and have been preceded by a quick Locate call.
Normal after Quick - failed	The number of normal Locate calls that failed and were preceded by a quick Locate call.
Normal Locate Calls	The total number of normal Locate calls.
Normal Locate Calls - successful	The number of normal Locate calls that were successful in locating the required object in the buffer pool.
Normal Locate Calls - failed	The number of normal Locate calls that failed. A failed normal Locate call indicates that an object has to be loaded from the database or from the BP cache.
Successful from Cache	The total number of successful Locate calls of objects that resided in the BP cache. This information is counted only if the previous Locate call (Normal after Quick failed or Normal Locates failed) failed. It indicates the number of database loads saved. This means, that, without the BP cache, the object would have to be loaded from the database.
Load Calls	The total number of load calls made since the buffer pool has been refreshed. The load calls are correlated with the access to the system file from which the objects are read.  The number of system file accesses is calculated as follows: <ul style="list-style-type: none"> <li>● Adabas system file: The number of Load Calls plus the number of Object Loads (see below). The total number does not include Adabas RC calls.</li> <li>● VSAM system file: The number of Load Calls.</li> </ul>

Field	Explanation
Object Loads	<p>The number of times an object was loaded from a Natural system file into the buffer pool.</p> <p>As several load calls may be necessary to load a single object, this value provides the actual number of object loads made since the most recent buffer pool refresh.</p> <p>When loading an object, the buffer pool manager uses different search algorithms: see METHOD=S and METHOD=N in Buffer Pool Search Methods, Principle of Operation, Natural Buffer Pool, in the Natural Operations for Mainframes documentation.</p>
Object Loads - finished	<p>Shows how many of the object loads finished successfully.</p> <p>An object load cannot finish if the load operation is canceled due to any of the following reasons:</p> <ul style="list-style-type: none"> <li>• A concurrent object load occurred: see "Object Loads - concurrent" below.</li> <li>• During the object load, an Adabas response code occurs.</li> <li>• During the object load, a SYSBPM delete operation is executed for this object.</li> </ul>
Object Loads - concurrent	<p>The number of object loads that have been performed simultaneously for the same object:</p> <p>Concurrent object loads occur if two or more Natural sessions that run simultaneously request the same object. While an object is being loaded by one session, other sessions request the same object and start loading it before a session has finished loading. In this case, the same object is loaded more than once.</p> <p>The first session that finishes loading the object will mark the object of the other sessions to be deleted from the buffer pool. The other sessions will then stop loading the object, remove the object marked for deletion from the buffer pool and use the object loaded successfully by the first session.</p> <p>The numbers of objects calculated by the counters "Object Loads - finished" and "Object Loads - concurrent" are usually identical. The numbers only differ if the concurrent load is only detected after both sessions have finished the load.</p>
Number Loads into BP	<p>The number of times a load into the buffer pool was performed successfully.</p> <p>The load into the buffer pool (storage allocation request) can be triggered either by a load from the database or by a load from the BP cache.</p>
Number Loads BP 2nd	<p>This field is displayed if METHOD=S (selection process) is used as search method for allocating storage.</p> <p>Alloc Requests 2nd shows the number of times a storage allocation request satisfied the search criteria of Algorithm 2 as described in METHOD=S in Buffer Pool Search Methods in the Natural Operations for Mainframes documentation.</p>
Number Loads into BP - Number Load Cycles	<p>This field is displayed if METHOD=N (next available) is used as search method for allocating storage as described in METHOD=N in Buffer Pool Search Methods in the Natural Operations for Mainframes documentation.</p> <p>This field indicates the number of times a search has been performed starting from the top of the buffer pool. This number gives an estimate of the frequency of cycling through the buffer pool in a wrap-around fashion.</p>

Field	Explanation
Number Loads into BP- Number Lock Retries	This field is displayed if METHOD=N (next available) is used as search method for allocating storage as described in METHOD=N in Buffer Pool Search Methods in the Natural Operations for Mainframes documentation.  This field indicates the number of times a chain of locked buffer pool entries had to be unlocked, because they could not satisfy the allocation request.
Number Loads into BP- Last Cycle Start	This field is displayed if METHOD=N (next available) is used as search method for allocating storage as described in METHOD=N in Buffer Pool Search Methods in the Natural Operations for Mainframes documentation.  The time and date when Number Load Cycles was last increased.
Largest Alloc (TR)	The largest single allocation size so far requested, specified in number of text records.
Number Load Failure	The total number of times an object load failed. The reason for a failure is that either all directory entries are in use at the time of the load request or not enough storage is available in the text record section to perform the load.
Number Load Failure - Sizes failing last	The number of text records that would have been required by the three most recent storage allocation requests that failed.
Locates/Loads	These statistics are expressed as a ratio using Total Locate Calls Successful and Object Loads. A value greater than 1 indicates that Natural located more objects in the buffer pool than it loaded from the system file. This ratio serves as a buffer pool efficiency indicator. The larger the number, the better the buffer pool is performing. This is the primary indicator of performance from one buffer pool session to the next.

For details on the search methods used for allocating space in the buffer pool, see Buffer Pool Search Methods under Natural Buffer Pool Principle of Operation in the Natural Operations for Mainframes documentation.

## Buffer Pool Fragmentation

This function provides an overview of the buffer pool fragmentation; that is, an overview of how many different Natural programming objects occupy how many text records, and how the object locations are spread over the buffer pool.

### To invoke Buffer Pool Fragmentation

- On the Buffer Pool Statistics main menu, enter Function Code **F**.  
Or, in the command line, enter DISPLAY FRAGMENTATION.

The Buffer Pool Fragmentation screen is displayed.

Some of the fields provided on the Buffer Pool Fragmentation are identical to the items explained in General Buffer Pool Statistics above:

- Buffer Pool Size
- Buffer Pool Address
- Text Record Section
- Text Record Size
- Number of Text Records  
(same as Text Records - Total)



The Hash Table Collisions screen is displayed.

The statistics displayed on the Hash Table Collisions screen are snapshots of the hash table which are taken every time you press ENTER. The following information is displayed:

Field	Explanation																				
Total Number of Slots	The total number of hash table slots; that is, the total possible entries in the hash table that link the object names with the location of the objects. The number of slots, that is, the size of the hash table will be calculated internally depending on the number of text records.																				
Number of Slots used	The number of slots in the hash table that have at least one object name mapped to them.																				
Number of Slots free	The number of slots in the hash table that have no object name mapped to them.																				
Max. Collisions per Slot	The maximum number of collisions of any slot. The maximum number of collisions is the longest possible search path for an object.  A collision is caused if the name of two different objects is mapped to the same slot by the hash algorithm. In this case, a collision resolution is used in order to find another slot.																				
Collisions	The number of current collisions. Depending on the collisions that occur, the table contains up to 10 rows: <table border="0" style="margin-left: 20px;"> <tr><td>0</td><td>No collision.</td></tr> <tr><td>1</td><td>1 collision.</td></tr> <tr><td>2</td><td>2 collisions.</td></tr> <tr><td>3</td><td>3 collisions.</td></tr> <tr><td>4</td><td>4 collisions.</td></tr> <tr><td>5</td><td>5 collisions.</td></tr> <tr><td>6 - 10</td><td>Between 6 and 10 collisions.</td></tr> <tr><td>11 - 15</td><td>Between 11 and 15 collisions.</td></tr> <tr><td>16 - 20</td><td>Between 16 and 20 collisions.</td></tr> <tr><td>21</td><td>More than 21 collisions.</td></tr> </table> <p>No collision means that only one object name is mapped per slot. To locate this object, you need to access the hash table once only.</p> <p>If the number of collisions is greater than zero (0), for example, <math>x, x+1</math> object names are mapped to the same slot. To locate one of these objects, you need to access the hash table up to <math>x+1</math>.</p>	0	No collision.	1	1 collision.	2	2 collisions.	3	3 collisions.	4	4 collisions.	5	5 collisions.	6 - 10	Between 6 and 10 collisions.	11 - 15	Between 11 and 15 collisions.	16 - 20	Between 16 and 20 collisions.	21	More than 21 collisions.
0	No collision.																				
1	1 collision.																				
2	2 collisions.																				
3	3 collisions.																				
4	4 collisions.																				
5	5 collisions.																				
6 - 10	Between 6 and 10 collisions.																				
11 - 15	Between 11 and 15 collisions.																				
16 - 20	Between 16 and 20 collisions.																				
21	More than 21 collisions.																				
Number of Slots	The number of slots related to the number of collisions.  In addition, the percentage of these slots related to all slots used is displayed.																				
Number of Slots Totaled	The same values as Number of Slots, but the values are totaled.																				

**Example of Hash Table Statistics**

```

14:36:26          ***** NATURAL SYSBPM UTILITY *****          2003-08-13
BPNAME NATGBP    - Buffer Pool Hash Table Statistics -          Type Global Nat
BPPROP OFF                                             Loc DAEF QA41

Total Number of Slots ..          523
Number of Slots used ..          475 ( 90.8 %)      Max. Collisions
Number of Slots free ..          48 ( 9.1 %)      per Slot ..... 7

      Collisions          Number of Slots      Number of Slots Totaled
      0                   0 ( 0.0 %)           0 ( 0.0 %)
      1                   164 ( 34.5 %)         164 ( 34.5 %)
      2                   194 ( 40.8 %)         358 ( 75.3 %)
      3                   96 ( 20.2 %)          454 ( 95.5 %)
      4                   16 ( 3.3 %)           470 ( 98.9 %)
      5                   4 ( 0.8 %)            474 ( 99.7 %)
      6 - 10              1 ( 0.2 %)           475 ( 100.0 %)

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help          Exit  Last          Flip          Canc
    
```

# SYSBPM - BP Cache Statistics

This function only applies to buffer pools of the type Natural.

BP Cache Statistics invokes the BP Cache Statistics main menu which is used to obtain statistical information on the BP cache.

Note that the BP Cache Statistics function can only be executed if a BP cache has been installed when initializing a global buffer pool (no BP cache support for local buffer pools).

## To invoke BP Cache Statistics

- On the SYSBPM main menu, enter Function Code **C**.  
Or, in the command line, enter `DISPLAY CSTATISTICS`.

The BP Cache Statistics main menu is displayed.

From the BP Cache Statistics main menu, you can select the following functions:

- General BP Cache Statistics
  - BP Cache Call Statistics
  - BP Cache Hash Table Statistics
- 

## General BP Cache Statistics

This function displays addresses and statistics regarding the activity of the BP cache.

### To invoke General BP Cache Statistics

- On the BP Cache Statistics main menu, enter Function Code **G**.  
Or, in the command line enter `DISPLAY CGENERAL`.

The General BP Cache Statistics screen is displayed.

The statistics displayed on the General BP Cache Statistics screen are snapshots of the buffer pool, which are refreshed each time you press `ENTER`. The following information is displayed:

Field	Explanation
Dataspace - Name	The name of the dataspace where the BP cache resides.
Dataspace - SToken	The term SToken (for Space Token) identifies a dataspace.
Dataspace - ALET	The term ALET (for Address List Entry Token) identifies an index for accessing the dataspace.
Dataspace - Size (MB)	The size of the BP cache in MB.
Dataspace - Current state	The status of the BP cache:  not initialized  locked for init  closed  free for operation  undefined
Dataspace - Initialization	The date and time when the BP cache was initialized.
Internal buffer offsets - Header buffer	The header of the BP cache which contains general BP cache information.
Internal buffer offsets - Hash buffer	Contains the hash table (see also BP Cache Hash Table Statistics below).
Internal buffer offsets - Directory buffer	The address of the BP cache directory section relative to the beginning of the BP cache. Each Natural programming object loaded in the BP cache requires a directory entry that contains information on this object. The space for these directory entries is acquired from the BP cache itself.
Internal buffer offsets - Text buffer	The address of the text buffer relative to the beginning of the BP cache. After allocating the space for all other buffers, the remaining space is divided into text records with a size of 4 KB. An object can occupy one or more text records, depending on its size.
Tot. Text Records	The total number of text records in the BP cache.  The number of text records depends on the BP cache size. The text record size for the BP cache is 4 KB.
Insert position	The index number of the text record into which the next object will be inserted.  Objects will be inserted into the BP cache when they have to be removed from the buffer pool.
Reuse cycles	The number of times the BP cache has been completely reused.  Every time the BP cache is full, the BP cache manager reuses the BP cache from the start and overwrites the object(s) from there. The objects will remain in the BP cache until the BP cache is used again.
Objects - Max Loaded	The maximum number of objects currently loaded in the BP cache.
Objects - Loaded	The number of objects currently loaded in the BP cache.

## BP Cache Call Statistics

This function provides statistical information on the loading (put), retrieving (get) and deleting of objects into/from the BP cache. This information also serves as an indicator of BP cache performance.

### To invoke BP Cache Call Statistics

- On the BP Cache Statistics main menu, enter Function Code **L**.  
Or, in the command line enter **DISPLAY CLOAD**.

The BP Cache Call Statistics screen is displayed.

The statistics displayed on the BP Cache Call Statistics screen are snapshots of the buffer pool which are refreshed each time you press ENTER. The following information is displayed:

Field	Explanation
Search calls	The number of Search calls the buffer pool sent to the BP cache while attempting to find an object in the BP cache.  If an object is found, a Search call results in a Get call.
Get calls (from BP cache)	The number of Get calls the buffer pool sent to the BP cache while attempting to load an object from the BP cache into the buffer pool.
Get calls - successful	The number of successful Get calls the BP cache performed, that is, the number of objects the BP cache swapped into the buffer pool.  A Get call is successful if an object the buffer pool attempted to load is actually loaded from the BP cache into the buffer pool.  A Get call is unsuccessful, for example, if an object was deleted after it was found by the Search call.
Put calls (to BP cache)	The number of Put calls the buffer pool sent to the BP cache while attempting to swap an object from the buffer pool into the BP cache.
Put calls - successful	The number of Put calls that resulted in an object to swapped from the buffer pool into the BP cache.
Put calls - obj. already cached	The number of Put calls the buffer pool sent to the BP cache for objects that were already loaded in the BP cache.
Delete calls	The number of Delete calls the buffer pool sent to the BP cache while attempting to delete an object from the BP cache.  A Delete call requests either a single object or a range of objects (see also the section Delete Objects).

Field	Explanation
Delete calls - successful	<p>The number of successful Delete calls the buffer pool sent to the BP cache.</p> <p>A Delete call is successful if at least one object is actually deleted from the BP cache.</p> <p>A Delete call is unsuccessful if the object requested was not loaded in the BP cache and hence could not be deleted.</p> <p>Compared with the total number of Delete calls, the number of successful Delete calls can be very low. This happens, for example, if several programming objects are cataloged with the CATALOG command. In this case, for every object cataloged successfully, Natural sends a Delete call to the BP cache. However, at the time the Delete call is sent, most of the cataloged objects are usually not loaded in the BP cache and the delete attempt fails.</p>
Get/Put Ratio	<p>The ratio of Get calls to Put calls the buffer pool sent to the BP cache.</p> <p>The value is calculated by dividing the successful Get calls by the successful Put calls. It shows the overall reuse factor.</p> <p>The higher the value, the better the BP cache efficiency.</p> <p>Example: A ratio of 5.70 indicates that an object loaded in the BP cache was, on average, returned to the buffer pool 5.7 times.</p>
Get/Search Percentage	<p>The percentage of successful Get calls compared with the total number of Search calls the buffer pool sent to the BP cache.</p> <p>The value shows the percentage of objects the buffer pool loaded from the BP cache not from the Natural system file FNAT or FUSER.</p> <p>The higher the value, the better the cache efficiency.</p> <p>Example: A value of 70% indicates that 70 % of all objects loaded into the buffer pool were retrieved from the BP cache and 30% were loaded from the system file.</p>
Initialization	The date and time when the BP cache was initialized.
Last reuse cycle	<p>The load date and time of the object that was most recently overwritten.</p> <p>An object is overwritten in the BP cache when its space has to be reused in order to load another object. The object that was loaded first in the BP cache will be swapped first. This means the load date and time of the object that has been in the BP cache longest corresponds to the date and time in "Last reuse cycle".</p>
Last access	The date and time when the buffer pool last accessed the BP cache.
Last Put (to BP cache)	The date and time when the buffer pool last sent a Put call to the BP cache.
Last Get (from BP cache)	The date and time when the buffer pool last sent a Get call to the BP cache.
Last Delete	The date and time when the buffer pool last sent a Delete call to the BP cache.

## BP Cache Hash Table Statistics

This function displays statistics about hash table slots and collisions per slot. The statistics determine the efficiency of the hash algorithm used.

**To invoke BP Cache Hash Table Statistics**

- On the BP Cache Statistics main menu, enter Function Code **H**.  
Or, in the command line enter DISPLAY CHASH.

The Cache Hash Table Collisions screen is displayed.

The statistics displayed on the Cache Hash Table Collisions screen are snapshots of the hash table which are refreshed every time you press ENTER. The following information is displayed:

Field	Explanation
Total Number of Slots	The total number of hash table slots; that is, the total possible entries that link the object name with the location of the object.  The number of slots, that is, the size of the hash table will be calculated internally depending on the number of text records.
Number of Slots used	The number of slots that have one or more entries.
Number of Slots free	The number of slots that have no entry.
Max. Collisions per Slot	The maximum number of collisions of all slots.  The maximum number of collisions is the longest possible search path for an object.
Collisions	The number of possible collisions.  <b>0</b> means no collision or one entry. When there are more than 5 collisions, the number of collisions will be specified in ranges (for example, 6 - 10).
Number of Slots	The number of slots grouped by their number of collisions.  For example, if the number of collisions is 3, the search algorithm must side step a maximum of 3 times to find an object. In addition, the percentage of these slots related to all slots used is displayed.
Number of Slots Totaled	The same values as Number of Slots, but the values are totaled.

# SYSBPM - Select Buffer Pool

SYSBPM provides functions for displaying the buffer pools defined for your Natural system environment and selecting a buffer pool other than the buffer pool used by your current Natural session at startup.

Below is information on the Select Buffer Pool function and the SYSBPM direct commands provided for displaying and resetting buffer pools:

- Invoking Select Buffer Pool
  - Display Buffer Pools
  - Reset Buffer Pool
- 

## Invoking Select Buffer Pool

### To invoke Select Buffer Pool

- On the SYSBPM Main Menu, enter Function Code **S**.  
Or, in the command line, enter **SELECT BP**.

The Select Buffer Pool window appears with the following information on your current buffer pool (global or local) and on all further global buffer pools currently available in your Natural system environment:

- **BPNAME:** The name of the buffer pool.
- **Type:** The type of the buffer pool, such as: Global Nat, Local Nat, Global Sort, Global DL/I, Edit (Editor) or Mon (Monitor).
- **Status:** The current status.
- **Preload list:** The name of the preload list (if loaded).
- **Address:** The address of the buffer pool.
- **Loc:** The location of the buffer pool indicated by HostID and SubsID (subsystem ID).

For further details on the fields, see SYSBPM Main Menu - Fields and Functions in Invoking SYSBPM.

- In Column **C**, enter any character to select a buffer pool of the type Natural, DL/I or Sort (all other buffer pools can only be displayed) and press **ENTER**.

The buffer pool specified is now defined in your current session and the contents of the buffer pool-related fields (as mentioned above) displayed on top of the SYSBPM Main Menu has changed accordingly.

Once you have selected a buffer pool from the Select Buffer Pool window, all SYSBPM functions apply to this buffer pool. Your Natural session itself, however, will continue to run with the startup buffer pool.

## Display Buffer Pools

### To display the buffer pools available in your Natural system

- In the command line, enter **DISPLAY BUFFERPOOL**.

The Display Buffer Pools window appears which provides the same information as described for the Select Buffer Pool window above.

## Reset Buffer Pool

 **To reset the buffer pool**

- In the command line, enter `RESET BUFFERPOOL`.

SYSBPM switches back to the buffer pool originally defined for your current Natural session and the contents of the buffer pool-related fields (as mentioned in Invoking Select Buffer Pool above) displayed on top of the SYSBPM Main Menu change accordingly.

Or, invoke the Select Buffer Pool function (see above) and select the startup buffer pool.

# SYSBPM - Blacklist Maintenance

This function is used to maintain a so-called blacklist of Natural programming objects, which are not to be executed and loaded into the buffer pool and, if they already are in the buffer pool, are to be deleted. If the BP cache is enabled, the Natural programming objects will also be deleted from the BP cache. The blacklist always applies to the buffer pool currently active.

As described below, on the blacklist you can maintain individual Natural programming objects and object sets which contain several Natural programming objects. In an object set, you specify the objects not to be executed and add a single set (instead of multiple individual objects) to the blacklist. You can also combine both: maintain objects individually or by sets. An object set is stored as a Natural programming object of the type Text.

For details on the blacklist, see the relevant section in Natural Buffer Pool in the Natural Operations for Mainframes documentation.

## To invoke Blacklist Maintenance

- On the SYSBPM Main Menu, enter Function Code **B**.  
Or, in the command line, enter BLACKLIST.

The Blacklist Maintenance menu is displayed.

The functions provided on the Blacklist Maintenance menu are listed and explained below:

- Maintain Blacklist
- List Object Sets
- Edit Object Set
- Add Object Set to Blacklist
- Delete Object Set from Blacklist

Also below is information on:

- Delete Object Set Text Member
  - Blacklist Maintenance in Batch Mode
- 

## Maintain Blacklist

This function invokes the Maintain Blacklist screen where you can display and maintain all Natural programming objects currently available on the blacklist.

### To invoke the Maintain Blacklist screen

- On the Blacklist Maintenance menu, enter Function Code **M**.  
Or, in the command line, enter DISPLAY BLACKLIST.

The Maintain Blacklist screen appears and displays the current blacklist.  
Press PF7 to scroll one page backward and PF8 to scroll one page forward.

Depending on the mode set when calling the Maintain Blacklist function earlier during a SYSBPM session, the Maintain Blacklist screen appears in Display Mode (default when initializing SYSBPM) or Add Mode. Use PF9 to switch from one mode to the other.

- Adding Objects
- Modifying Objects
- Deleting Objects

## Adding Objects

### To add objects to the blacklist

1. On the Blacklist Maintenance menu, enter Function Code **M**.  
Or, in the command line, enter DISPLAY BLACKLIST.

The Maintain Blacklist screen is displayed.

2. If required, press PF9 to switch to Add Mode.  
A screen with empty input fields appears.
3. In the relevant input fields, enter the name of the library where the objects are stored, the names of the objects and the corresponding database IDs (DBID) and file numbers (FNR).

If DBID and FNR are left blank, they will be taken from the current system file FUSER or FNAT in libraries whose names start with SYS (except the library SYSTEM).

If you want to clear the Add Mode screen, in the direct command line, enter CLE or CLEAR.

4. Press PF5 to confirm the addition.  
Or, in the command line, enter UP or UPDATE.

A corresponding message appears.

## Modifying Objects

### To modify objects on the blacklist

1. On the Blacklist Maintenance menu, enter Function Code **M**.  
Or, in the command line, enter DISPLAY BLACKLIST.

The Maintain Blacklist screen is displayed.

2. If required, press PF9 to switch to Display Mode and obtain the list of all objects currently on the blacklist.
3. In the relevant input field(s), replace the existing entries with new values.
4. Press PF5 to confirm the modification.  
Or, in the command line, enter UP or UPDATE.

A corresponding message appears.

## Deleting Objects

### To delete individual objects from the blacklist

1. On the Blacklist Maintenance menu, enter Function Code **M**.  
Or, in the command line, enter DISPLAY BLACKLIST.

The Maintain Blacklist screen is displayed.

2. If required, press PF9 to switch to Display Mode and obtain a list of all objects currently on the blacklist.
3. In Column **C**, next to the object(s) desired, enter the line command **DE**.
4. Press ENTER to confirm the deletion.  
A corresponding message appears.

 **To delete all objects from the blacklist**

1. On the Blacklist Maintenance menu, enter Function Code **M**.  
Or, in the command line, enter DISPLAY BLACKLIST.

The Maintain Blacklist screen is displayed.

2. Press PF2.  
The Confirm Delete window is displayed:
  - To execute the deletion, enter Y (Yes).
  - To cancel the deletion:  
Press PF3 without entering anything in the window.  
Or, enter or confirm N (No) which is the default.
3. Press ENTER to confirm the action. A corresponding message appears.

## List Object Sets

This function invokes the List Object Sets screen which displays a list of all existing object sets.

 **To invoke the List Object Sets screen**

- On the Blacklist Maintenance menu, enter Function Code **L**, a library name and an object set name. Asterisk (\*) notation is also allowed for an object set name.

Or, in the command line, enter LIST SET *library-name set-name*.  
Asterisk (\*) notation is also allowed for *set-name*.

The List Object Sets screen appears and displays the specified set(s).

You can manipulate an object set from the List Object Sets screen by using any of the line commands provided to modify a set and add it to or delete it from the blacklist. For a list of possible commands, enter a question mark (?) in any of the leftmost screen columns which contain the prefix information.

## Edit Object Set

This function invokes the Edit Object Set screen where you can create a new object set, add objects to an existing set or modify them, or delete objects from a set.

The editing functions provided on the Edit Object Set screen are a subset of the functions provided by the Software AG Editor as described in the relevant documentation. To invoke the Help window with a list of the commands available, in the command line, enter a question mark (?). Press PF7 to scroll backward and PF8 to scroll forward in the window.

For a list of the line commands available, in any of the leftmost columns (prefix information), type in a question mark (?).

Below is information on:

- Creating Object Sets
- Modifying Object Sets

## Creating Object Sets

### ▶ To create an object set

- On the Blacklist Maintenance menu:
  - Enter Function Code **E**.
  - Enter the name of a library.
  - Do **not** enter the name of an object set but clear the contents (if any) of the corresponding field.

The Edit Object Set screen is displayed.

- In the relevant input fields, enter the name of the library where the objects are stored, the names of the objects and the corresponding database IDs (DBID) and file numbers (FNR).  
If DBID and FNR are left blank, they will be taken from the current system file FUSER or FNAT in libraries whose names start with SYS (except the library SYSTEM).

In the command line, enter *SA set-name* to save the object set as Natural Text member.

## Modifying Object Sets

Below are the functions provided to add an object to an object set, to modify existing objects or to delete them from the set. Note that any of these object set modifications will **not** update the current blacklist.

### ▶ To add a new object to an object set

- On the Blacklist Maintenance menu, enter Function Code **E**, a library name and an object set name.

Or, on the List Object Sets screen, in the leftmost column, next to the object set desired, enter the line command **E**.

Or, in the command line, enter  
*EDIT SET library-name set-name*.

The Edit Object Set screen appears and displays the specified object.

- Complete the input fields by entering the name of the library where the objects are stored, the names of the objects and the corresponding database IDs (DBID) and file numbers (FNR).  
If DBID and FNR are left blank, they will be taken from the current system file FUSER or FNAT in libraries whose names start with SYS (except the library SYSTEM).

In the command line, enter *SA* to save the modification.

### ▶ To modify an object of an object set

- On the Blacklist Maintenance menu, enter Function Code **E**, a library name and an object set name.

Or, on the List Object Sets screen, in the leftmost column, next to the object set(s) desired, enter the line command **E**.

Or, in the command line, enter  
*EDIT SET library-name set-name*.

The Edit Object Set screen appears and displays the specified object set.

- In the relevant input field(s), replace the existing entries with new values.

In the command line, enter *SA* to save the modification.

 **To delete an object from an object set**

- On the Blacklist Maintenance menu, enter Function Code **E**, a library name and an object set name.

Or, on the List Object Sets screen, in the leftmost column, next to the object set(s) desired, enter the line command **E**.

Or, in the command line, enter  
EDIT SET *library-name set-name*.

The Edit Object Set screen appears and displays the specified object set.

- In the leftmost column, next to the object desired, enter the line command **D** and press ENTER.

In the command line, enter SA to save the modification.

## Add Object Set to Blacklist

This function is used to add all objects of an object set to the blacklist.

 **To add an object set to the blacklist**

- On the Blacklist Maintenance menu, enter Function Code **A**, a library name and an object set name.

Or, on the List Object Sets screen, in the leftmost column, next to the object set(s) desired, enter the line command **AC**.

Or, on the Edit Object Set screen, in the command line, enter **AC**.

Or, in the command line, enter ADD SET *library-name set-name*.

A message appears confirming that the object set was added to the blacklist.

**Note:**

The command **AC** denotes ACTIVATE which is the equivalent of Add Object Set to Blacklist.

## Delete Object Set from Blacklist

This function is used to delete all objects of an object set from the blacklist. Note that the Delete Object Set function will **not** delete the object set as a Natural Text member. The objects of the object set can be added to the blacklist again at any time, as described above. See also To delete an object set Text member below.

 **To delete an object set from the blacklist**

- On the Blacklist Maintenance menu, enter Function Code **D**, a library name and an object set name.

Or, on the List Object Sets screen, in the leftmost column, next to the object set(s) desired, enter the line command **DA**.

Or, on the Edit Object Set screen, in the command line, enter **DA**.

Or, in the command line, enter  
DELETE SET *library-name set-name*.

A message appears confirming that the object set was deleted from the blacklist.

**Note:**

The command **DA** denotes DEACTIVATE which is the equivalent of Delete Object Set from Blacklist.

## Delete Object Set Text Member

### ▶ To delete an object set Text member

- On the Blacklist Maintenance menu, enter Function Code **L**, a library name and an object set name.

Or, in the command line, enter LIST SET *library-name list-name*.

The List Object Sets screen is displayed.

- In the leftmost column, next to the object desired, enter the line command **D** and press ENTER. The Delete window appears.
- Confirm the deletion by entering the name of the object set. A corresponding confirmation message appears.

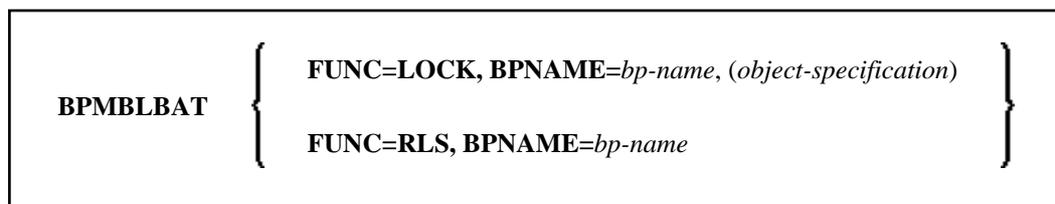
Note that the deletion of an object set Text member **will not** update the current blacklist.

## Blacklist Maintenance in Batch Mode

SYSBPM blacklist maintenance functions can also be executed in batch mode as described in SYSBPM - Batch Processing.

In addition, the Natural system library SYSBPM provides the command BPMBLBAT that further facilitates batch processing of the blacklist maintenance functions add and delete all objects.

Log on to the library SYSBPM and execute the BPMBLBAT command using the syntax indicated in the diagram below. The symbols used in the diagram are explained in the section System Command Syntax in the Natural System Command Reference documentation.



The following section contains information on:

- Explanation of Syntax
- Examples of Input

### Explanation of Syntax

Described below are the keywords and the *object-specification* clause indicated in the BPMBLBAT syntax diagram above. Examples of Input below demonstrate the use of the keywords.

FUNC=LOCK	Adds object names to a blacklist as described in Adding Objects (Maintain Blacklist) above.
FUNC=RLS	Deletes a complete blacklist with all object names contained in the list. This functions corresponds to delete function described under To delete all objects from the blacklist in Deleting Objects, Maintain Blacklist, above. (In batch mode, you cannot delete single object names from a blacklist.)
BPNAME	The name of the buffer pool where the blacklist is loaded.
<i>object-specification</i>	<p>The keywords that apply to the object-specification clause (see the respective syntax below) are:</p> <p><b>LIB</b>            The name of the library where the objects are stored.</p> <p><b>DBID</b>            The database ID (DBID) and the file number (FNR) where the objects are stored.</p> <p><b>FNR</b></p> <p>If DBID and FNR are left blank, they will be taken from the current system file FUSER or FNAT in libraries whose names start with SYS (except the library SYSTEM).</p> <p><i>object-name</i>    The name(s) of the object(s) to be added to the blacklist: Enter each name in a separate line in the positions 1 to 8. To indicate the end of the input, in a separate line, enter a period (.)</p>

**Syntax of Object-Specification**

Shown and explained below is the syntax that applies to *object-specification*:

```
LIB=lib-name, DBID dbid , FNR fnr [ object-name ]..
```

**Examples of Input**

**Example 1 - Adding Objects to a Blacklist**

The example input below demonstrates how to add to a blacklist the objects A, B and C in the buffer pool NATGBP:

```
/*Job
.
.
.
*/Job

LOGON SYSBPM
BPMBLBAT
FUNC=LOCK , BPNAME=NATGBP , LIB=SAGTEST , DBID=10 , FNR=32
A
B
C
.
FIN
```

**Example 2 - Deleting a Blacklist:**

The example input below demonstrates how to delete a blacklist in the buffer pool NATGBP:

```
/*Job
.
.
.
*/Job
LOGON SYSBPM
BPMBLBAT
FUNC=RLS , BPNAME=NATGBP
.
FIN
```

# SYSBPM - Preload List Maintenance

This function only applies to buffer pools of the type Natural.

Preload List Maintenance is used to maintain so-called preload lists. In a preload list, you can specify the names of Natural programming objects that are to be loaded into the buffer pool when the buffer pool is initialized.

The preload lists themselves are stored as Natural programming objects of the type Text in the library SYSBPM.

For further details on the preload list, see the relevant section in Natural Buffer Pool in the Natural Operations for Mainframes documentation.

## To invoke Preload List Maintenance

- On the SYSBPM Main Menu, enter Function Code **P**.  
Or, in the command line, enter PRELOADLIST.

The Preload List Maintenance menu is displayed.

The functions provided on the Preload List Maintenance menu are listed and explained below:

- List Preload Lists
- Edit Preload List
- Generate Preload List from Buffer Pool

Also below is information on:

- Delete Preload List
- 

## List Preload Lists

This function invokes the List Preload Lists screen which displays a list of all existing preload lists.

### To invoke List Preload Lists

- On the Preload List Maintenance menu, enter Function Code **L** and the name of a preload list.  
Asterisk (\*) notation is also allowed for a preload list name.

Or, in the command line, enter LIST PRELOADLIST *list-name*.  
Asterisk (\*) notation is also allowed for *list-name*.

The List Preload Lists screen is displayed.

For a list of possible line commands, enter a question mark (?) in any of the leftmost screen columns which contain the prefix information.

Further commands can be entered in the command line. To invoke the Help window with a list of commands available, enter a question mark (?) in the command line. Press PF7 to scroll backward and PF8 to scroll forward in the window.

## Edit Preload List

This function invokes the Edit Preload List screen where you can create a new preload list, add objects to an existing list or delete objects from it.

### Attention:

The editing functions provided on the Edit Preload List screen are a subset of the functions provided by the Software AG Editor (see the relevant documentation). Therefore, before you start a Natural session to edit a preload list, set the Natural profile parameter EDPSIZE to a value greater than zero (see also Profile Parameters in the Natural Parameter Reference documentation). We recommend that you set EDPSIZE to a minimum of 100.

To invoke the Help window with a list of the commands available, in the command line, enter a question mark (?). Press PF7 to scroll backward and PF8 to scroll forward in the window.

For a list of the line commands available, in any of the leftmost columns (prefix information), enter a question mark (?).

Below is information on:

- Creating Preload Lists
- Modifying Preload Lists

## Creating Preload Lists

### To create a preload list

- On the Preload List Maintenance menu:
  - Enter Function Code **E**.
  - Clear the contents of the field Preload List Name, that is, do **not** enter the name of a preload list.
  - In the fields Generation Library and Generation Objects, leave the default asterisk (\*).
 The Edit Preload List screen is displayed.
- In the relevant input fields, enter the name of the library where the objects are stored, the names of the objects and the corresponding database IDs (DBID) and file numbers (FNR).  
If DBID and FNR are left blank, they will be taken from the current system file FUSER or FNAT in libraries whose names start with SYS (except the library SYSTEM).  
The resident flag will be set to **Y** (Yes) in Column **R** on the editing screen if no value is entered. Resident means that the object is not deleted from the buffer pool, not even if its Use Count is set to **0**.

In the command line, enter `SA set-name` to save the object set as Natural Text member in the library SYSBPM.

See also Generate Preload List from Buffer Pool below.

## Modifying Preload Lists

### To add a new object to a preload list

- On the Preload List Maintenance menu, enter Function Code **E** and the name of a preload list.

Or, on the List Preload Lists screen, in the leftmost column, next to the preload list desired, enter the line command **E**.

Or, in the command line, enter `EDIT PRELOADLIST list-name`.

The Edit Preload List screen appears and displays the specified preload list.

- Complete the input fields by entering the name of the library where the objects are stored, the names of the objects and the corresponding database IDs (DBID) and file numbers (FNR).  
If DBID and FNR are left blank, they will be taken from the current system file FUSER or FNAT in libraries whose names start with SYS (except the library SYSTEM).

In the command line, enter SA to save the modification.

#### To modify an object of a preload list

- On the Preload List Maintenance menu, enter Function Code **E**, a library name and the name of a preload list.

Or, on the List Preload Lists screen, in the leftmost column, next to the object set desired, enter the line command **E**.

Or, in the command line, enter `EDIT PRELOADLIST list-name`.

The Edit Preload List screen appears and displays the preload list specified.

- In the relevant input field(s), replace the existing entries with new values.

In the command line, enter SA to save the modification.

#### To delete an object from a preload list

- On the Preload List Maintenance menu, enter Function Code **E**, a library name and the name of a preload list.

Or, on the List Preload Lists screen, in the leftmost column, next to the object set(s) desired, enter the line command **E**.

Or, in the command line, enter `EDIT PRELOADLIST list-name`.

The Edit Preload List screen appears and displays the preload list specified.

- In the leftmost column, next to the object desired, enter the line command **D** and press ENTER.

In the command line, enter SA to save the modification.

## Generate Preload List from Buffer Pool

This function is used to generate a new preload list by using the names of the objects currently loaded in the buffer pool. From the objects that are currently in the buffer pool, you can select those you wish to be included in the preload list.

#### To generate a preload list, use either of the options below

1. On the Preload List Maintenance menu, enter Function Code **G** and the name of a preload list, and, in the fields Library, Objects, Resident, Use Count and Total Use Count, specify the objects to be included in the list:
  - To include all objects that are currently in the buffer pool, enter an asterisk (\*) in the fields Library, Objects and Resident, and leave the fields Use Count and Total Use Count blank.
  - To include specified objects in the buffer pool, in the fields below enter the following values:

Library Objects	A single name or asterisk (*) notation.
Resident	An asterisk (*) for all objects or Y (Yes) for all objects currently marked as resident in the buffer pool.
Use Count Total Use Count	A numeric start <i>value</i> (>), for example >10. Selects all objects with a Use Count/Total Use Count greater than or equal to <i>value</i> are selected.

- Or, in the command line, enter  
GENERATE PRELOADLIST *list-name* or  
GENERATE PRELOADLIST *list-name gen-library*

(See also the explanations of field values above).

A message appears confirming that the preload list was generated from the buffer pool.

All preload list objects will be generated as resident (entry **Y** in Column **R**) by default. Choose manually, which objects you want to remove from the list.

Objects from the library SYSBPM will not be included in the generated preload list as it can be assumed that these are objects which were only loaded into the buffer pool in order to execute this function.

## Delete Preload List

### To delete a preload list

- On the Preload List Maintenance menu, enter Function Code **L** and the name of a preload list.  
Or, in the command line, enter LIST PRELOADLIST *list-name*.

The List Preload Lists screen is displayed.

- In the leftmost column, next to the object desired, enter the line command **D** and press ENTER.  
The Delete window appears.
- Confirm the deletion by entering the name of the preload list.  
A corresponding confirmation message appears.

# SYSBPM Direct Commands

The SYSBPM utility provides commands to directly execute SYSBPM functions or navigate in screens in online or batch mode.

SYSBPM direct commands that refer to the BP cache or buffer pool hash table only apply to buffer pools of the type Natural.

The SYSBPM direct commands listed in the table below can be entered in the command line of any SYSBPM screen. An underlined portion of a SYSBPM command represents an acceptable abbreviation. Letters in italics represent variable information. You must supply a valid value when specifying this term.

Command	Parameters	Function
<u>ADD</u> <u>BLACKLIST</u>	none	Invokes the Maintain Blacklist screen.
<u>ADD</u> <u>SET</u>	<i>library-name set-name</i>	Adds all objects of a specified object set to the blacklist as described in Add Object Set to Blacklist.
<u>BLACKLIST</u>	none	Invokes the Blacklist Maintenance menu.
<u>BOTTOM</u>	none	Scrolls to the end of a list.
<u>CANCEL</u>	none	Same as EXIT.
<u>CHECK</u> <u>HASH</u> <u>CHECK</u> <u>HT</u>	none	Checks the BP hash table for consistency and returns the number of inconsistencies found.  See also REBUILD HASH.
<u>CLOSE</u> <u>BPC</u>	none	BP cache required.  Invokes the function Close BP Cache. The buffer pool runs without BP cache afterwards. You can restart the BP cache by using the INITIALIZE BPC command.

Command	Parameters	Function
<u>DELETE</u>	none	Deletes all objects from the buffer pool and the BP cache (BPC).  If entered on the Directory Information screen: see DELETE under Commands in the relevant section.
<u>DELETE</u>	<i>library-name object-name dbid fnr</i>	Deletes the specified object(s) from the buffer pool and the BP cache (BPC) as described in Delete Objects.
<u>DELETE ALL</u>	none	Deletes all objects from the blacklist as described in Delete Object from Blacklist.
<u>DELETE BUFFERPOOL</u> <u>DELETE BP</u>	none	Deletes all objects from the buffer pool only.
	<i>library-name object-name dbid fnr</i>	Deletes the specified object(s) from the buffer pool only.
<u>DELETE BPC</u>	none	BP cache required.  Deletes all objects from the BP cache (BPC) only.
	<i>library-name object-name dbid fnr</i>	BP cache required.  Deletes the specified object(s) from the BP cache (BPC) only.
<u>DELETE BLACKLIST</u>	none	Invokes the Maintain Blacklist screen where you can delete blacklist entries.
<u>DELETE SET</u>	<i>library-name set-name</i>	Deletes all objects of the specified object set from the blacklist as described in Delete Object Set from Blacklist.
<u>DISPLAY ALL</u>	none	Same as DISPLAY LIST.
<u>DISPLAY BUFFERPOOL</u> <u>DISPLAY BP</u>	none	See Display Buffer Pools in Select Buffer Pool.

Command	Parameters	Function
<u>DISPLAY</u> <u>BLACKLIST</u>	none	Invokes the Maintain Blacklist screen.
<u>DISPLAY</u> <u>CDIRECTORY</u>	none	BP cache required. Invokes the Directory Information screen.
<u>DISPLAY</u> <u>CGENERAL</u>	none	BP cache required. Invokes the General BP Cache Statistics screen.
<u>DISPLAY</u> <u>CHASH</u>	none	Invokes the function BP Cache Hash Table Statistics and displays the Cache Hash Table Collisions screen.
<u>DISPLAY</u> <u>CLIST</u>	<i>library-name object-name dbid fnr</i>	BP cache required. Invokes the List Objects screen.  In contrast to the command <u>DISPLAY LIST</u> , this command generates a statistics report that displays data about BP cache objects at the beginning of the list.
<u>DISPLAY</u> <u>CLOAD</u>	none	BP cache required. Invokes the BP Cache Call Statistics screen.
<u>DISPLAY</u> <u>CSTATISTICS</u>	none	BP cache required. Invokes the BP Cache Statistics Main Menu.
<u>DISPLAY</u> <u>DIRECTORY</u>	<i>library-name object-name dbid fnr</i>	Invokes the Directory Information screen.
<u>DISPLAY</u> <u>FRAGMENTATION</u>	none	Invokes the Buffer Pool Fragmentation screen.
<u>DISPLAY</u> <u>FUNCTION</u>	none	Invokes the Internal Function Usage screen.
<u>DISPLAY</u> <u>GENERAL</u>	none	Invokes the General Buffer Pool Statistics screen.

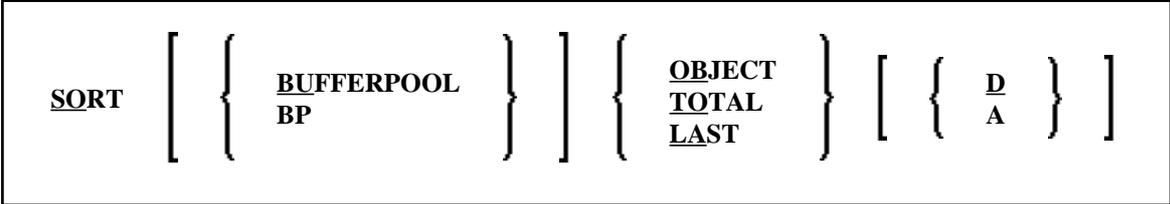
Command	Parameters	Function
<u>D</u> ISPLAY <u>H</u> ASH <u>D</u> ISPLAY HT	none	Invokes the function Buffer Pool Hash Table Statistics and displays the Hash Table Collisions screen.
<u>D</u> ISPLAY <u>H</u> DIRECTORY	<i>library-name object-name dbid fnr</i>	Invokes the Directory Information Hex screen that displays in hexadecimal format the directory information of an object.
<u>D</u> ISPLAY <u>H</u> EX	<i>library-name object-name dbid fnr</i>	Invokes the Hexadecimal Display screen that displays in hexadecimal format the source of an object.
<u>D</u> ISPLAY LIST	<i>library-name object-name dbid fnr</i>	Invokes the List Objects screen.  In contrast to the command DISPLAY CLIST, this command generates a statistics report that displays data about buffer pool objects at the beginning of the list.
<u>D</u> ISPLAY <u>L</u> OAD	none	Invokes the Buffer Pool Load/Locate Statistics screen.
<u>D</u> ISPLAY <u>S</u> TATISTICS	none	Invokes the BP Statistics Main Menu.
<u>E</u> DIT <u>P</u> RELOADLIST	<i>list-name</i>	Invokes the Edit Preload List screen.
<u>E</u> DIT <u>S</u> ET	<i>library-name set-name</i>	Invokes the Edit Object Set screen as described in Blacklist Maintenance.
<u>E</u> XIT	none	Leaves the current function/screen and displays the previous screen.
FLIP	none	Switches the PF-key line, if applicable.
<u>G</u> ENERATE <u>P</u> RELOADLIST	<i>list-name gen-library</i>	Invokes the function Generate Preload List from Buffer Pool.

Command	Parameters	Function
INITIALIZE	none, 1, 2, 4, 8, 12, 16	<p>Reinitializes the buffer pool and the BP cache. If no text record size is specified, the current text record size will be taken.</p> <p>Only use this function if the Current Use Count (see Directory Information) is equal to zero (see Warning below) or if the buffer pool has been destroyed.</p> <p><b>Warning:</b> If you try to reinitialize the buffer pool while objects are being executed by active sessions in this buffer pool, the window Confirm Initialization appears with the Current Use Count for this buffer pool (not counting the SYSBPM user himself). If Current Use Count is <b>not</b> equal to zero and you enter <b>Y</b> to confirm the reinitialization of the buffer, the results of the active sessions are unpredictable and Natural can even abend.</p>
INITIALIZE BP	none, 1, 2, 4, 8, 12, 16	<p>Reinitializes the buffer pool only. If no text record size is specified, the current text record size will be used.</p> <p>See also Warning above.</p>
INITIALIZE BPC	none	<p>BP cache required.</p> <p>Reinitializes the BP cache only. The text record size of the BP cache is fixed (4 KB).</p>

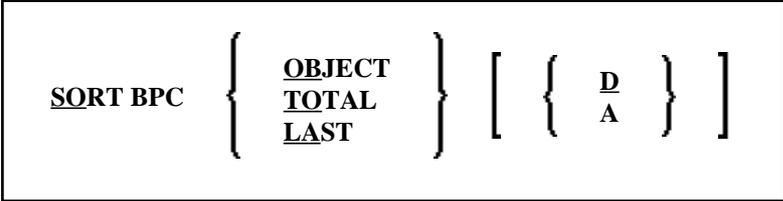
Command	Parameters	Function
LAST	none	Displays the SYSBPM direct command entered most recently. The equivalent PF key is PF4.
<u>LIST</u> <u>PRELOADLIST</u>	<i>list-name</i>	Invokes the List Preload Lists screen for the specified object.
<u>LIST</u> <u>SET</u>	<i>library-name set-name</i>	Invokes the List Object Sets screen for the specified library or object as described in Blacklist Maintenance.  Asterisk (*) is also allowed for <i>set-name</i> .
MENU	none	Invokes the SYSBPM Main Menu as described in Invoking and Operating SYSBPM.
<u>PRELOADLIST</u>	none	Invokes the Preload List Maintenance menu.
QUIT	none	Same as EXIT.
<u>REBUILD</u> <u>HASH</u> <u>REBUILD</u> <u>HT</u>	none	This function is used to rebuild hash tables if inconsistencies are found with CHECK HASH.  REBUILD HASH deletes the current hash table and rebuilds a new hash table from the current buffer pool contents.
<u>RESET</u> <u>BUFFERPOOL</u> <u>RESET</u> <u>BP</u>	none	See Reset Buffer Pool in Select Buffer Pool.
<u>SELECT</u> <u>BUFFERPOOL</u> <u>SELECT</u> <u>BP</u>	none	Only applies to buffer pools of the type Natural, DL/I or Sort.  See Select Buffer Pool.
<u>SORT</u> <u>BPC</u>	(syntax below)	Sorts the BP cache as described in Display Sorted Extract.

Command	Parameters	Function
<u>Sort</u> <u>BufferPool</u>	(syntax below)	Sorts the buffer pool as described in Display Sorted Extract.
STOP	none	Leaves the SYSBPM utility.
<u>TOP</u>	none	Scrolls to the beginning of a list.
<u>WRITE BP</u> <u>WRITE BPC</u> <u>WRITE ALL</u>		Writes object directory data to a local file or a PC text file.  See also Write to Work File.
+	none	Scrolls one page down in a list.
-	none	Scrolls one page up in a list.

# SORT



# SORT BPC



# SYSBPM - Batch Processing

The functions provided by the SYSBPM utility can also be executed in batch mode.

For this purpose, we recommend that you use the SYSBPM programming interface USR4340 or USR0340N described in the section Programming Interfaces.

You can also use the SYSBPM utility in batch by simulating the online input command sequence.

Since SYSBPM uses functionality of the Software AG Editor, when writing your batch job, observe the following instructions:

- Set the profile parameter EDPSIZE.  
  
Or set the profile parameter BPI with TYPE=EDIT for the Software AG editor buffer pool and define the editor work file in the batch job
- Note that the field Function Code is not available on every SYBPM screen.
- Note that you may have to skip input fields in order to position the cursor in the command line input field for entering direct commands.
- To navigate through the SYSBPM utility, simulate PF keys by using the terminal commands %K.  
For example, use %K3 to leave the List Objects screen.
- To enter more than one line of input for a map, use the continuation character defined with the session parameter CF (default is %).

Listed below are the topics and documentation sources that refer to the instructions mentioned above:

- [Related Topics](#)
- 

## Related Topics

Topic	Documentation
Blacklist Maintenance in Batch Mode	Blacklist Maintenance, SYSBPM documentation
SYSBPM Direct Commands	SYSBPM documentation
EDPSIZE	Profile Parameters, Natural Parameter Reference
BPI	Profile Parameters, Natural Parameter Reference
Installing the Software AG Editor	Natural Installation Guide for Mainframes
%K and %KP	Natural Terminal Commands
EDBP	Software AG Editor Buffer Pool Definitions, Natural Parameter Reference
Editor Buffer Pool	Operating the Software AG Editor, Natural Operations for Mainframes
Editor Work File	Operating the Software AG Editor, Natural Operations for Mainframes
Natural in Batch Mode	Natural Operations for Mainframes
Using the INPUT Statement in Non-Screen Modes	Natural Statements documentation
Using the INPUT Statement in Batch Mode on Mainframe Computers	Natural Statements documentation

# SYSBPM - Appl. Programming Interfaces

This section describes the application programming interfaces USR0340N, USR0341N and USR4340N which are used for handling Natural programming objects currently loaded in the buffer pool (BP) and/or BP cache. The application programming interfaces (APIs) are supplied in the Natural system library SYSEXT.

For further information on the interfaces see:

- The relevant Natural text members and the example programs in the Natural system library SYSEXT, and
- The SYSBPM functions referenced in the table below.

## Related topic:

SYSBPM Batch Processing

API	Functionality
USR0340N	<ul style="list-style-type: none"> <li>• Deletes objects from the buffer pool and/or BP cache.</li> <li>• Marks objects as resident.</li> <li>• Removes the resident flag from objects.</li> <li>• Reads object directory information.</li> <li>• Retrieves general buffer pool statistics and buffer pool load/locate statistics.</li> </ul> <p>Corresponding SYSBPM functions: Delete Objects List Objects Directory Information General Buffer Pool Statistics Buffer Pool Load/Locate Statistics.</p>
USR0341N	<p>Collects garbage to clean up the buffer pool by removing objects which are no longer needed.</p> <p>Selection criteria for specified objects are the relative age of an object. Relative age is the time the object has been loaded in the buffer pool which calculates from BP Last Activity date. Minimum age is 30 minutes.</p> <p>See also the SYSBPM function List Objects.</p>
USR4340N	<p>We recommend that you use this interface for batch processing instead of using the SYSBPM utility in batch.</p> <p>Lists objects loaded in the buffer pool and/or BP cache sorted by Object Size (ObjSize), Total Use Count (TotalUC) or BP Last Action. BP Last Action only applies to the buffer pool.</p> <p>Corresponding SYSBPM functions: Display Sorted Extract List Objects</p>

# SYSDDM Utility

The utility SYSDDM is used to create and maintain Natural data definition modules (DDMs).

The SYSDDM utility documentation covers the following topics:

- DDMs
- SYSDDM and Predict
- Invoking SYSDDM
- Edit DDM
- Read DDM
- Catalog DDM
- Delete DDM
- List DDMs
- List DDMs with Additional Information
- Show Defined DBIDs and Used FNRs
- Copy DDM to Another FDIC File
- Generate DDM from Adabas FDT

## Related documentation:

- SQL Services (DDM Generation), Natural for DB2.
  - DL/I Services, Natural for DL/I.
  - Adabas documentation
- 

## DDMs

For general information on DDMs, refer to the Natural Programming Guide.

A Natural application can only access a database file if a corresponding DDM has been created and cataloged for the file.

Cataloged DDMs are stored in the Natural system file FDIC.

A DDM can be created either with the SYSDDM utility (as described in this section) or with Predict (as described in the Predict documentation).

## SYSDDM and Predict

If Predict is installed at your site, you should **not** use SYSDDM; instead, it is recommended that you use the functions offered by Predict for the creation and maintenance of DDMs.

With Predict, it is possible to control the availability of SYSDDM. It may therefore be that the use of SYSDDM has been restricted and certain SYSDDM functions are not available to you. See the Predict documentation for further information.

## Invoking SYSDDM

 To invoke the SYSDDM utility

- In the direct command line, enter SYSDDM.

The SYSDDM Menu is displayed:

```

13:52:14          ***** NATURAL SYSDDM UTILITY *****          2002-07-02
User SAG          - Menu -          FDIC (10,460)
                                     Work area empty

          DDM Maintenance          List/Copy Services

          E Edit DDM          L List DDMs
          R Read DDM          X List DDMs with Additional Information
          C Catalog DDM          S Show Defined DBIDs and Used FNRs
          U Delete DDM          M Copy DDM to Another FDIC File
          ? Help
          . Exit

                                     Other Services

                                     G Generate DDM from Adabas FDT
                                     B SQL Services
                                     D DL/I Services

Code ..... _          FDIC Type ..... A
DDM Name .. _____          DDM Type ..... _
FNR ..... 0          DBID .. 0          Adabas Password ..
Replace ... N          DBID Type ..... 7

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

## Overview of Functions

The SYSDDM Menu is organized in the sections DDM Maintenance, List/Copy Services and Other Services and provides the following functions:

Function	Explanation
Edit DDM	Reads a DDM from the system file FDIC into the SYSDDM work area, and invokes the DDM editor.
Read DDM	Reads a DDM from the system file FDIC into the SYSDDM work area but does not invoke the DDM editor.
Catalog DDM	The DDM currently in the SYSDDM work area is cataloged, making it available for use within Natural applications. The DDM must have been placed in the work area by the function Generate DDM from Adabas FDT, or have been entered by using the function Edit DDM. For a VSAM DDM (DDM Type = V), SYSDDM prompts you for additional information; for details, see the Natural for VSAM documentation, Natural File Access.
Delete DDM	Deletes a cataloged DDM from the system file FDIC.
List DDMs	Displays a list of the DDMs stored in the specified FDIC system file. From the list, you can select individual DDMs for further processing. This function corresponds to the system command LIST DDM (see LIST in the Natural System Command Reference documentation).
List DDMs with Additional Information	Displays a list of the DDMs stored in the specified FDIC system file. From the list, you can select individual DDMs for further processing. This function differs from the List DDMs function in that it displays additional items of information on the individual DDMs.
Show Defined DBIDs and Used FNRs	Shows you which DBIDs are defined, as well as all file numbers of a given DBID for which DDMs have been defined.
Copy DDM to Another FDIC File	Copies a DDM from one FDIC system file to another.
Generate DDM from Adabas FDT	Generates a DDM from an Adabas field description table (FDT) and places it in the SYSDDM work area for further processing.
SQL Services	This function is available only if Natural for DB2 or Natural for SQL/DS is installed. It is used to generate DDMs from DB2 or SQL/DS tables and is described in DDM Generation in the documentation Natural for DB2.
DL/I Services	This function is available only if Natural for DL/I is installed. It is used to maintain the Natural for DL/I environment. Functions are provided for inquiry into and modification of structures, such as DL/I Database Descriptions (DBDs), Program Specification Blocks (PSBs), Program Communication Blocks (PCBs), DDMs and segment layouts. The function DL/I Services is described in the documentation Natural for DL/I.

The following parameters can be specified on the SYSDDM Menu for the various functions:

Parameter	Explanation
DDM Name	The name of the DDM to be processed. To process multiple DDMs, you can use asterisk notation for the name.
FNR	The file number of the database file for which the DDM is (to be) defined.
DBID	The database which contains the file for which the DDM is (to be) defined.
Replace	Y A DDM which is being copied or cataloged will replace an existing DDM of the same name. N Existing DDMs are not replaced.
FDIC Type (display only)	The database type of the system file. Possible types are the same as for DDM Type (see below).
DDM Type	The type of DDM. Possible types are:  A Adabas V VSAM 2 DB2 D DL/I P Entire System Server C Command Processor S Super Natural E Entire DB Engine
Adabas Password	The password required by Adabas if Adabas Security is installed.
DBID Type (display only)	The database type of the database specified in the DBID field. Possible types are the same as for DDM Type (see above). Exception: for an Adabas database, the Adabas version (for example, 5, 6 or 7) is displayed.

## Edit DDM

This function reads a DDM from the system file FDIC, places it into the SYSDDM work area and invokes the DDM editor.

When you modify a DDM, all programming objects which reference this DDM must be cataloged again.

Below is information on:

- DDM Editor
- Field Attributes
- DDM Editor Commands
- Extended Field Editing

## DDM Editor

### ▶ To invoke the DDM Editor for modifying or creating a DDM

- On the SYSDDM Menu, enter Function Code **E** and the name of an existing DDM.

Or, to create a new DDM, on the SYSDDM Menu, enter Function Code **E** and do not specify a DDM name. If the work area contains any data after the DDM editor is invoked, enter **CLEAR** in the direct command line.

The Edit DDM screen is displayed, similar to the one below:

```

16:03:29          ***** Edit DDM (ADA) *****          2002-07-02
DDM Name EMPLOYEES          Def.Seq.          DBID          0 FNR    316
Command
I T L DB Name          F          Leng S D Remark
----- top -----
  1 AA PERSONNEL-ID          A          8    D
*          C=NNNNNNN
*          C=COUNTRY
G 1 AB FULL-NAME
  2 AC FIRST-NAME          A          20 N
  2 AD MIDDLE-I          A          1 N
  2 AE NAME          A          20 D
  1 AD MIDDLE-NAME          A          20 N
  1 AF MAR-STAT          A          1 F
*          M=MARRIED
*          S=SINGLE
*          D=DIVORCED
*          W=WIDOWED
  1 AG SEX          A          1 F
  1 AH BIRTH          D          6 D
  1 AH N$BIRTH          I          2 D
G 1 A1 FULL-ADDRESS
DDM EMPLOYEES read into source area.

```

If you enter the command **HELP** (or a question mark) in the command line, the editor help information is displayed.

Title and header of the DDM editor contain the following information:

<p>Edit DDM (<i>DDM-type</i>)</p>	<p>The value displayed in parentheses next to the map title Edit DDM denotes the type of DDM.</p> <p>Possible types are:</p> <p>ADA          Adabas</p> <p>VSAM</p> <p>DB2</p> <p>DL/I</p> <p>PROCESS    Entire System Server</p> <p>CMD-PROC   Command Processor</p> <p>SNAT        Super Natural</p> <p>ENTIREDB   Entire DB Engine</p>
<p>DDM Name</p>	<p>The name used to reference the DDM in a Natural program. The name must be unique within the specified Natural system file.</p>
<p>Def. Seq.</p>	<p>The default sequence by which the file is read when it is accessed with a READ LOGICAL statement in a Natural program. See also the Natural READ statement as described in the Natural Statements documentation.</p>
<p>DBID</p>	<p>The database which contains the file to be accessed with the DDM.</p> <p>If 0 (zero) is specified, the default DBID for the Natural user system file (FUSER) as defined in the Natural parameter module is used.</p>
<p>FNR</p>	<p>The number of the file being referenced.</p> <p>If an Adabas file is used, the Adabas file number must be entered.</p> <p>If a DL/I segment type is used, the file number specified is used internally by Natural for DL/I.</p> <p>For VSAM files, see the Natural for VSAM documentation.</p>

### Field Attributes

The DDM itself comprises the following field definition attributes which can be entered or modified:

Attribute	Explanation
<p>I</p>	<p>Line indicator.</p> <p>This field is used by the DDM editor to mark lines.</p> <p>E    Lines containing an error detected during execution of a CHECK command.</p> <p>S    Lines containing a scanned value.</p> <p>X/Y Lines selected for copy/move operation.</p>

Attribute	Explanation
T	<p>Field Type:</p> <ul style="list-style-type: none"> <li>G Group header</li> <li>M Multiple-value field</li> <li>P Periodic group header</li> <li>* Comment line</li> <li><i>blank</i> Elementary field</li> </ul> <p><b>Note:</b> Groups defined in a DDM need not necessarily be defined as groups in the Adabas FDT.</p>
L	<p>Level number assigned to the field.</p> <p>Valid level numbers are 1 - 7.</p> <p>Level numbers must be specified in consecutive ascending order.</p>
DB	<p>For Adabas files, the Adabas two-character field name.</p> <p>For DL/I segment types, the 2-character code which is used in DL/I.</p> <p>For VSAM files, see the documentation Natural for VSAM.</p>
Name	<p>An external field name of 3 to 32 characters.</p> <p>This is the name used within Natural programs to reference the field.</p>
F	<p>Field format.</p> <p>For valid formats, refer to Definition of Format and Length in User-Defined Variables (Natural Statements documentation).</p>
Leng	<p>Standard field length.</p> <p>This length can be overridden in a Natural program.</p> <p>For numeric fields (format N), the length is specified as <i>nn.m</i>, where <i>nn</i> represents the number of digits before the decimal point and <i>m</i> represents the number of digits after the decimal point.</p> <p>For dynamic fields, the length may be specified as DYNAMIC.</p>
S	<p>Null-value suppression option (only for Adabas files):</p> <ul style="list-style-type: none"> <li>N Indicates that the field is defined with the Adabas null-value suppression option. This means that null values for the field are not stored in the inverted list and are not returned when the field is used in the WITH clause of a FIND statement, or in a HISTOGRAM or READ LOGICAL statement. If the Remarks column contains NC (not counted), an N in this column indicates that the field is defined with the SQL null-value option. Below this field, the corresponding null-indicator field is listed.</li> <li>M Indicates that the field is defined with the SQL null-value option "not null". The Remarks column for this field contains "NN NC" (not null, not counted). Below this field, the corresponding null-indicator field is listed.</li> <li>F Indicates that the field is defined with the Adabas fixed-storage option.</li> </ul>

Attribute	Explanation
D	<p>Descriptor Option.</p> <p>A Indicates that the field is an alternate index for a VSAM file.</p> <p>D Indicates that the field is an Adabas descriptor.</p> <p>H Indicates that the field is an Adabas hyperdescriptor. A hyperdescriptor is a user exit in Adabas and has in Natural the same functionality as a phonetic descriptor.</p> <p>N Indicates that the field is defined as a non-descriptor. A non-descriptor is not a descriptor, but can be used as a search field for a so-called non-descriptor search.</p> <p>P Indicates that the field is an Adabas phonetic descriptor.</p> <p>S Indicates that the field is an Adabas superdescriptor. If a superdescriptor contains a multiple-value field or a field from a periodic group (or part of such a field), the superdescriptor is marked with an <b>M</b> or a <b>P</b> in the Field Type column; this enables Natural to create the correct search algorithms for this superdescriptor. For a DL/I segment type, <b>S</b> indicates a superdescriptor; that is, a search field of a parent segment.</p> <p>U Indicates that the field is an Adabas subdescriptor or Adabas collation descriptor.</p> <p>If a subdescriptor contains a multiple-value field or a field from a periodic group (or part of such a field), you have to mark the subdescriptor with an <b>M</b> in the Field Type column. This enables Natural to create the correct search algorithms for this subdescriptor.</p> <p>A collation descriptor is used to sort (collate) descriptor field values in a non-standard sequence. If a field is a collation descriptor, the Remark column (see below) reads: Collation, the number of the Adabas user exit that contains the collation sequence (1-8) and the short name of the parent field to which the collation sequence applies, for example, Collation 5 on AA.</p> <p>X Indicates an alternate subdescriptor or superdescriptor; that is, an alternate index for a VSAM file.</p> <p>For VSAM files, see the Natural for VSAM documentation.</p>
Remarks	A comment which applies to a field and/or the DDM.

## DDM Editor Commands

Most of the editor and line commands available with the Natural program editor are also available in the DDM editor. Not available are special commands, such as PROFILE, RENUMBER, SET, SHIFT and some line commands. For more details on editor commands, refer to the relevant section in Program Editor in the Natural Editors documentation.

The following editor commands are also available:

### CATALOG

```
CATALOG [ DDM-name ] [ REPLACE ]
```

Catalogs the DDM in the work area. If the DDM definition is already cataloged, the replace option must be used.

**CHECK**

**CHECK**

Validates the DDM in the work area against the Adabas FDT. Should any inconsistency occur, the field definition causing the error is marked for correction.

**CLEAR**

**CLEAR**

Clears the work area.

**HELP**

**{ HELP }**  
**{ ? }**

Displays editor help information.

**LENGTH / SIZE**

**{ LENGTH }**  
**{ SIZE }** [ *from-field to-field* ]

Calculates the maximum length for one record in bytes. If you specify *from-field* and *to-field*, only the length from *from-field* to *to-field* is calculated.

**LIST DDM**

**LIST DDM [ DDM-name ]**

Lists another DDM without leaving the DDM editor (corresponds to the system command LIST DDM).

**READ**

**READ [ DDM-name ]**

Reads a DDM into the work area. Any DDM currently in the work area is overwritten.

**QUIT**

```
{ QUIT }
```

Leaves the DDM editor. The DDM in the work area is still available until another DDM is read into the work area, the work area is used otherwise (for example, by the program editor) or the Natural session is terminated.

**UNCAT**

```
UNCAT [ DDM-name ]
```

Deletes either the DDM currently in the work area or an optionally specified DDM from the current library.

**Extended Field Editing**

With the DDM editor, you can also modify individual field attributes. You can specify default options for field headers and edit masks (as well as additional field definitions specific to VSAM files).

**To invoke the extended editing mode**

- Next to the field desired, position your cursor at Column **T** and type in the line command **.E** over the values in Columns **T** and **L**.

The Extended Field Editing screen for the field marked with the command is displayed:

```
17:38:04          ***** Edit DDM (ADA) *****          2002-07-02
                - Extended Field Editing -
DDM Name EMPLOYEEES          Def.Seq.          DBID          0 FNR          316

I T L DB Name          F          Leng S D
----- top -----
  1 AA PERSONNEL-ID          A          8          D
-----

Remark .....
Field Header ..... PERSONNEL/ID
Field Edit Mask ..
```

On the Extended Field Editing screen, you can specify field headers, edit masks and comments (remarks) to be applied when the field is used in a DISPLAY or INPUT statement, as well as further specifications for VSAM DDMs. All the other information specific to the field (field type, length, name, format, remarks) can also be modified on this screen.

When you press ENTER on this screen (with or without having entered anything), you will return to the Edit DDM screen.

You can select a range of field definitions for editing by entering **.Ennn** where **nnn** is the number of fields to be selected.

For extended field editing in VSAM DDMs, see the Natural for VSAM documentation.

## Read DDM

This function reads a DDM from the system file FDIC and places it into the SYSDDM work area. However, unlike the function Edit DDM, Read DDM does not invoke the DDM editor.

## Catalog DDM

To catalog a DDM, you either select the function Catalog DDM on the SYSDDM Menu or enter the command CATALOG in the command line of the DDM editor.

For this function, you have to specify the DDM name and file number (FNR).

If the DBID is not entered, it is generated dynamically at execution time based on the DBID of the Natural user system file (FUSER) in use (see also the UDB profile parameter in the Natural Parameter Reference documentation).

For additional options for VSAM files, see the documentation Natural for VSAM.

## Delete DDM

This function is used to delete a single cataloged DDM from the system file FDIC.

You have to specify the name of the DDM to be deleted. You are then asked to confirm the deletion on a subsequent screen.

To delete multiple DDMs, you specify a DDM name with asterisk notation. This will automatically invoke the function Delete DDMs of the SYSMAIN utility (see DDMs, as described in the section The SYSMAIN Utility in the Natural Utilities documentation).

The contents of the SYSDDM work area is not affected by the deletion.

When you delete a DDM with SYSDDM, the corresponding Natural Security file profile is automatically deleted, too.

## List DDMs

This function corresponds to the system command LIST DDM. It displays a list of the DDMs stored in the specified FDIC system file.

From the list, you can select individual DDMs for further processing.

To select a DDM from the list, you mark it with a command in the Cmd column. For information on possible commands, you enter a question mark (?) in the Cmd column.

For information on all options available with the system command LIST, see the relevant section in the Natural System Command Reference documentation.

## List DDMs with Additional Information

This function displays a list of the DDMs stored in the specified FDIC system file.

From the list, you can select individual DDMs for further processing.

To select a DDM from the list, you mark it with a command in the **C** column. For information on possible commands, you enter a question mark (?) in the **C** column.

For each DDM listed, the following information is displayed:

- Database ID, file number, DDM type, DDM length in bytes;
- Security type (only under Natural Security):  
Public, Private, Access or Undef(ined);
- File type: Log.View, Phy.File or Log.File for VSAM DDMs;  
Userfile for Super Natural DDMs;
- VSAM name;
- Remarks; for example, SupNat (for a Super Natural DDM)  
or the VSAM file organization (KSDS, RRDS, ESDS or VRDS).

## Show Defined DBIDs and Used FNRs

When you invoke this function, a menu will be displayed from which you can select the following functions:

- Database IDs Defined in Natural
- File Numbers of Existing DDMs for a Database

### Database IDs Defined in Natural

This function displays a list of all DBIDs defined in NTDB macros of the Natural parameter module, sorted by database types (the NTDB macro is described in the Natural Parameter Reference documentation).

The default database type is shown at the top of the screen. DBIDs of all database types, except the default, are listed.

### File Numbers of Existing DDMs for a Database

This function displays for a given DBID a list of all file numbers for which DDMs have been defined.

You enter the desired DBID on the menu Show Defined DBIDs and Used FNRs when you invoke the function.

You can also invoke this function by entering a DBID in the command line of the screen Database IDs Defined in Natural and pressing PF5.

## Copy DDM to Another FDIC File

This function is used to copy DDMs from one system file (FDIC) and/or database to another. This may be necessary, for example, when a Natural application is transferred from test to production status.

This function uses the function for copying DDMs of the SYSMAN utility (see also DDMs, as described in the section SYSMAN Utility in the Natural Utilities documentation).

## Generate DDM from Adabas FDT

This function is used to generate a DDM from an existing Adabas Field Description Table (FDT).

You have to enter the file number (FNR) of the Adabas file.

You can also enter a DBID. If you do not enter one, the DBID currently in effect for the session is used.

The generated DDM is placed in the SYSDDM work area for further processing.

# SYSEDT Utility - Editor Buffer Pool Services

The Editor Buffer Pool Services utility SYSEDT is intended for Natural administrators and used to:

- Display parameters and runtime information of the editor buffer pool,
- Modify parameters,
- Delete logical work and recovery files.

The SYSEDT utility documentation covers the following topics:

- Defining a Natural Security Library Profile
  - Invoking and Operating SYSEDT
  - General Information
  - Generation Parameters
  - Users
  - Logical Files
  - Recovery Files
  - System Administration Facilities
- 

## Defining a Natural Security Library Profile

If you have Natural Security installed, you must create a library security profile for the SYSEDT utility.

For details, see Library Maintenance as described in the Natural Security documentation.

## Invoking and Operating SYSEDT

Below is information on:

- Invoking the SYSEDT Utility
- Invoking a SYSEDT Function
- Using Direct Commands Help

### Invoking the SYSEDT Utility

 **To invoke the SYSEDT utility**

- In the direct command line, enter SYSEDT.

The SYSEDT Main Menu is displayed with the following functions:

- General Information
- Generation Parameters
- Users
- Logical Files
- Recovery Files
- System Administration Facilities

The SYSEDT functions are explained in the remainder of this section.

## Invoking a SYSEDT Function

### To invoke a SYSEDT function

- On the SYSEDT Main Menu, enter the corresponding function code.  
Or, on the SYSEDT Main Menu, press the appropriate PF key.  
Or, skip the SYSEDT Main Menu and access the desired function direct by entering the direct command SYSEDT followed by the corresponding function code in the NEXT line.

The individual functions are described in the following section.

## Using Direct Commands Help

If you enter a question mark (?) in the command line, all direct commands available within the SYSEDT utility are displayed in alphabetical order.

## General Information

### To invoke the General Information function

- On the SYSEDT Main Menu, enter Function Code **G**.  
Or, on the SYSEDT Main Menu, press PF10 (GInfo).

The General Information screen is displayed which provides an overview of the current status of the editor buffer pool:

Item	Shows
Usage Statistics	The currently available total number, the currently used number, and the currently used percentage of the available number of the items that follow.
Buffer Pool Blocks	The number of blocks in the editor buffer pool.
Work File Records	The number of records in the editor work file.
Control	The number of control records, which is always one.
Work	The number of work records.
Recovery Records	The number of recovery records.
Logical Files	The number of logical files.
Requests	The total number of read and write requests, the number of read and write requests for buffer pool blocks (Pool column), and the number of read and write requests for work or recovery files (File column). The Copy column shows read requests, which (in contrast to locked read requests) result in the deletion of the corresponding buffer pool block.
Read Work	The number of read requests for logical file records. A logical file record can be found in the buffer pool (Pool column) or on the work file (File column). It can be read by a locked or by a copy request: locked means that the record is kept in the buffer pool for some time; copy means that it is deleted from the buffer pool after having been read.
Write Work	The number of write requests for logical file records. A record can be either written to the buffer pool (Pool) or moved to the work file (File) if there are no free blocks available.
Read Recovery	The number of read requests for recovery records in the editor work file.
Write Recovery	The number of write requests for recovery records in the editor work file.
Timeout Values	Shows items with timeout values specified in seconds. These timeout values can be modified after pressing PF5 (Updat) and dynamically set after pressing PF5 (Save) again. The modified values are not kept during a restart of the buffer pool. The values from the work file control record are used instead.
Logical Files	The time after which a logical file is deleted if it has not been accessed during this time.
Files Delete Check	The time after which all logical files are checked periodically whether they can be deleted.
Changed Blocks	The time after which blocks that have been modified can be freed by writing them to the work file.
Unchanged Blocks	The time after which blocks that have not been modified can be freed by writing them to the work file.
Locked Blocks	The time after which blocks that have been read with locked can be freed by writing them to the work file.

## Generation Parameters

### To invoke the Generation Parameters function

- On the SYSEDT Main Menu, enter Function Code **P**.  
Or, on the SYSEDT Main Menu, press PF11 (Parms).

The Generation Parameters screen appears.

Below is a description of the individual parameters in alphabetical order which are displayed on the Generation Parameters screen:

Parameter	Explanation
CTOUT	Timeout value (in seconds) for changed buffer pool blocks.
DDNAME	Name of the editor work file for the JCL definition.
DSNAME	Name of the work file dataset.
DTOUT	Period of time (in seconds) for logical files to be checked for deletion.
FMODE	Mode of the work file name, which can be from A1 to Z9. This parameter applies under CMS only; it is not displayed in any other environment.
FTOUT	Timeout value (in seconds) for a logical file, which has not been accessed, to be deleted.
IMSG	Initialization message to be issued on the operator console.
ITOUT	Timeout value (in seconds) for the buffer pool initialization.
LRECL	Work file record length.
LTOUT	Timeout value (seconds) for locked buffer pool blocks.
MAXLF	Maximum number of logical files in the editor buffer pool.
PWORK	Percentage of work file records used as work records.
RECNUM	Total number of work file records.
RWORK	Percentage of work records for regular logical files.
UTOUT	Timeout value (in seconds) for unchanged buffer pool blocks.

For further information on these parameters, refer to the section Operating The Software AG Editor (in the Natural Operations for Mainframes documentation).

The Start column refers to the buffer pool restart. The following start values can be displayed for the above parameters:

Value	Explanation
L	The value for the corresponding parameter is taken from either the editor parameter module or the work file definition.
C	A modification of the corresponding parameter value forces a buffer pool cold start. Recovery records are lost.
W	A modification of the corresponding parameter value results in a buffer pool warm start. Recovery records are kept.

## Modifying Parameter Values

### To modify parameter values

1. Press PF5 (Updat).
2. Press PF5 again to save the new parameter values in the editor work file control record.

They will be activated when the buffer pool is started again.

## Users

### To invoke the Users function

- On the SYSEDT Main Menu, enter Function Code **U**.  
Or, on the SYSEDT Main Menu, press PF7 (Users).

The Users screen is displayed which provides the following information:

Item	Shows
User ID	The Natural user ID.
Logical Files	The number of logical files defined per user.
Pool Blocks	The number of buffer pool blocks per user.
Work Records	The number of work records per user.
Recovery Files	The number of recovery files per user.
Recovery Records	The number of recovery records per user.

The first column contains an input field labeled **M** for Mark. You may enter a question mark (?) in this field to invoke a window which shows you the codes that are valid input for this field.

To perform the function described, you enter the code in the Mark field.

Code	Function
/	Position line to top of screen.
P	
F	Select the logical files of this user.
R	Select the recovery files of this user.
D	Delete all logical files and/or recovery files for this user.

## Logical Files

### To invoke the Logical Files function

- On the SYSEDT Main Menu, enter Function Code **F**.  
Or, on the SYSEDT Main Menu, press PF6 (Files).

The Logical Files screen is displayed which provides the following information:

Item	Shows
File No.	The logical file number.
User ID	The Natural user ID.
Type	The logical file type.
Pool Blks	The number of buffer pool blocks currently used per logical file.
File Recs	The number of work file records currently allocated per logical file.
Last Access	The date and time of the last read or write request per logical file.

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 the function described.

Code	Function
/	Position line to top of screen.
P	
S	Select the logical files of this user.
D	Delete the logical file.

## Recovery Files

### To invoke the Recovery Files function

- On the SYSEDT Main Menu, enter Function Code **R**.  
Or, on the SYSEDT Main Menu, press PF8 (Recov).

The Recovery Files screen is displayed which provides the following information:

Item	Shows
User ID	The Natural user ID.
Member	The library member name.
Library	The library name.
Type	The library type.
Recs	The number of recovery records per recovery file.
Creation Date/Time	The creation date and time of the recovery file.

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 the function described.

Code	Function
/	Position line to top of screen.
P	
S	Select the recovery files of this user.
D	Delete the recovery file.

## System Administration Facilities

### To invoke the Administration Facilities screen

- On the SYSEDT Main Menu, enter Function Code **A**.  
Or, on the SYSEDT Main Menu, press PF5 (Admin).

The Administration Facilities screen is displayed which offers you the function of either terminating the editor buffer pool or leaving the SYSEDT utility.

If you choose to terminate the buffer pool (Function Code **T**), a window appears prompting you for confirmation.

If you enter Yes for confirmation, a further window is displayed, asking you whether or not you want the editor buffer pool to be immediately restarted.

If you enter Yes again, the buffer pool is immediately restarted, which gives you the possibility to immediately activate modified generation parameters.

If you specify No, you leave SYSEDT and can perform actions outside your TP environment, for example, change the size of your editor work file; see also Editor Work File and Editor Buffer Pool in the Natural Operations for Mainframes documentation.

# SYSERR Utility - Overview

When you develop a Natural application, you may want to separate error or information messages from your Natural code and manage them separately. This makes it easy for you, for example, to standardize messages, to have predefined message ranges for different kinds of message, to translate messages into other languages or to attach to a message a long text that explains it in more detail.

With the SYSERR utility, you can write your own application-specific messages. In addition, you can customize the texts of the existing Natural system messages.

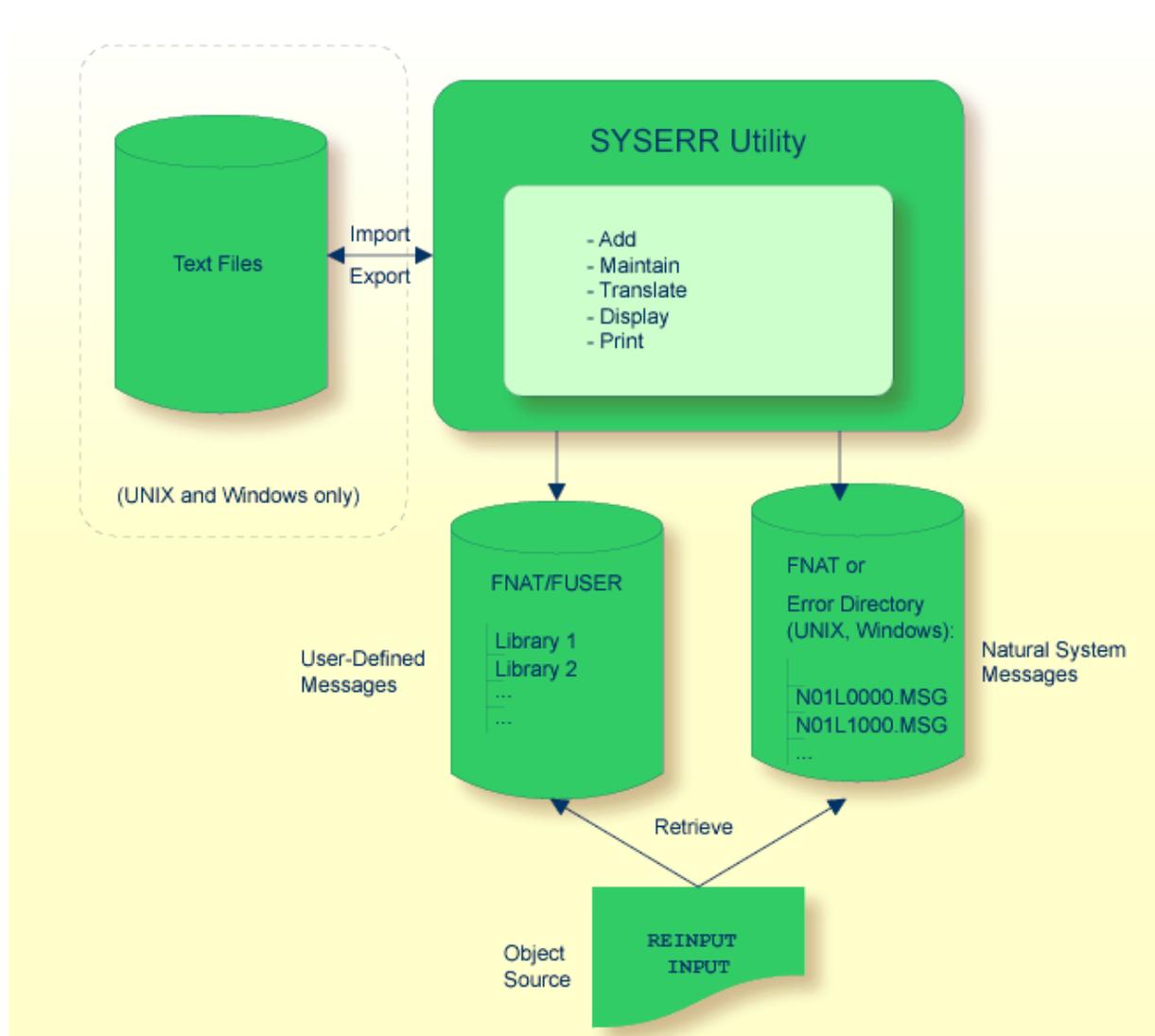
The SYSERR utility documentation provides the following topics:

•	General Information on Messages	Natural message types and languages and how they are used.
•	Invoking SYSERR	Starting the SYSERR utility.
•	Parameters	Parameters provided with SYSERR functions, such as message types.
•	Functions	SYSERR functions available for creating and maintaining messages.
•	Direct Commands	Direct commands for specifying settings and executing SYSERR functions.
•	Upper Case - ERRUPPER	Converting messages from lower to upper case.
•	Replacing Characters - ERRCHAR	Replacing characters in message texts.
•	Unloading Messages - ERRULDUS	Unloading messages from Natural system files into work files.
•	Loading Messages - ERRLODUS	Loading messages from work files into Natural system files.
•	Appl. Programming Interface USR0020P	Reading messages from Natural system files.

# SYSERR - General Information on Messages

This section contains information on the types of message and message languages that can be managed with the SYSERR utility and how messages are issued and retrieved in your Natural system environment.

The following graphic demonstrates the functionality of the SYSERR utility and how messages are processed within Natural:



The section covers the following topics:

- Message Types
- Message Languages
- Issuing Messages
- Retrieving Natural System Short Messages
- Retrieving User-defined Short Messages
- Obtaining Message Information

---

## Message Types

There are two types of message: Natural (system) messages and user-defined messages:

Natural system messages are issued by the Natural nucleus and Natural utilities. Natural system messages are delivered by Software AG and stored as message files in the Natural system file FNAT. Natural system messages begin with NAT, followed by a four-digit number, for example, NAT0230.

User-defined messages are issued by applications written by a user. User-defined messages are stored as message files in libraries (including SYS-libraries) in the system file FUSER or FNAT.

A message can be translated into different languages. Each language is stored in a separate message file. A maximum of 9999 messages can be stored per library and message file.

There are four types of message text:

- Natural system short message
- Natural system long message
- User-defined short message
- User-defined long message

A short message is the one-line message which is displayed in the message line when the corresponding error situation occurs.

A long message is a detailed explanation of the corresponding short message and includes instructions for solving problems.

### **Attention:**

Keep in mind that any modifications of Natural system messages can result in wrong messages or a loss of modifications when a new Natural version is released.

## Message Languages

Messages can be created in up to 60 languages as described for the system variable \*LANGUAGE in the Natural System Variables documentation.

The following rules and restrictions apply:

- Natural system short messages must be entered in English first, and can then be translated into any other language.
- Natural system long messages can be entered in English, but cannot be translated into other languages.
- User-defined short messages can be entered in any language, and then translated into any other language.
- User-defined long messages can be entered in any language, but only if the corresponding short message in the same language already exists.

## Issuing Messages

The section below contains information on the statements that are used for a Natural system message or a user-defined message in a Natural program.

- ▶ **To issue a Natural system short message in a program**

- Enter the Natural statement

```
INPUT WITH TEXT *-nnnn
```

or

```
REINPUT WITH TEXT *-nnnn
```

where *nnnn* is the number of the requested message.

### ▶ To issue a user-defined short message in a program

- Enter the Natural statement

```
INPUT WITH TEXT *nnnn
```

or

```
REINPUT WITH TEXT *nnnn
```

where *nnnn* is the number of the requested message.

### Example Program:

```
DEFINE DATA LOCAL
01 #XYZ (A10)
END-DEFINE
*
* USER-DEFINED MESSAGE NUMBER 1 IN LIBRARY TESTLIB:
* 'INPUT MISSING.'
*
INPUT #XYZ
IF #XYZ = ' ' THEN
REINPUT WITH TEXT *0001
END-IF
WRITE #XYZ
END
```

For further details on the INPUT and REINPUT statements, refer to the relevant sections in the Natural Statements documentation.

## Retrieving Natural System Short Messages

When a program references a Natural system short message, Natural looks for the requested message number in the Natural system file FNAT in the following order:

1. under the current language code as determined by the system variable \*LANGUAGE,
2. under Language Code 1 (English).

If neither of the above is found, a program references a message that does not exist and you only receive the message number prefixed with NAT, for example, NAT0230.

## Retrieving User-defined Short Messages

When a program references a user-defined short message, Natural first looks for the requested message number *nnnn* under the current language code as determined by the system variable \*LANGUAGE (see the Natural System Variables documentation). If that message does not exist, Natural looks for the requested message number *nnnn* under Language Code 1 (English). If that message does not exist either, Natural looks for message

number  $n000$  (where  $n$  is the first digit of the requested message number) under Language Code 1.

These three search steps are first performed in the current library. If nothing is found there, further libraries are searched in the same way until a corresponding message is found.

The sequence of libraries for the search is as follows:

1. The current library as determined by the system variable \*LIBRARY-ID,
2. The steplib; if Natural Security is installed, the sequence in which the steplibs are specified in the Natural Security profile of the current library,
3. The default steplib as determined by the system variable \*STEPLIB,
4. The library SYSTEM in the system file FUSER (\*),
5. The library SYSTEM in the system file FNAT (\*).

(\*) If the name of the current library begins with SYS, SYSTEM FNAT is searched before SYSTEM FUSER.

## Obtaining Message Information

When you receive a short message, you may be looking for additional information on the problem situation.

- With the system command HELP, you can display Natural system long messages or user-defined long messages.
- With the system command LASTMSG, you can list the short text of the message(s) that occurred last and additional information on the error situation. The information displayed includes associated error messages that possibly preceded the last message.

Both commands are described in the Natural System Command Reference documentation.

# Invoking SYSERR

## ▶ To invoke the SYSERR utility

- Enter the system command SYSERR.

The main menu of the SYSERR utility is displayed:

```

11:18:52                ***** NATURAL SYSERR UTILITY *****                2000-07-04
                        - Menu -

Code  Function
-----
AD   Add new messages
DE   Delete messages
DI   Display messages
MO   Modify messages
PR   Print messages
SC   Scan in messages
SE   Select messages from a list
TR   Translate messages into another language
?   Help
.   Exit
-----

Code .. ___  Message type .... US
                Library ..... SYSTEM__
                Message number .. 1___ - 9999
                Language codes .. 1_____

Please enter code.
Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                Help           Exit                               Canc

```

From the main menu of the SYSERR utility, you can execute all functions available for adding and maintaining messages. The individual functions are explained in the section Functions. The parameters that apply with the functions are explained in general in the section Parameters; any restrictions that apply to the use of these parameters are described for each function concerned in the section Functions.

The SYSERR utility provides an extensive online help system. To obtain field-specific help information, either enter a question mark in the relevant field and press ENTER or place the cursor in the field and press PF1.

# SYSERR - Parameters

To invoke a SYSERR utility function, on the SYSERR main menu, enter the code for the function desired and one or more of the parameters described below. Any restrictions that apply to the use of parameters with a particular function are described in the section Functions.

The section below contains information on:

- Message Type
  - Library
  - Message Number
  - Language Codes
- 

## Message Type

Specifies the type of message to be processed. The table below lists the message types available:

Type	Explanation
NS	Natural system short messages
NL	Natural system long messages
US	User-defined short messages
UL	User-defined long messages

## Library

Specifies the library for which messages are to be created or maintained. The specification of a library is not required when accessing Natural system messages (Message types NS and NL) and any input values in the field Library will be ignored.

## Message Number

Specifies the first and last number of a message range. The maximum message number for a library and language is 9999. The message number 0000 is not allowed. To specify only one message number, either enter the number of the message in the left Message Number field and clear the right field, or enter the number in both fields.

## Language Codes

Specifies a maximum of 9 from 60 language codes available. The language codes are single alphanumeric characters in the ranges 1 - 9, A - Z and a - y. To view or select language codes, enter a question mark (?) in the first position of the Language Codes field and press ENTER. For more information, see the system variable \*LANGUAGE in the Natural System Variables documentation.

# SYSERR - Functions

To invoke a SYSERR utility function, on the SYSERR main menu, enter the corresponding function code and one or more parameters described in this section. For general instructions on the use of parameters, refer to the section Parameters.

The following functions are provided:

- Adding Messages
  - Deleting Messages
  - Displaying Messages
  - Modifying Messages
  - Printing Messages
  - Scanning Messages
  - Selecting Messages from a List
  - Translating Messages into other Languages
  - Copying Messages to a System File - SYSMAIN Utility
-

# Adding Messages

**▶ To add new messages**

1. In the fields on the SYSErr main menu, enter the following values:

Field	Input Value
Code	AD
Message Type	NS Natural system short messages NL Natural system long messages US User-defined short messages UL User-defined long messages  A long message can only be added if the corresponding short message already exists, as the long message is intended to be an explanation of the short message.
Library	Any existing Natural library.
Message Number	Two numbers of up to four digits corresponding to the first and last numbers of the range of messages to be added. If you only want to add one message, either enter the number of the new message in the left Message Number field and clear the right field, or enter the number in both fields.
Language Codes	The code of the language for which the message is to be added. If the message type is NS or NL, the language code must be 1 for English. For other message types, the first language code entered in the field is used; all others are ignored.

2. Press ENTER.  
The Add Short Message screen is displayed:

```

11:20:23          ***** NATURAL SYSErr UTILITY *****          2003-09-16
                    - Add Short Message -

Number           Short Message
-----
SYSErr1004
                .....1.....2.....3.....4.....5.....+..

Sample ..... Message sample number 0000
    
```

3. In the input line next to the message number, type in a short message text.  
If the text you type in contains the string **0000**, the string **0000** is replaced by the message number when saving the message. The use of a sample message (as indicated in the example above) is explained in Copying a Sample Message below.
4. Press ENTER to save the new short message.

5. Press PF9/Long to add a corresponding long message text.  
The Add Long Message screen is displayed:

```

11:21:59          - Add Long Message SYSERR1004 Language 1 -          2003-09-16
 1 Tx. Message sample number 1004
 2      .
 3      .
 4 Ex. .
 5      .
 6      .
 7      .
 8      .
 9      .
10     .
11     .
12     .
13     .
14     .
15     .
16     .
17     .
18 Ac. .
19     .
20     .

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

6. Enter text in the three input areas: Tx. (text), Ex. (explanation) and Ac. (action).
7. Press ENTER to save the long message.
8. Press PF9/Short to return to the short message or to add the next short message in ascending order if you have selected a range of message numbers.
9. Press PF3 or PF12 to return to the SYSERR main menu.  
Or press PF8 or PF7 to add the next short message in ascending or descending order if you have selected a range of message numbers.

## Copying a Sample Message

Below the message input line, a line is displayed labeled Sample. The text contained in this line can be copied to the message input line by entering .C in the empty message input line.

If the sample message contains the string 0000, this string 0000 is replaced by the message number when the sample is copied as illustrated in the example below.

```

11:21:13          ***** NATURAL SYSERR UTILITY *****          2003-09-16
                   - Add Short Message -

Number           Short Message
-----
SYSERR1004      Message sample number 1004
                .....1.....+.....2.....+.....3.....+.....4.....+.....5.....+..

Sample ..... Message sample number 0000
    
```

For information on how to create a sample message, see the SAMPLE command described in the section SYSERR Direct Commands.

## Deleting Messages

 **To delete messages**

- In the fields on the SYSERR main menu, enter the following values:

Field	Input Value
Code	DE
Message Type	NS Natural system short messages NL Natural system long messages US User-defined short messages UL User-defined long messages  It is possible to delete a long message without deleting the corresponding short message, but not vice versa. If you try to delete a short message for which a long message exists, you are asked to confirm the deletion of both.
Library	Any existing Natural library.
Message Number	Two numbers of up to four digits corresponding to the first and last numbers of the range of messages to be deleted.
Language Codes	The code(s) of the language(s) in which the messages are to be deleted. To indicate that the messages specified are to be deleted in all languages available, enter an asterisk (*).

# Displaying Messages

 **To display messages**

1. In the fields on the SYSERR main menu, enter the following values:

Field	Input Value
Code	DI
Message Type	NS Natural system short messages NL Natural system long messages US User-defined short messages UL User-defined long messages
Library	Any existing Natural library.
Message Number	Two numbers of up to four digits corresponding to the first and last numbers of the range of messages to be displayed.
Language Codes	The code of the language in which the messages are to be displayed. Only one language code is accepted. If more than one code is specified, only the first one is used; all others are ignored.

2. Press ENTER.

For short messages, the Display Short Messages screen is displayed:

```

15:41:11          ***** NATURAL SYSERR UTILITY *****          2003-09-17
                  - Display Short Messages -

Number           Short Message (English)
-----
NAT0001         Missing/invalid syntax; undefined variable name/keyword.
NAT0002         No file is available with specified name or number.
NAT0003         Invalid character string for file name or file number.
NAT0004         DEFINE DATA must be the first statement if present.
NAT0005         Closing parenthesis missing in arithm/logical expression.
NAT0006         ESCAPE statement used when no processing loop active.
NAT0007         Invalid THRU or TO clause in READ LOGICAL or HISTOGRAM.
    
```

Press PF8 to page forwards.

For long messages, the Display Long Message screen is displayed where the messages are displayed one after another by pressing PF8 to page forwards or PF7 to page backwards.

The Display Long Messages screen is similar to the Modify Long Message screen as shown in Modifying Messages below.

# Modifying Messages

**▶ To modify messages**

1. In the fields on the SYSERR main menu, enter the following values:

Field	Input Value
Code	MO
Message Type	NS Natural system short messages NL Natural system long messages US User-defined short messages UL User-defined long messages
Library	Any existing Natural library.
Message Number	Two numbers of up to four digits corresponding to the first and last numbers of the range of messages to be modified.
Language Codes	The code of the language in which the messages are to be modified. Only one language code is accepted. If more than one code is specified, only the first one is used; all others are ignored.

2. Press ENTER.

The Modify Short Message screen is displayed:

```

18:52:33          ***** NATURAL SYSERR UTILITY *****          2003-09-16
                   - Modify Short Message -

Number           Short Message (English)
-----
SYSERR1004      Message sample number 1004
                .....1.....2.....3.....4.....5.....

1 Tx. Input missing.
2      .
3      .
4 Ex. Input value missing in field XYZ.
5      Enter an alphanumeric value.
6      .
7      .
8      .
18 Ac. Enter value in field XYZ.
19     .
20     .

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

For reference purposes, the long message is displayed in the bottom half of the screen.

When you modify long messages, the Modify Long Message screen is displayed:

```

18:54:02      - Modify Long Message SYSERR1004 (English) -                2003-09-16
1 Tx. Input missing.
2      .
3      .
4 Ex. Input value missing in field XYZ.
5      Enter an alphanumeric value.
6      .
7      .
8      .
9      .
10     .
11     .
12     .
13     .
14     .
15     .
16     .
17     .
18 Ac. Enter value in field XYZ.
19     .
20     .

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Mod          Exit          -      +          Copy          Canc

```

3. Press ENTER to save modifications.
4. Press PF8 or PF7 to modify the next message in ascending or descending order if you have selected a range of numbers.

# Printing Messages

**▶ To print messages**

1. In the fields on the SYSEERR main menu, enter the following values:

Field	Input Value
Code	PR
Message Type	NS Natural system short messages NL Natural system long messages US User-defined short messages UL User-defined long messages
Library	Any existing Natural library.
Message Number	Two numbers of up to four digits corresponding to the first and last numbers of the range of messages to be printed.
Language Codes	The code of the language in which the messages are to be printed. Only one language code is accepted. If more than one code is specified, only the first one is used; all others are ignored.

2. Press ENTER.  
The print window is displayed:

```

+-----Print Natural System Messages-----+
!                                     !
!   Language code .... 1             !
!                                     !
!   Long texts, too .. N             !
!   Message number ... 1___ - 25     !
!   Lines per page ... 60_           !
!   Left margin ..... 10            !
!   Top margin ..... 0_             !
!   Bottom margin .... 0_           !
!   Printer ID ..... PRT1_____    !
!                                     !
!                                     !
+-----+
    
```

3. Specify the options provided in the print window and the logical printer name.  
See the DEFINE PRINTER statement in the Natural Statements documentation for details on logical printer names.
4. Press ENTER to output the selected messages on a printer.

**▶ To print all Natural system messages**

- In the fields on the SYSEERR main menu, enter the following values:

Function Code PR,  
 Message Type NS or NL,  
 Message Numbers 1 - 9999,  
 and Language Code 1 (English) or 2 (German).

A Library ID is not required and possible entries are ignored.

## Scanning Messages

This function is used to scan messages for a specific string of characters. Only short messages can be scanned.

### To scan messages

1. In the fields on the SYSERR main menu, enter the following values:

Field	Input Value
Code	SC
Message Type	NS Natural system short messages US User-defined short messages
Library	Any existing Natural library.
Message Number	Two numbers of up to four digits corresponding to the first and last numbers of the range of messages to be scanned.
Language Codes	Specify a maximum of nine language codes from the ranges 1 - 9, A - Z and a - y, or specify an asterisk (*).

2. Press ENTER.  
The scan window is displayed (see the Example below).
3. Specify the following:

Scan Value(s)	In the four empty fields, enter up to four terms to be searched for. The scan finds the specified terms in both upper and lower case.
OR/AND/NOT	You can vary the conditional operators OR, AND or NOT between the search terms. In the Example below, the search would be for all short messages that contain both the words "buffer" and "pool". If the operator NOT were specified, the search would find all messages that contain neither the word "buffer" nor the word "pool".
Absolute	If you mark this field, the string of characters is found even if it is part of a larger string in the message text. For example, if you scan for the value "meter", the search would also find words, such as "parameter" and "millimeter".
Immediate	If you mark this field, messages are displayed individually, one after another. Otherwise, a list of messages is displayed after the search is completed. If you specify more than one language or an asterisk (*) in the Language Codes field, Immediate must be marked.

**Example:**

```

+-----+
!  Scan value(s)      Or/And/Not  !
!  -----          -----      !
!  BUFFER_____      AND          !
!  POOL_____        !
!  _____        !
!  _____        !
!  Absolute ..... X  !
!  Immediate ..... _ !
!                                     !
+-----+
    
```

4. Press ENTER.

All messages to which the specified scan criteria apply are displayed. The word in which the search string is found is displayed in intensified form. The example output of a scan is shown below:

```

11:32:27          ***** NATURAL SYSERR UTILITY *****          2000-07-04
                   - Scan in Short Messages -

Number           Short Message (English)
-----
NAT0777         Buffer pool full.
    
```

From this screen, you can display the search criteria for the current scan by pressing PF10.

## Selecting Messages from a List

This function is used to display a range of messages and select single ones for further processing. Only short messages can be displayed.

 **To select messages**

1. In the fields on the SYSERR main menu, enter the following values:

Field	Input Value
Code	SE
Message Type	NS Natural system short messages US User-defined short messages
Library	Any existing Natural library.  If an asterisk (*) is appended to the library ID, a list of all libraries available is displayed for selection.
Message Number	Two numbers of up to four digits corresponding to the first and last numbers of the range of messages to be displayed for selection.
Language Codes	The code of the language in which the messages are to be displayed.  If more than one code is specified, only the short message text of the first one is displayed. Enter an asterisk (*) to display the languages available for each message.

2. Press ENTER.  
The Select Messages screen is displayed:

```

11:33:21          ***** NATURAL SYSERR UTILITY *****          2000-07-04
                    - Select Messages -

Se Number          Short Message (English)          Languages
                    -----
                    short          long
-----
__ NAT0001          Missing/invalid syntax; undefined variable name/ 1          1
__ NAT0002          No file is available with specified name or numb 1          1
__ NAT0003          Invalid character string for file name or file n 1          1
__ NAT0004          DEFINE DATA must be the first statement if prese 1          1
__ NAT0005          Closing parenthesis missing in arithm/logical ex 1          1
__ NAT0006          ESCAPE statement used when no processing loop ac 1          1
__ NAT0007          Invalid THRU clause in READ LOGICAL/HISTOGRAM st 1          1
    
```

3. In the column **Se**, enter any of the following line commands:

Command	Function
DE	Delete the message.
DI	Convert the Select Messages screen into the Display Short Messages screen as shown in Displaying Messages above. Additionally, place the message selected with this command at the top of the list and reduce the number of messages displayed as described for <b>.X</b> below.
LA	Show into which languages the message has been translated.
MO	Modify the message.
PR	Output the message on a printer.
SH	Display the short message. This command is only available if an asterisk (*) has been entered in the Language Code field on the SYSERR main menu.
TR	Translate the message into another language.
.X	Define a shorter message range by placing a selected message at the top of the list and thus reducing the number of messages displayed:  The message selected with this command is placed at the top of the list and any messages that were listed above this message are removed from the display. The message range on the SYSERR main menu is reset accordingly and starts with the message selected here on the Select Messages screen.
.Y	Define a shorter message range by listing messages only up to a selected message:  All messages that were listed below the message selected with this command are removed from the display. The message range on the SYSERR main menu is reset accordingly and ends with the message selected here on the Select Messages screen.

4. Press ENTER to continue.

## Translating Messages into other Languages

This function is used to translate short messages from one language to one or more other languages. To translate long messages into other languages, use the Add Long Messages function described in Adding Messages.

 **To translate messages**

1. In the fields on the SYSERR main menu, enter the following values:

Field	Input Value
Code	TR
Message Type	NS Natural system short messages US User-defined short messages
Library	Any existing Natural library.
Message Number	Two numbers of up to four digits corresponding to the first and last numbers of the range of messages to be displayed for selection.
Language Codes	Specify a maximum of nine language codes. The language codes are single alphanumeric characters in the ranges 1 - 9, A - Z and a - y.

2. Press ENTER.

The Translate Short Message screen is displayed:

```

11:42:21          ***** NATURAL SYSERR UTILITY *****          2003-07-04
                   - Translate Short Message -

Number ..... SYSERR0002
Languages ... 123...7.9ABCD.....

-----+...+...1...+...2...+...3...+...4...+...5...+..
English      Short message number 2 _____
German       Short message number 2 (German) _____
French       Short message number 2 (French) _____
Spanish      _____
Italian      _____
Dutch        _____
Turkish      Short message number 2 (Turkish) _____
Danish       _____
Norwegian    Short message number 2 (Norwegian) _____
-----+...+...1...+...2...+...3...+...4...+...5...+..

  1 Short message number 2
  4 Line 4 of message number 2 (long text, English)
 18 Line 18 of message number 2 (long text, English)

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

The fields Number and Language display the message number and the codes of the languages in which the message already exists (in the example screen above: 1, 2, 3, 7, 9, A, B, C and D).

The section below Number and Language lists the languages and their corresponding translations or displays empty input lines for entering translations in the new languages that were specified earlier in the Language Code field of the SYSERR main menu when the function was invoked (in the example above: 1, 2, 3, 4, 5, 6, 7, 8 and 9 where 4, 5, 6 and 8 are new languages).

For reference purposes, the bottom section of the screen displays three lines of the long message that corresponds to language that is listed first in the languages/short messages section (in the example above, English). Lines 1, 4 and 18 are displayed by default. You can display any other line of the long message by overtyping any of the three line numbers (1, 4 or 18 ) with another line number and pressing ENTER.

3. Enter the translation in the input line next to the new language specified.
4. Press ENTER.

**▶ To modify translations of short messages**

1. On the Translate Short Message screen, press PF10.  
The Options window is displayed:

```

+----- Options -----+
!                               !
! Modification of all fields allowed ..... N      !
!                               !
! Currently recognized language codes ..... 123456789 !
!                               !
+-----+
    
```

2. In the field Modification, enter **Y** to override the default.

In this window, you can also specify up to nine new language codes for translation.

**▶ To copy a translation into an empty input line**

1. On the Translate Short Message screen, in an empty input line, over the first two positions type in **.C**.
2. Place the cursor anywhere in the line of a short message that already exists for another language.  
(You can only copy text that appears in display mode.)
3. Press ENTER.

## Copying Messages to a System File - SYSMAIN Utility

A message or a range of messages can be copied from one library to another or from one system file to another.

**▶ To copy messages from one library or file to another**

- Use the Copy Messages function of the SYSMAIN utility.  
Or, use the unload facility ERRULDUS and the load facility ERRLODUS.

# SYSERR - Direct Commands

From the SYSERR main menu, you can execute the following commands by entering them in the Command line:

Command	Function
LAYOUT	Specifies valid message ranges to categorize messages. Overlapping of ranges is possible. A new message can only be added if its number is within the range specified in the layout.
NEXT	Searches for the next free message number within the message number range specified. Free means that this message number is available and has not yet been assigned to a message file in any language.
NEXTTAB	Same as NEXT, but returns a list of numbers from which you can select a certain number.
RESTART	Re-initializes SYSERR (and its default values) without leaving the utility.
SAMPLE	<p>Invokes the "Edit SAMPLE message" window where you create or modify a sample message to be used as a master for creating new short messages.</p> <p>To create or modify a sample message, proceed as follows:</p> <p>In the editor area of the "Edit SAMPLE message" window, type in the message text desired or modify existing text. If you enter the string <b>0000</b> (combined with text or not), the string <b>0000</b> is replaced by the number of the new message when copying the message. See also Copying a Sample Message in the section Functions.</p> <p>In the field "Read or Write sample", enter a <b>W</b> to save your entries.</p> <p>In the field Library, enter the name of the library for which the sample message is to be used. If you leave the field Library blank, the sample applies to Natural system messages.</p> <p>Press PF3 to exit the "Edit SAMPLE message" window.</p> <p>You can define one sample message for each language and library.</p>
SECURITY	Invokes a security window where you can enter a password and cipher code for accessing security-protected Adabas and VSAM files.
SHIFT	If activated, automatically shifts the text of a short message to the left margin when confirming a modification or adding a new message.
TRACE	Counts the number of database accesses. When the message number specified has been reached, a window is displayed. The default number is 900. If set to 0, the trace facility is shut off. The commands TRACE ON and TRACE OFF can be entered directly in the command line. TRACE ON sets the access counter to 900; TRACE OFF sets the access counter to 0.
USEREXIT	Invokes the program USEREXIT in the Natural system library SYSERR.

# **SYSERR - Upper Case - ERRUPPER**

Natural system messages are provided in lower case. If your terminals cannot display lower-case characters correctly, convert the messages from lower to upper case by executing the program ERRUPPER in the Natural system library SYSERR. Once the messages have been converted to upper case, however, you cannot convert them back to lower case. To recover lower-case messages, you have two options:

- Reload the messages by using ERRLODUS or the Natural Object Handler.
- Unload the lower-case messages to a free language code by using ERRULDUS or the Natural Object Handler before conversion so that a backup always exists.

See also the Natural Object Handler documentation.

# Replacing Characters - ERRCHAR

If your terminal does not display certain characters correctly, it is possible to search for these characters and replace them by new characters of your choice. This is done by executing the program ERRCHAR in the Natural system library SYSERR. However, it is only possible to replace characters in Natural system short messages. Using ERRCHAR, you scan for a specific character and replace the hexadecimal code that represents this character with another hexadecimal code.

After executing the program ERRCHAR, the ERRCHAR menu is displayed with the following functions:

- Scan for a given character
- Scan and Replace characters
- Display one message in hexadecimal format
- EBCDIC character table for your terminal
- Translate using character set ERRCSET

The following input fields are provided on the ERRCHAR menu:

Field	Description
Error Number	The range of messages to be included in the search or search/replace operation.
Language Code	The language code of Natural system short messages to be included in the search or search/replace operation.
Scan Value	The hexadecimal value to be scanned for.
Replace Value	The hexadecimal value to replace all scan values found. Use the function "EBCDIC character table for your terminal" to determine which characters your terminal can represent.

# SYSERR - Unloading Messages - ERRULDUS

The program ERRULDUS is used to unload all message types supported by Natural; that is, user-defined long and short messages created with the SYSERR utility and Natural system messages. The messages are read from the FNAT or FUSER system file and written into Work File 2.

## ▶ To invoke ERRULDUS

- On the SYSERR main menu, in the command line, enter ERRULDUS.  
The ERRULDUS menu is displayed:

```

11:43:45          ***** NATURAL SYSERR Utility *****          2000-07-04
                  - ERRULDUS (Unload Texts to Work File 2) -

                  Code  Function
                  -----
                  US  User supplied short error texts
                  UL  User supplied long  error texts
                  U   User supplied short AND long texts
                  NS  NATURAL short error texts
                  NL  NATURAL long  error texts
                  H   NATURAL help texts
                  .   Exit
                  -----

Code ..... _
Source Library .. SYSERR_      Source Language Code .. 01
Target Library .. SYSERR_      Target Language Code .. 01
Error Number .... 1_ - 9999
Replace ..... N
Report ..... ON_

Please enter valid code.
Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Exec Help      Exit                                  Canc

```

The ERRULDUS menu contains the following input fields:

Field	Description
Code	The type of messages to be unloaded. Valid types are:  US User-defined short messages UL User-defined long messages U User-defined short and long messages NS Natural system short messages NL Natural system long messages H Natural help texts . Terminate processing
Source Library	The name of the library from which messages are to be unloaded. The name can be truncated by using an asterisk (*). If you enter an asterisk (*) only, messages from all libraries are unloaded. Source Library is ignored for messages of type NS or NL.
Source Language Code	The language code of the messages to be unloaded.
Target Library	The name of the library to which messages are to be loaded. Target Library is ignored for messages of type NS or NL.
Target Language Code	The language code to which messages are to be loaded.
Starting Error Number	The number of the first message to be unloaded. If no starting number is specified, it is assumed to be 1.
Ending Error Number	The number of the last message to be unloaded.
Replace	Enter <b>Y</b> to overwrite the target library. Default is <b>N</b> (no).
Report	If specified as <b>ON</b> , a list of all messages that were unloaded is displayed when unloading is complete. If specified as <b>OFF</b> , no such list is displayed.

ERRULDUS reads input until a period (.) is detected in the Code field. After unloading, a statistical listing is displayed.

# SYSERR - Loading Messages - ERRLODUS

The program ERRLODUS is used to load messages previously unloaded with the program ERRULDUS.

The messages are read from Work File 2 and are written to the FNAT or FUSER system file. ERRLODUS overwrites existing messages in the system file if Replace was specified as **Y** when unloading.

If you specify ERRLODUS as ON, a list of all messages that were loaded (added or replaced) is displayed when loading is complete.

**Note:**

Under Natural Security, the online use of ERRLODUS is only available for users of the type Administrator, regardless of the setting of the Utilities option in the security profile of the library. See also Utilities in Library Maintenance (General Options) in the Natural Security documentation.

# Appl. Programming Interface USR0020P

The application programming interface USR0020P in the Natural system library SYSEXT is provided to read messages from the FNAT or FUSER system file. Thus, it is possible, for example, to have long messages displayed in an application (as part of your own user-defined help system) without having to use the Natural system library SYSERR.

Log on to the Natural system library SYSEXT and, in the command line, enter the command MENU. In the list provided, mark the program USR0020P with a question mark (?). A window is then displayed, in which you can select the function to be executed for the program. If you enter an **I**, detailed information on the use of USR0020P is displayed.

# SYSMAIN Utility - Overview

The SYSMAIN utility, which is available online and in batch mode, is used to transfer objects within the Natural system from one environment to another.

The SYSMAIN utility documentation covers the following topics:

- General Information
- Invoking SYSMAIN
- Functions and Function Processing
- Direct Commands
- Parameters and Keywords
- Programming Objects
- Debug Environments
- Error Messages
- Profiles
- Rules
- DDMs
- DLI Subfiles
- Commands Issued to SYSMAIN
- PF Keys
- Data Rejected
- Special Considerations for Natural Administrators

# SYSMAIN General Information

This section covers the following topics regarding the SYSMAIN utility:

- Objects
- Environments
- Functions

## Objects

The following Natural objects can be transferred with SYSMAIN:

Object	Explanation
Programming Objects	Programs, subprograms, subroutines, classes, maps, data areas (local, parameter and global), copycodes, help routines, expert models, recordings, texts, reports, macros, processors and dialogs.
Debug Environments	User debug environments for online program testing.
Error Message Texts	Short and long texts of Natural system and user-supplied error messages.
Profiles	Editor profiles, map profiles, device profiles, and parameter profiles (created with the SYSPARM utility).
Rules	Automatic and free rules.
DL/I Subfiles	Natural NSBs, NDBs and UDFs.
DDMs	Data definition modules.

## Environments

The environment in which a Natural object is located depends on the object type.

The environment for each type of object is defined as follows:

Programming Object	Debug Environment	Error Message	Profile	Rule	DL/I Subfile	DDM
database						
FUSER and FNAT file	FUSER file	FUSER and FNAT file	FNAT file	FDIC file	FDIC file	FDIC file
file name (VSAM only)						
library	library	library				
		language				

## Functions

SYSMAIN provides the following functions:

Function	Explanation
COPY	Copy object from one environment to another environment.
DELETE	Delete object from a specific environment.
FIND	Locate a single object within a specific environment.
LIST	Display a range of objects within a specific environment.
MOVE	Transfer object from one environment to another.
RENAME	Give an object a new name, and (optionally) transfer it to a new environment.

Not all functions can be applied to all objects. The following table shows which functions are valid for each type of object:

Function	Programming Object	Debug Environment	Error Message	Profile	Rule	DDM	DL/I Subfile
COPY	X	X	X	X	X	X	X
DELETE	X	X	X	X	X	X	X
FIND	X		X				
LIST	X	X	X	X	X	X	X
MOVE	X	X	X	X	X	X	X
RENAME	X	X	X	X	X		

# Invoking SYSMAIN

- Invoking SYSMAIN Online
  - Terminating SYSMAIN Online
  - Invoking SYSMAIN in Batch Mode
  - Invoking SYSMAIN by an Appl. Programming Interface
- 

## Invoking SYSMAIN Online

There are two methods for invoking the SYSMAIN utility:

### To invoke SYSMAIN online from any Natural library

- Enter the command SYSMAIN.  
The current setting of the system variable \*LIBRARY-ID is passed to SYSMAIN and used as the default source library for processing of programming objects.  
If you issue a Natural system command from SYSMAIN, the command will apply to the library from which SYSMAIN has been invoked.

### To invoke SYSMAIN online from the Natural Main Menu

1. Select Maintenance and Transfer Utilities. A corresponding menu is displayed.
2. Select Transfer Objects to Other Libraries.

## Terminating SYSMAIN Online

The SYSMAIN utility is terminated with a period (.), with PF3 or with CLEAR from the SYSMAIN main menu; or with a period (.) in the command line of any other SYSMAIN menu.

**Do not** terminate the SYSMAIN utility with the terminal command %%, because the environment may not be reset correctly.

## Invoking SYSMAIN in Batch Mode

The SYSMAIN utility is invoked in batch mode in the same way as it is invoked online; however, one or more direct command strings must follow the SYSMAIN command. Direct commands for each SYSMAIN function as it applies to a specific object are included in sections discussing processing of specific Natural objects. If you want to execute other Natural commands after the SYSMAIN command(s), you must first terminate SYSMAIN by using the commands END or Quit.

There are two ways to enter direct commands:

- The direct command string follows the SYSMAIN command in the same input line; each parameter in the command string can be delimited by a blank character instead of the delimiter.
- The direct command string follows the SYSMAIN command in the next input line; each parameter in the command string must be delimited by the delimiter and **not** by a blank character. If the direct command string is longer than one single line, the CF character (see also the session parameter CF as described in the Natural Parameter Reference documentation) must be placed at the end of the line to continue with the direct command in the next line.

## Invoking SYSMAIN by an Appl. Programming Interface

MAINUSER is a subprogram which allows you to perform the various SYSMAIN functions directly from any user-written object (subroutine, program or subprogram) without going through the normal steps of invoking SYSMAIN. Upon completion of processing of the SYSMAIN functions, the utility is terminated and control is returned to the program, subprogram or subroutine from which the request was issued. MAINUSER can be used in either online or batch mode.

**Note:**

The maximum DATSIZE used by SYSMAIN during processing is 32 KB, depending on the actual request issued.

MAINUSER must not be located in a user library. You must therefore copy it to library SYSTEM on the files FNAT or FUSER or to any SYS-prefixed library which is the steplib for the application.

MAINUSER is invoked with the CALLNAT statement and its relevant parameters (see the CALLNAT statement in the Natural Statements documentation). The parameters are:

Parameter	Explanation
Command (A250)	The direct command string to be executed by SYSMAIN.
Error (N4)	The return code issued by SYSMAIN at the end of processing to indicate a normal end of processing or an error.
Message (A72)	The message corresponding to the error given online.
Library (A8)	The library containing SYSMAIN. If not specified, the default is SYSMAIN.

MAINUSER is invoked as follows:

```
CALLNAT 'MAINUSER' command error message library
```

An example of a callable routine is the program MAINCALL in the library SYSMAIN.

MAINUSER must **not** be invoked from within the library SYSMAIN.

# SYSMAIN Functions and Function Processing

The SYSMAIN functions are executed from menu screens (in menu-driven mode) by entering either an appropriate function code on the menu or direct commands in the command line, or in batch mode with direct commands. Object selection criteria are specified using parameters and keywords.

This section covers the following topics:

- Commands
  - Function Processing
- 

## Commands

Command	Explanation
COPY	<p>This command copies an object from a source environment to a target environment. The object remains unchanged in the source environment.</p> <p>If the target environment already contains an object with the same name (or in the case of an error message, the same number) as the object to be copied, the specified object is not copied. The REPLACE parameter can be used to overwrite the object in the target environment.</p>
DELETE	<p>This command deletes an object from a source environment.</p> <p>If a class is to be deleted, the system command UNREGISTER is carried out for this class. The class is not deleted if an error occurs in the COM environment.</p> <p><b>Note:</b> If a DDM is deleted with SYSMAIN, the corresponding Natural Security file profile is also deleted.</p>
FIND	<p>This command locates a specific programming object or error message in a source environment. During online processing, a window showing the library currently being searched is displayed.</p>
LIST	<p>This command displays a range of objects within a range of libraries in a source environment.</p>
MOVE	<p>This command transfers objects from a source environment to a target environment. The object is deleted from the source environment and added to the target environment.</p> <p>If the target environment already contains an object with the same name (or in the case of an error message, the same number) as the object to be moved, the specified object is not moved.</p> <p>If a class is to be transferred, the system command UNREGISTER is carried out for this class. The class is not transferred if an error occurs in the COM environment.</p> <p>The REPLACE parameter can be used to overwrite the object in the target environment.</p>
RENAME	<p>This command gives an object a new name using one of the following two options:</p> <ol style="list-style-type: none"> <li>1. Rename the object in the source environment.</li> <li>2. Rename the object and transfer it to another (that is, target) environment.</li> </ol> <p>The RENAME function deletes the original object in the source environment; therefore, you are prompted with an option to retain the original object (if the original object is to be retained, it is not modified).</p> <p>If the target environment already contains an object with the same name (or in the case of an error message, the same number) as the object to be renamed, the specified object is not renamed. The REPLACE parameter can be used to overwrite the object in the target environment.</p> <p>Only a single programming object, environment, profile or rule can be renamed using automated processing. If a range of programming objects, environments, profiles or rules is to be renamed, selective processing must be used. A range of error messages can be renamed with automated processing.</p>
HELP	<p>General help information on the SYSMAIN utility is displayed if you enter a question mark (?) in the Code field of any menu.</p> <p>You can obtain field-specific help by placing the cursor in the field in question and pressing PF1 or entering the appropriate help character in the field in question and pressing ENTER.</p>
EXIT	<p>This command terminates the SYSMAIN utility.</p> <p>PF3 and CLEAR also terminate SYSMAIN if they are pressed when the SYSMAIN Main Menu is displayed; however if they are pressed when any other SYSMAIN menu is displayed, the main menu is displayed.</p>

## Function Processing

When operating in menu-driven mode, an object and function are selected from the SYSMAIN Main Menu.

```

14:19:40          ***** NATURAL SYSMAIN UTILITY *****          1999-12-01
User SAG              - Main Menu -                               Library SYSMAIN

          Code  Object                                     Code  Function

          A    Programming Objects                       C    Copy
          D    Debug Environments                       D    Delete
          E    Error Message Texts                     F    Find
          P    Profiles                                  L    List
          R    Rules                                     M    Move
          S    DL/I Subfiles                             R    Rename
          V    DDMS                                     ?    Help
          ?    Help                                     .    Exit
          .    Exit

Object Code .. A          Function Code .. _

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Menu Exit Copy Del Find List Move Ren

```

An appropriate subfunction menu is then displayed (see the example below for the copy function for programming objects).

The fields contained on the subfunction menus correspond to SYSMAIN parameters; see SYSMAIN Parameters and Keywords.

```

18:47:52          ***** NATURAL SYSMAIN UTILITY *****          1999-11-18
User SAG          - Copy Programming Objects -          Library SYSMAIN

                Code  Function

                A   Copy All/Individual Objects
                C   Copy only Cataloged Objects
                S   Copy only Saved Objects
                W   Copy only Stowed Objects
                ?   Help
                .   Exit

                Code ..... A           Sel. List ... Y
Object Name ..... *_____ Type ..... _____
                Set Number .. __      XREF .. N
Source Library ... OLDLIB__ Database .... 10__ File .. 50__
Target Library ... NEWLIB__ Database .... 10__ File .. 60__
Options Replace ... N           Criteria .... N

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Menu Exit Copy Del Find List Move Ren Fsec Fdic Fnat
    
```

Two other fields contained on the subfunction menus are the command line and Sel. List (selection list).

## Command Line

In the command line, you can enter one of the following:

- A direct command for processing a SYSMAIN function.
- A special command to the SYSMAIN utility. This command can be preceded by SET (see Commands Issued to SYSMAIN) or by a reserved command.
- A Natural system command. If the command is not uniquely identifiable as a Natural system command, it should be preceded by two slashes ( // ) to ensure the correct response from SYSMAIN.

## Selection List

The Selection List determines which type of processing is to occur in menu-driven mode:

Y	A Selection List is displayed containing all objects which meet the specified selection criteria. You can select objects to be processed (see Selective Processing). <b>Y</b> is the default.
N	Objects are processed automatically, without display of an intervening Selection List (see Automated Processing).

Once the required object and SYSMAIN function have been selected, you specify the values of the various parameters.

## Selective Processing

Selective processing is an online facility which displays a selection list listing all objects meeting the specified selection criteria.

- **Menu-driven mode**  
Selective processing is the default type of processing when operating in menu-driven mode (it can be deactivated by entering an **N** in the Sel. List field of the menu).
- **Direct command mode**  
To activate selective processing in direct command mode, either include the keyword **HELP** in the *with-clause* of the direct command or enter a question mark (?) as the final character of the object name; see Direct Commands.

You can then select objects from the selection list for processing. The status of each object (for example, Moved, Copied, Renamed, Not Replaced, etc.) is displayed in the Message column after it has been processed.

Specific details for processing each type of object are discussed in subsequent sections of this section.

An example selection list is displayed in the following section. The third and fourth lines of every selection list are reserved for display of the SYSMAIN direct command and parameters specified. This display is identical to the complete direct command syntax, although some keywords and parameters are optional, as shown in the direct command syntax diagrams.

```

13:01:52          ***** NATURAL SYSMAIN UTILITY *****          1999-11-25
  User SAG              - Copy Selection -              Library SYSMAIN

COPY ALL * WITH XREF N FROM OLDLIB WHERE DBID 10 FNR 32 TO NEWLIB WHERE DBID
10 FNR 51

  C Name      Type  S/C  Message      C Name      Type  S/C  Message
  - - - - -  - - - - -  - - - - -  - - - - -  - - - - -  - - - - -  - - - - -
  _ AA        Progrm S/C      _ T0000001 Progrm S/C
  _ T0000002 Progrm S/C      _ T0000003 Progrm S/C
  _ T0000004 Progrm S/C      _ T0000005 Progrm S/C
  _ T0000006 Progrm S/C      _ T0000007 Progrm S/C
  _ T0000008 Progrm S/C      _ T0000009 Progrm S/C
  _ T0000010 Progrm S/C      _ T0000011 Progrm S/C
  _ T0000012 Progrm S/C      _ T0000013 Progrm S/C
  _ T0000014 Progrm S/C      _ T0000015 Progrm S/C
  _ T0000016 Progrm S/C      _ T0000017 Progrm S/C
  _ T0000018 Progrm S/C      _ T0000019 Progrm S/C
  _ T0000020 Progrm S/C      _ T0000021 Progrm S/C
                                     Listed Library: OLDLIB

          Enter options (above), or '?' (Help) or '.' (Exit) _
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      Help Menu Exit Copy Del Find List Move Ren Canc
    
```

**Automated Processing**

Automated processing is an online or batch facility which requires little or no terminal I/O after a function has been selected (which means that a selection list is not displayed).

- **Direct command mode**  
Automated processing is the default type of processing when operating online in direct command mode.
- **Menu-driven mode**  
To activate automated processing when operating in menu-driven mode, enter an N in the field Sel. List of the menu.
- **Batch mode**  
Automated processing is always used in batch mode.
- **Online report mode**  
To activate automated processing when operating in online report mode, enter an N in the field Sel. List of the menu and enter the command BATCH (or BAT) in the command line of your terminal screen.

Online report mode can be used to obtain the SYSMAIN Batch Report online; that is, a list of the objects that were affected by a SYSMAIN function being executed and of the actions performed on each of these objects (see also the appropriate sections on the processing of Natural objects in batch mode later in this section).

If the command BATCH (or BAT) is entered on the command line, SYSMAIN processes the request as if in batch mode. Hence, only the result of each action is present in a report type format. This form of processing also allows you to use the %H terminal command, with which you can obtain a hardcopy of the report (if required).

The functions DELETE and MOVE always delete objects in the source environment. Therefore, during online automated processing, a special confirmation screen is displayed, which gives you the option of continuing or terminating the function.

If the Replace parameter has been set to **Y**, you are given the opportunity to confirm every replace operation **before** it is done. A window is displayed, and you can choose to:

- replace the object indicated,
- not replace the object indicated (default),
- terminate the processing of the function by either entering a period (.) or pressing PF3.

If the target environment already contains an object with the same name as the object to be copied, moved or renamed, the specified object is not processed and processing continues with the next object. The REPLACE parameter can be used to override this feature.

The status of individual objects is not displayed, but an appropriate SYSMAIN message is displayed upon completion of processing. However, if the following message is displayed, it indicates that some objects were not processed:

**NAT4893 NORMAL END BUT SOME DATA WERE REJECTED**

Error message NAT4810 (see the section Data Rejected) lists reasons why an object may not have been processed. Additional reasons for an object not to be processed are discussed in specific sections relevant to each object type.

Batch mode or selective processing should be used if it is necessary to see the status of each object after it is processed.

## Enhanced Selection Criteria

When you select programming objects to be processed by SYSMAIN, in addition to the selection criteria already available, you can also select objects by the date/time, user ID and terminal ID related to their saving or cataloging. For example, you can select only those objects that were cataloged on a specific day between 8:00 and 12:00 by a specific user on a specific terminal, which means that the processing of objects according to the selection criteria is based on all selected criteria as a whole, not on each condition.

When you enter **Y** in the Criteria field in any screen for programming objects, a window is provided where you can enter your additional selection criteria.

```

18:49:47          ***** NATURAL SYSMAIN UTILITY *****          1999-11-18
User SAG          - Copy Programming Objects -          Library SYSMAIN
+-----+
!      --- Additional Criteria ---      !
!                                       !
! Object Type ..... _____      !
! Date/Time From .. _____      !
! Date/Time To ... _____      !
! User ID ..... _____      !
! Terminal ID ..... _____      !
!                                       !
! Command ==>                          !
!                                       !
Code .. !
Object Name .. +-----+
                Set Number .. ___ XREF .. N
Source Library ... OLDLIB__ Database .... 10__ File .. 32__
Target Library ... NEWLIB__ Database .... 10__ File .. 32__
Options Replace ... N      + Criteria .... y

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Menu Exit Copy Del Find List Move Ren Fsec Fdic Fnat
    
```

**Note:**

A plus character (+) in front of the Criteria indicates that additional criteria have been specified already.

# SYSMAIN Direct Commands

SYSMAIN functions can be executed using direct commands, which can be issued either at the same time the SYSMAIN utility is invoked or from the command line of a menu screen.

Direct commands consist of keywords and parameters (see SYSMAIN Parameters and Keywords).

You can issue a direct command in the following ways:

- **In any library:**  
Enter SYSMAIN followed by a direct command string.
- **From the command line of the SYSMAIN menu:**  
Enter a direct command string.
- **From a programming object:**  
Invoke the subprogram MAINUSER with the direct command string as a parameter.
- **In batch mode:**  
Enter SYSMAIN followed by a direct command string.

The below section covers the topics:

- General Direct Command Syntax
- Effects of Direct Commands

## General Direct Command Syntax

Below is the general direct command syntax that applies to the SYSMAIN utility. For the syntax and direct commands that applies to each object type, refer to the relevant sections in the SYSMAIN documentation.

```

FUNCTION [OBJECT] object-name [AS new-name]
      { FROM } [LIBRARY] lib-name [where-clause]
      { FM  }
      TO [LIBRARY] lib-name [where-clause] [with-clause]

```

In the function-specific syntax diagrams, either **FROM** or **FM** is shown to make the diagrams easier to read; however, **FM** can always be used as a synonym for **FROM** and vice versa.

### where-clause

The *where-clause* is optional. The syntax is:

```

[WHERE] [DBID dbid] [FNR file-nr] [NAME name] [CIPHER cipher ]
[PASSWORD password]
[DICTIONARY (dbid, fnr, psw, ciph)]
[SECURITY (dbid, fnr, psw, ciph)]
[LANGUAGE language]

```

## with-clause

The *with-clause* is optional. The syntax is:

```
[WITH] [TYPE type] [FMDATE date] [TODATE date] [FMTIME time]
[TOTIME time] [USER user-id] [TID terminal-id] [XREF xref] [HELP]
[REPLACE] [RCOP] [MON] [EXTEND] [NOPROMPT] [SETUSER user]
[SETNO nr] [VDBID dbnr] [VFNR fnr]
```

## Sequence of Syntax Elements

The sequence of the direct command syntax is not completely fixed, thus allowing more flexibility in command entry. The options and rules which apply are:

- *FUNCTION*, *OBJECT* and *object-name* must normally be the first three parameters of the command string (*OBJECT* can be omitted in some cases).
- The *library-name* (for programming objects and error messages) must be specified immediately after the **FROM** and **TO** keywords. (If the optional keyword **LIBRARY** is used, it must be entered between the **FROM** or **TO** keyword and the *library-name*).
- The *where-clause* must always follow the **FROM** or **TO** keyword and the *library-name*; the sequence of the keywords and values within the clause can be specified in any order.
- The keywords and values of the *with-clause* can be specified in any order, and the *with-clause* can be placed in any location within the direct command string, except as in the first three positions.

## Effects of Direct Commands

Direct commands can result in:

- the display of a selection list for selective processing of the object if a question mark (?) followed by a blank is entered immediately following the object name, or if the keyword **HELP** is included in the *with-clause*; see Selective Processing;
- the automatic processing of a single object or range of objects; see Automated Processing.

When a direct command is issued, you normally are returned to the library from which the command was issued. For more detailed information on direct commands, see the sections which relate to each type of object.

# SYSMAIN Parameters and Keywords

Parameters and keywords are the selection criteria used to identify the objects to be processed by the SYSMAIN utility.

Parameters are applicable online and in batch mode; however when using direct commands, a parameter must normally be preceded, or in some cases replaced, by a keyword.

In the following section, keywords are listed under the parameter to which they correspond. They are also included in the direct command syntax for each command.

Parameters are listed alphabetically in this section. They correspond to input fields on a menu screen. After each parameter, its format and length is indicated in parentheses.

<b>A - P</b>	<b>R - S</b>	<b>T - Z</b>
CRITERIA (A1)	REPLACE (A1)	TARGET CIPHER (A8)
DATE FROM (A10)	RULE NAME (A32)	TARGET DATABASE (N5)
DATE TO (A10)	RULE TYPE (A2)	TARGET FILE (N5)
DDM DBID (N5)	SELECTION LIST (A1)	TARGET LANGUAGE (A9)
DDM FNR (N5)	SET NUMBER (N2)	TARGET LIBRARY (A8)
DDM NAME (A32)	SET USER (A8)	TARGET NAME (A8)
ENVIRONMENT NAME (A8)	SOURCE CIPHER (A8)	TARGET PASSWORD (A8)
ERROR NUMBER FROM (N4)	SOURCE DATABASE (N5)	TERMINAL ID (A8)
ERROR NUMBER TO (N4)	SOURCE FILE (N5)	TIME FROM (A5)
ERROR TYPE (A1)	SOURCE LANGUAGE (A9)	TIME TO (A5)
FDIC (A27)	SOURCE LIBRARY (A8)	USER ID (A8)
FSEC (A27)	SOURCE NAME (A8)	XREF (A1)
NEW NAME (A8)	SOURCE PASSWORD (A8)	
NEW NUMBER FROM (N4)	SUBFILE NAME (A8)	
NEW NUMBER TO (N4)	SUBFILE TYPE (A3)	
OBJECT NAME (A9)		
OBJECT TYPE (A15)		
PROFILE NAME (A8)		
PROFILE TYPE (A3)		

This section also covers the following topics:

- Description of Parameters and Keywords
- Additional Keywords for Direct Commands
- Range Notation for SYSMAIN Parameters

## Description of Parameters and Keywords

### CRITERIA (A1)

This parameter indicates whether additional selection criteria are to be used. If you are working in menu-driven mode, the following values can be entered in the input field:

?	A window with help information about the Criteria parameter appears.
N	No additional selection criteria are to be used. (This is the default.)
Y	A window is displayed where you can specify the additional selection criteria listed below.

### OBJECT TYPE (A15)

Keyword: **TYPE**

The specific type of programming object to be processed (see description of the OBJECT TYPE parameter of the programming objects processing screens).

In menu-driven mode, you can obtain a list of all programming object types available when you enter a question mark (?) in the Type field; a corresponding window appears.

If one or more object types have already been specified on the corresponding programming object processing screen, the OBJECT TYPE parameter of the additional criteria window is already preset with the same specification when the window appears.

### DATE FROM (A10)

Keyword: **FMDATE**

Date on which the programming object was cataloged or saved. All programming objects cataloged or saved on or after this date are selected. The date must be specified according to the setting of the DTFORM parameter.

### TIME FROM (A5)

Keyword: **FMTIME**

Time on which the programming object was cataloged or saved on a specific date. All programming objects cataloged or saved on or after this date and time are selected. The time must be specified in the format *HH:II* (HH = hours, II = minutes).

### DATE TO (A10)

Keyword: **TODATE**

If entered alone (not in conjunction with the Date From parameter), all programming objects cataloged or saved up to this date are selected. The date must be specified according to the setting of the DTFORM parameter.

A date range can be specified by entering Date From and Date To values.

**TIME TO (A5)**

Keyword: **TOTIME**

If entered alone (not in conjunction with the Time From parameter), all programming objects cataloged or saved up to this time on a specific date are selected. The time must be specified in the format *HH:II* (HH = hours, II = minutes).

A time range can be specified by entering Time From and Time To values.

**USER ID (A8)**

Keyword: **USER**

All programming objects cataloged or saved by the specified user are selected.

**TERMINAL ID (A8)**

Keyword: **TID**

All programming objects cataloged or saved on the specified terminal are selected.

**DDM DBID (N5)**

Keyword: **VDBID**

Specific DBID under which a DDM was cataloged. Only DDMs with this DBID are processed. If this parameter is left blank or a **0** is entered, there is no database verification and all databases are selected.

**DDM FNR (N5)**

Keyword: **VFNR**

Specific FNR under which a DDM was cataloged. Only DDMs with this FNR are processed. If this parameter is left blank or a **0** is entered, there is no file number verification and all file numbers are selected.

**DDM NAME (A32)**

The name of the DDM to be processed. See also Range Notation for SYSMAIN Parameters.

**ENVIRONMENT NAME (A8)**

The name of the debug environment to be processed. See also Range Notation for SYSMAIN Parameters.

**ERROR NUMBER FROM (N4)**

The number of the error message to be processed. Each Natural or user-supplied error message within a Natural library is uniquely defined by the error number.

**ERROR NUMBER TO (N4)**

Keyword: **THRU**

Used in conjunction with the Error Number From parameter to specify a range of error numbers.

**ERROR TYPE (A1)**

Keyword: **TYPE**

The specific type of error message to be processed:

S	short error message
E	extended error message
A	all error message types (default)

**Note:**

This parameter is only applicable with direct commands. In menu-driven mode, error message types are shown as subfunctions.

**FDIC (A27)**

Keyword: **DIC**

Specifies the Adabas security for the FDIC source and/or target system file. (See the sections on direct commands for specific objects for details regarding syntax.)

**FSEC (A27)**

Keyword: **SEC**

Specifies the Adabas security for the FSEC source and/or target system file. (See the sections on direct commands for specific objects for details regarding syntax.)

**NEW NAME (A8)**

Keyword: **AS**

The name to be given to a programming object, debug environment, profile **or** rule when it is renamed with the RENAME function.

**NEW NUMBER FROM (N4)**

Keyword: **AS**

The new number to be assigned to an error message when it is renamed with the RENAME function.

**NEW NUMBER TO (N4)**

Keyword: **THRU**

Used in conjunction with the New Number From parameter to specify a range of error numbers.

If you are renumbering a range of error messages within the same environment, the range values must not overlap. For example, it is not possible to rename error numbers 1 - 6 as new error numbers 5 - 10.

**OBJECT NAME (A9)**

The name of the programming object to be processed. If the LIST function is selected, this parameter is referred to as **OBJECT NAME START VALUE** (see also Range Notation for SYSMAIN Parameters).

## OBJECT TYPE (A15)

Keyword: **TYPE**

The specific type of programming object to be processed:

P	program
N	subprogram
S	subroutine
M	map
H	helproutine
Y	expert model
R	report
A	parameter data area
G	global data area
L	local data area
C	copycode
T	text
Z	recording
O	ISPF macro
3	dialog
4	class
5	processor
*	all programming object types (default)

**Note:**

You can specify several types at the same time and in any sequence. For example, if you specify PAM, programs, parameter data areas and maps are processed.

If one or more object types have already been specified in the window for additional selection criteria, the same specification is displayed for the OBJECT TYPE parameter in the corresponding programming object processing screen, once you have left the window (see also the OBJECT TYPE parameter as part of the enhanced selection criteria).

**PROFILE NAME (A8)**

The name of the profile to be processed. See also Range Notation for SYSMAIN Parameters.

**PROFILE TYPE (A3)**

Keyword: **TYPE**

The specific type of profile to be processed:

E	editor profile
D	device profile
M	map profile
P	parameter profile
*	editor, device and map profiles (default)

**REPLACE (A1)**

Keyword: **REP**

Replace option for an object which is being moved, copied or renamed. If you are working in menu-driven mode, the following values can be entered in the input field:

Y	An object with the same name which is already present in the Target library is to be replaced.
N	An object with the same name which is already present in the Target library is not to be replaced. N is the default.

**Note:**

If a programming object is replaced it is also deleted from the Natural buffer pool; any existing cross-reference records are also deleted if Predict is installed.

**RULE NAME (A32)**

The name of the rule (automatic or free rule) to be processed. See also Range Notation for SYSMAIN Parameters.

**RULE TYPE (A2)**

Keyword: **TYPE**

The specific type of rule to be processed:

A	automatic rule
F	free rule
AF	both automatic and free rules

**SELECTION LIST (A1)**

Keyword: **HELP**

Indicates whether automated or selective processing is to apply to an object. If you are working in menu-driven mode, you can enter the following values in the input field:

Y	A Selection List is displayed containing all objects which meet the specified selection criteria. You can select objects to be processed; see Selective Processing. <b>Y</b> is the default.
N	Objects are processed automatically, without display of an intervening selection list; see also Automated Processing.

**SET NUMBER (N2)**

Keyword: **SETN**

Supports Predict Sets. Number of the retained Set created with the Predict XREF Save Set facility. You can apply all SYSMAIN processing functions to the objects included in this Set.

If any valid number is specified, SYSMAIN assumes a Predict Set. If no number is specified, normal object processing is assumed.

**SET USER (A8)**

Keyword: **SETU**

Provides the possibility to overwrite the User ID specification for a Predict Set as a part of the security for Predict files. If you are working in menu-driven mode, the security for Predict (FDIC) Files screen is invoked by entering the SET FDIC command or by pressing PF11 (Fdic) in any programming object processing screen.

**Note:**

SET USER is only evaluated if a valid number has been specified for SET NUMBER.

**SOURCE CIPHER (A8)**

Keyword: **CIPH**

Cipher key of the Source file (used in the *where-clause*).

**SOURCE DATABASE (N5)**

Keyword: **DBID**

The number of the database (1 - 253) which contains the object to be processed.

**SOURCE FILE (N5)**

Keyword: **FNR**

The number (1 - 255) of the Natural FNAT, FDIC or FUSER file which contains the object to be processed.

**SOURCE LANGUAGE (A9)**

Keyword: **LANG**

The language in which the error message is written.

Each error message within each library can exist in 1 to 60 languages. The languages can be specified using any combination of language codes.

For information on which language code is assigned to which language, see the Natural system variable \*LANGUAGE in the Natural System Variables documentation.

To select error messages in all existing languages, asterisk notation (\*) can be used.

**SOURCE LIBRARY (A8)**

Keyword: **LIB**

The name of the library which contains the object to be processed or to which the error message is assigned.

If error messages are processed and the field is left blank, the Natural system error messages are processed.

**SOURCE NAME (A8)**

Keyword: **NAME**

The DDNAME/FCT entry for the Source file number (VSAM only).

**SOURCE PASSWORD (A8)**

Keyword: **PSW**

Password for the Source file (used in the *where-clause*).

**SUBFILE NAME (A8)**

The name of the DL/I subfile to be processed. See also Range Notation for SYSMAIN Parameters.

**SUBFILE TYPE (A3)**

Keyword: **TYPE**

The specific type of DL/I subfile (Natural NSB, NDB or UDF) to be processed.

**TARGET CIPHER (A8)**

Keyword: **CIPH**

Cipher key of the Target file (used in *where-clause*).

**TARGET DATABASE (N5)**

Keyword: **DBID**

The number of the database into which the object is to be moved or copied.

**TARGET FILE (N5)**

Keyword: **FNR**

The number of the Natural system file or Predict file into which the object is to be moved or copied.

**TARGET LANGUAGE (A9)**

Keyword: **TO, LANG**

The language in which the error message is to be written.

Each error message within each library can exist in 1 to 60 languages. The languages can be specified using any combination of language codes. For information about language codes, see the system variable \*LANGUAGE in the Natural System Variables documentation.

**TARGET LIBRARY (A8)**Keyword: **LIB**

The name of the library into which the object is to be placed or to which the error message is to be assigned.

If error messages are processed and the field is left blank, the Natural system error messages are processed.

**TARGET NAME (A8)**Keyword: **NAME**

The DDNAME/FCT entry for the Target file number (VSAM only).

**TARGET PASSWORD (A8)**Keyword: **PSW**

Password for the Target file (used in the *where-clause*).

**XREF (A1)**Keyword: **XREF**

Indicates whether SYSMAIN is to support XREF data stored on Predict system files.

N	Cross-reference data are not processed, except when using the DELETE function. If a cataloged object is deleted, SYSMAIN always deletes any existing XREF data for this object.
Y	All cross-reference data are processed.
S	A specified object is processed regardless of whether it has cross-reference data or not. Any existing XREF data are processed, which means that the fact that objects have or have no XREF data is no criterion for rejecting the processing of an object.
F	All cross-reference data are processed and the object must be documented in Predict.

## Additional Keywords for Direct Commands

In addition to the keywords shown with the parameters above, the following keywords can also be used with direct commands to specify selection criteria:

Keywords	Explanation
ALL	All saved and/or cataloged programming objects are selected for processing.
CAT	All cataloged programming objects are selected for processing. (Any corresponding saved programming object is not processed.)
<u>EXTEND</u>	Refers to the List Objects function. If EXTEND is specified, the saved object is also displayed; if EXTEND is not specified, only the object name and directory information are displayed (in batch mode only).
HELP	Activates online selective processing.
IN/FM	Refers to a source environment.
MON	Activates online trace facility.
<u>NOPROMPT</u>	Suppresses all prompts.
<u>RCOP</u>	Used with direct commands to specify that a copy of the object being renamed is to be made.
<u>SAVED</u>	All saved programming objects are selected for processing. (Any corresponding cataloged object is not processed.)
STOWED	All programming objects which are both saved and cataloged are selected for processing.
TO	Refers to a target environment.
WITH	Optional keyword to indicate the start of a <i>with-clause</i> .
WHERE	Optional keyword to indicate the start of a <i>where-clause</i> .
.	End of command. If this character is detected anywhere within a command string, all subsequent data are ignored.

## Range Notation for SYSMAIN Parameters

All SYSMAIN functions allow the Environment Name, Object (Source) Name, Profile Name, Rule Name, Subfile Name and DDM Name parameters to be specified as a range.

In addition, the "Object Start Value" and "Reposition to" fields for the FIND and LIST functions also specify a range.

With the FIND and LIST functions, also a range of libraries with programming objects can be specified with the Source Library parameter. The same applies to the LIST function with debug environments and to the FIND function with error messages. Using this option, however, can be rather time-consuming depending on how often the selection criteria occur.

You can use the following range notation:

<i>value*</i>	<p>All libraries or objects whose names begin with <i>value</i> are processed.</p> <p>Example: A value of MENU* results in processing of all libraries or objects with a name beginning with MENU, such as MENU1, MENUOFF and MENUX.</p> <p><b>Note:</b> A question mark (?) has the same effect as an asterisk (*).</p>
<i>value&gt;</i>	<p>All libraries or objects whose names begin with a value greater than or equal to <i>value</i> are processed.</p> <p>Example: A value of MEN&gt; results in processing of all libraries or objects whose name begins with a value greater than or equal to MEN, such as MENU, ORDER, SYSDDM and TRS.</p>
<i>value&lt;</i>	<p>All libraries or objects whose names begin with a value less than or equal to <i>value</i> are processed.</p> <p>Example: A value of MEN&lt; results in processing of all libraries or objects whose name begins with a value less than or equal to MEN, such as MAINMENU, CMD and ADALOG.</p>

# SYSMAIN Programming Objects

All SYSMAIN functions can be performed on programming objects.

In environments in which different FNAT and FUSER files are used, programming objects are stored in the files according to the name of the library in which they are contained. If the library begins with SYS (except for the library SYSTEM), objects are stored in the FNAT files. In all other libraries, the objects are stored in the FUSER files.

The section below covers the following topics:

- Programming Object Menus
- Processing Status
- Direct Commands for Programming Objects
- Programming Objects in Batch Mode
- XREF Considerations for Programming Objects

## Programming Object Menus

The subfunction menu displayed depends on the function selected.

The COPY subfunction menu is shown as the example in this documentation; however, menus for other functions are similar in format.

```

17:54:34          ***** NATURAL SYSMAIN UTILITY *****          1999-11-25
User SAG          - Copy Programming Objects -          Library SYSMAIN
                                                Recat: ON

                Code  Function

                A  Copy All/Individual Objects
                C  Copy only Cataloged Objects
                S  Copy only Saved Objects
                W  Copy only Stowed Objects
                ?  Help
                .  Exit

                Code ..... A          Sel. List ... Y
Object Name ..... *_____          Type ..... _____
                Source Library ... OLDLIB__          Set Number .. __ XREF .. N
                Target Library ... NEWLIB__          Database .... 10__ File .. 32__
                Options Replace ... N          Criteria .... N

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                Help Menu Exit Copy Del Find List Move Ren Fsec Fdic Fnat
    
```

**Note:**

If the profile parameter RECAT=ON has been set, this is indicated on the menus.

The functions listed on the menus for programming objects have the following meanings:

Function	Explanation
A	Process All/Individual Objects Process any object which exists as a saved, cataloged or stowed object.
C	Process only Cataloged Objects Process any object which exists as a cataloged object.
S	Process only Saved Objects Process any object which exists as a saved object.
W	Process only Stowed Objects Process only objects which exist in both saved <b>and</b> cataloged form. The exception to this is copycode and recording, neither of which can be cataloged. They are, however, included in processing when this option is specified.

If the profile parameter RECAT has been set to ON for the Natural session, normal dynamic recatalog option rules apply (see the RECAT profile parameter in the Natural Parameter Reference documentation). The following considerations apply:

- If an object exists in stowed format, neither the saved nor the cataloged object can be processed independently of one another.
- If an object exists only in cataloged format, it may not be possible to select it on the Selection List screen and thus it cannot be processed.
- When automated processing is being used, any object not satisfying these rules is immediately ignored and processing continues with the next object.

If selective processing has been requested, a selection list is displayed. If there are more programming objects in a library than can be displayed on a single screen, the additional objects can be displayed by pressing ENTER.

Select specific programming objects for processing by entering the desired option in column **C** (code) to the left of the object names. A help character (?) in this column invokes a window listing the options available for the specific function. These options make it possible to perform additional functions before actively processing the object.

The options which can be entered on a COPY, DELETE, FIND, LIST, MOVE or RENAME selection list have the following meanings.

Option	Explanation
A	Process Cat. and/or Sav. Object  Process all object types listed in the corresponding S/C column; that is, saved (S) objects, cataloged (C) objects or stowed (S/C) objects.
C	Process Cataloged Object only  Process only the cataloged form of the object, even if there is a corresponding saved object. If C is specified for an object which exists only as a saved object, an error occurs.
S	Process Saved Object only  Process only the saved form of the object, even if there is a corresponding cataloged object. If S is specified for an object which exists only as a cataloged object, an error occurs.
B	Delete Object in Buffer Pool  Select objects and optionally delete them from the Natural buffer pool. Deletion of the specified object(s) must be confirmed by entering DELETE in a window that appears once you have specified the object(s) and pressed ENTER.
H	Hardcopy of Saved Object  Make a hardcopy of the saved object.
I	List Directory Information  Review a cataloged and/or saved object before processing it. It lists directory information and corresponds to the system command LIST DIR.
L	List Saved Object  Review the source code of an object before processing it (valid only with saved objects).
X	Export Object to PC  Download Natural source objects to a personal computer (see also below)  <b>Note:</b> This option is applicable only if Natural Connection is installed. To be able to use this option, the work files 6 and 7 must be specified as PC work files.
Z	Calculate Sizes  Review the various sizes of the saved and cataloged objects, for example, the DATSIZE, ESIZE and MCG size.

If you have Natural Connection installed and choose to download Natural source objects to a PC (that is, to specify option X), a window will be displayed in which you specify the following:

Option	Explanation
Drive	The name of the PC drive to which you want the object(s) to be downloaded.
Path	The path name of the PC directory to which you want the object(s) to be downloaded.
Extension	The name of the extension of the specified PC path. If you specify NS*, the asterisk (*) will be replaced by the specified object type.

**Note:**

If the specified directory does not exist, you will receive an appropriate error message.

With the FIND and LIST functions, options **A**, **C** and **S** are not available. However, it is possible to list libraries. The Library Selection screen is displayed either when you leave the Find Selection or the List Selection screen, or when you specify a range of libraries in the FIND or LIST processing screens.

On the Library Selection screen, you can enter the following options:

Option	Explanation
D	<p>Display Objects in a Lib. (Short)</p> <p>Short list of programming objects in the specified library. The listing of an object only shows the name, the saved/cataloged specification and the type of the object.</p>
L	<p>List Objects in a Lib. (Extended)</p> <p>Extended list of programming objects in the specified library. The listing shows one object per line with the corresponding directory information.</p>
R	<p>Verify Subroutine Usage</p> <p>Provide a list at object level that informs you of the external subroutines used by each object in the specified library. All external subroutines performed by any object are listed, as well as the names of the corresponding cataloged subroutines. If an object is a cataloged subroutine itself, the name of the external subroutine it contains is also provided.</p>
S	<p>List all Subroutines in a Library</p> <p>List the names of all external subroutines used in a library.</p>

If the FIND or LIST function is used, the current setting of the field Object Start Value is of importance. The Object Start Value can be set to blank or an asterisk (\*) to list the entire library; or a range notation can be used. If the library being listed contains no objects which satisfy the range criteria specified, an appropriate message is displayed. This option is useful with large libraries in order to restrict the number of objects being displayed.

The Object Start Value is valid in online mode only, and only when an actual library is to be listed. It is not a selection criterion for the LIST function.

## Processing Status

After the individual objects have been selected on the selection list, they are processed by SYSMAIN and an appropriate message is displayed in the message column.

### Options H, I and L

If the **H**, **I** or **L** option was selected, the message returned upon completion of processing is Printed, Directory or Listed.

### Options B, X and Z

If the **B** option was selected, the message returned upon completion of processing is either "NBP deleted" or Ignored, depending on whether deletion from the buffer pool for the specified object(s) has been confirmed or not.

If the **X** option was selected, the message returned upon completion of processing is Exported, and if the **Z** option was selected, the message is Sized.

### Options A, C and S

If the **A**, **C** or **S** option was selected, the message returned upon completion of normal processing is Copied, Deleted, Moved or Renamed, depending on the function selected.

An object can be processed again using the **A**, **C**, or **S** options, after the **B**, **H**, **I**, **L**, **X** or **Z** option processing has been completed.

Other messages which may be returned are:

Message	Explanation
Replaced	The Replace option was set to <b>Y</b> and the Target object was deleted before the COPY, MOVE or RENAME function was completed.
Not Replaced	The Replace option was set to <b>N</b> and an object with the same Object Name already exists in the target environment. The function was not completed.
Subrtn Exists	The cataloged subroutine had an external subroutine name already used by another cataloged subroutine in the target environment. The function was not completed.
Class Exists	The cataloged class had an external class name already used by another cataloged class in the target environment. The function was not completed.
GUID Exists	The cataloged class had a GUID already used by another cataloged class in the target environment. The function was not completed.
LRec-Err: <i>mn</i>	An error was returned for the class during copying of the class link records. The function was not completed. Possible values: 3 = inconsistency in class records, 4 = no record found.
DB Error: <i>nnn</i>	A database error was returned for the object during processing. The function was not completed.
In Use	An Adabas response code 145 was returned during the UPDATE/ READ processing of an object. The function was not completed.
No Xref	A cataloged object was being processed and the XREF parameter was set to <b>Y</b> or <b>F</b> . No XREF data exist on the FDIC file specified for the object. The function was not completed.
Err NAT2999	A cataloged object was being processed with the XREF parameter set to <b>F</b> . No Predict entry exists on the FDIC file specified for the object. The function was not completed.
Exit: <i>nnn</i>	A user exit routine was active and a non-zero return code was returned by the exit ( <i>nnn</i> = the return code). The function was not completed. See also under User Exit Routines.
< = Found	Marks the objects found in one or more libraries by a FIND function (when the library is listed).
Invalid Date	The FROM or TO dates of the saved and/or cataloged object do not conform to the normal format. The object was not processed. Resave or catalog the object to validate the dates.

## Direct Commands for Programming Objects

The direct command syntax for processing of programming objects is shown in this section. (Since the *where-clause* and *with-clause* syntax are identical for each command, they are only shown once with the COPY and MOVE command syntax below.)

### COPY and MOVE Direct Command Syntax

```

[ COPY ] [ MOVE ] [ ALL
                   CATALOGED
                   SAVED
                   STOWED ] name FM [ LIBRARY ] lib-name [ where-clause ]
                                TO [ LIBRARY ] lib-name [ where-clause ] [ with-clause ]
    
```

#### where-clause

```

[ WHERE ] [ DBID dbid ] [ FNR file-nr ] [ NAME_name ] [ CIPHER_cipher ]
[ PASSWORD ] password
[ PSW ]
[ DIC (dbid, fnr, psw, ciph) ]
[ SEC (dbid, fnr, psw, ciph) ]
    
```

**Note:**

Commas must be used as separators between the values following the DIC and SEC keywords; or if a value is missing. For example: DIC(10,,secret,2a). If the Natural session parameter ID has been set to a comma, use a slash (/) sign as the separator between values.

If no DBID or FNR is specified, and SYSMAIN is called via the system command SYSMAIN or via the subprogram MAINUSER, the following rules apply:

The DBID and FNR of the file from which SYSMAIN was called are always used. If, for example, you enter the command SYSMAIN in the FUSER library, the DBID and FNR of the file are used.

**with-clause**

```
[WITH] [TYPE type] [FMDATE date] [TODATE date] [FMTIME time]
[TOTIME time] [USER user-id] [TID terminal-id] [XREF xref]
[REPLACE] [RCOP] [HELP] [MON] [NOPROMPT]
[SETUSER user-id] [SETNO set-number] [EXTEND]
```

**Examples:**

```
COPY PROG1 FM TESTORD TO ORDERS DBID 1 FNR 6 REP
C PGM* WITH REP TYPE PNS FM PRODLIB TO TESTLIB
```

```
M PROG1 TO NEWLIB
MOVE STOWED * TO NEWLIB WHERE DBID 100 FNR 160 FMDATE 02-11-01 FM OLDLIB
WITH XREF Y
```

**DELETE Direct Command Syntax**

```
DELETE { ALL
        CATALOGED
        SAVED
        STOWED } name [IN [LIBRARY] lib-name [where-clause]] [with-clause]
```

**Examples:**

```
DELETE CAT M> IN LIB ORDERS
DEL * IN TESTLIB DBID 1 FNR 5 NAME SYSNAT
D SA * LIBTEST TYPE GLA
DEL * TYPE PM IN TESTORD FMDATE 02-01-01 TODATE 02-05-31
```

## FIND, LIST and LISTLIB Direct Command Syntax



**Note:**

The direct command LISTLIB is only available in batch mode and is used to obtain a list of library names.

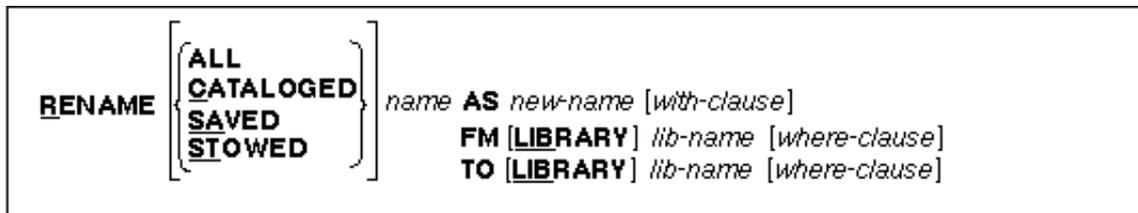
**Examples:**

```
FIND SAVED MENU
FIND STOWED MAINMENU IN SYS* WHERE DBID 1 FNR 5
FIND ALL PROG2 IN PROD* FNR 27 DBID 1
```

```
LIST * IN lib-name
LIST DT* IN lib-name
L SAVED TEST* IN lib-name TYPE PNS FNR 6
L SA TEST* TYPE PM IN lib-name FNR 6 DBID 2 FMDATE 02-01-01
```

```
LISTLIB ALL MENU IN SYS* DBID 10 FNR 44
```

## RENAME Direct Command Syntax



**Examples:**

```
RENAME PGM1 AS PROG1
REN PGM1 AS PROG1 FM TESTLIB DBID 1 FNR 5 TO PRODLIB DBID 2 FNR 6
REN PGM* TYPE PS RCOP FM TESTLIB TO PRODLIB
```

## Programming Objects in Batch Mode

In batch mode, direct commands have to be used to process programming objects.

A report is provided to show the status of programming objects processed in batch mode.

### Note:

By using Online Report Mode, you can obtain the SYSMAIN batch report online. If required, you can also obtain a hardcopy of the report using the **%H** option.

## XREF Considerations for Programming Objects

All XREF data stored in the Predict system file can also be processed with SYSMAIN. The XREF parameter on the COPY, MOVE and RENAME menus indicates whether SYSMAIN processes XREF data. XREF data are always deleted if the DELETE function is performed.

If Predict has not been installed, set the XREF indicator to **N** and thus no validation of Predict files is performed. If the FDIC file(s) being used are not valid Predict files, an error message is returned.

The rules for setting the XREF indicator are the same as the ones imposed by Natural Security, however in a non-security environment there are no restrictions.

### XREF set to N

If the XREF indicator is set to **N** (no), no XREF data are processed, but in situations where a programming object is deleted, SYSMAIN deletes the XREF data. The target Predict system file is determined according to the current settings of the source or target definition in SYSMAIN FDIC. The default is the value assigned to the Natural parameter FDIC at the start of the Natural session.

### XREF set to Y or F

If the XREF indicator is set to **Y** (yes) or **F** (force), the following actions are applied during processing:

- SYSMAIN verifies that XREF data already exist in the Predict system source file.
- If the replace option is active (REPLACE=Y) and a programming object is to be deleted from the target environment, XREF data are deleted from the Predict system target file.
- If a programming object is being copied to a new environment, the XREF data of the programming object are copied from the Predict system source file to the Predict system target file. The library name is changed accordingly and in the case of the RENAME function, the object name is also changed.
- If the MOVE function was requested, the XREF data of the programming object are deleted from the Predict system source file.

### XREF set to F

If the XREF indicator is set to **F** (force), SYSMAIN additionally checks that the programming object (program, subroutine, subprogram, map or help routine only) has a Predict program entry defined on the Predict system target file. If not, processing of the object is terminated.

### XREF set to S

If the XREF indicator is set to **S** (special), the special case applies where a range of specified objects is processed with corresponding XREF data regardless of whether all of the objects have cross-reference data or not: the objects that have cross-reference data are processed with their cross-reference data, and the objects that have none are also processed.

## Errors

If any of the following inconsistencies occur during the SYSMAIN processing of XREF data, all processing for the object or function is terminated and an error message is displayed:

- the value of the XREF indicator in Natural Security is **F** or **Y** and you specified a value of **Y** or **N** respectively,
- the XREF indicator is set to **F** and SYSMAIN finds no documented program entry in Predict for the object being processed,
- an invalid Predict file is specified,
- the value of the XREF indicator in Natural Security is **F** or **Y** and you specified a value of **S**.

# SYSMAIN Debug Environments

All SYSMAIN functions, except the FIND function, can be performed on debug environments.

The debug environment specification must always correspond to the DBID and FNR of the relevant FUSER file.

The section below covers the following topics:

- Debug Environment Menus
- Processing Status
- Direct Commands for Debug Environments
- Debug Environments in Batch Mode

## Debug Environment Menus

The Debug Environments menu contains all of SYSMAIN functions for the processing of debug environments (there are no subfunction menus for debug environments):

```

18:40:20          ***** NATURAL SYSMAIN UTILITY *****          1999-11-18
User SAG          - Debug Environments -                          Library SYSMAIN

                Code  Function

                C    Copy   Debug Environments
                D    Delete Debug Environments
                L    List   Debug Environments
                M    Move   Debug Environments
                R    Rename Debug Environments
                ?    Help
                .    Exit

                Code ..... C                Sel. List ... Y
Environment Name .. TEST1_____ New Name .... _____
Source Library ... OLDLIB__ Database .... 10__ File .. 32__
Target Library ... NEWLIB__ Database .... 10__ File .. 32__
Options Replace ... N

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                Help Menu Exit Copy Del                List Move Ren Fsec                Fnat
    
```

If selective processing has been selected, a selection list is displayed. If there are more debug environments in a library than can be displayed on a single screen, the additional debug environments can be displayed by pressing ENTER.

Select specific debug environments for processing by entering the desired options in column C (code) to the left of the debug environment name. A question mark (?) in this column invokes a window listing the options available for the specific function. With these options you can override the current subfunction setting and perform additional functions before actively processing the environment.

**Note:**

When a debug environment is moved or copied from one library to another, it must be reinstated in the target library. This can simply be done by modifying the corresponding break point or watchpoint.

## Processing Status

After the individual debug environments have been selected on the selection list with option **A** (Process Debug Environment), they are processed by SYSMAIN and an appropriate message (that is, Copied, Deleted, Moved or Renamed) is displayed in the message column depending on the function selected.

Other messages which may be returned are:

Message	Meaning
Exit: <i>nnn</i>	A user exit routine was active and a non-zero return code was returned by the exit ( <i>nnn</i> = the return code); function not completed.
Replaced	Replace option was set to <b>Y</b> and the target debug environment was deleted before the COPY, MOVE or RENAME function was completed.
Not Replaced	Replace option was set to <b>N</b> and a debug environment with the same debug environment NAME already exists in the target environment; function not completed.
Invalid	An invalid code was specified for one of the debug environments listed.
Name Error	Rename function was used, but the new name specified was found to be invalid. Either no code was specified for the selection or the specified name either contained invalid special characters or did not start with an alphabetic character.

## Direct Commands for Debug Environments

The direct command syntax for processing of debug environments is shown in this section. (Since the *where-clause* and *with-clause* syntax are identical for each command, they are only shown once with the COPY and MOVE command syntax below.)

### COPY and MOVE Direct Command Syntax

```
[COPY  
MOVE] DEBUG name FM [LIBRARY] lib-name [where-clause]  
      TO [LIBRARY] lib-name [where-clause] [with-clause]
```

#### where-clause

```
[WHERE] [DBID dbid] [FNR file-nr] [NAME name]  
[CIPHER cipher] [PASSWORD] password  
                  [PSW]
```

#### with-clause

```
[WITH] [REPLACE] [HELP] [RCOP] [MON]
```

#### Examples:

```
COPY D env1 FM oldlib WHERE DBID 1 FNR 5 TO newlib WHERE DBID 2 FNR 5 WITH REP  
COPY DEBUG env FM lib1 FNR 6 TO lib2 FNR 7 REP  
MOVE DEBUG env2 FM lib1 WHERE DBID 1 FNR 5 TO lib2 WHERE DBID 2 FNR 5  
MOVE DEBUG env1 FM oldlib FNR 6 TO newlib FNR 7 REP
```

### DELETE Direct Command Syntax

```
DELETE DEBUG name [IN [LIBRARY] lib-name [where-clause]] [with-clause]
```

#### Examples:

```
DEL DEBUG U* IN lib-name FNR 150  
DEL DEBUG TEST* IN lib-name IN DBID 177 FNR 205
```

## LIST Direct Command Syntax

```
LIST DEBUG name [IN [LIBRARY] lib-name [where-clause]] [with-clause]
```

### Examples:

```
LIST DEBUG env* IN lib-name DBID 1 FNR 5  
LIST DEBUG DT* IN lib-name DBID 10
```

## RENAME Direct Command Syntax

```
RENAME DEBUG name AS new-name [with-clause]  
                IN [LIBRARY] lib-name [where-clause]  
                TO [LIBRARY] lib-name [where-clause]
```

### Examples:

```
R DE env1 AS env2 RCOP  
REN DEBUG env1 AS env2 IN lib-name DBID 1 FNR 4 TO lib-name DBID 1 FNR 5  
REN DEBUG env1 AS newenv IN lib-name FNR 4 TO lib-name FNR 5 REPLACE RCOP
```

## Debug Environments in Batch Mode

Direct commands must be used to process debug environments in batch mode.

A report is provided to show the status of debug environments processed in batch mode.

### Note:

By using Online Report Mode, you can obtain the SYSMAIN batch report online. If required, you can also obtain a hardcopy of the report by using the %H option.

# SYSMAIN Error Messages

All SYSMAIN functions can be performed on error messages. Only authorized users can modify Natural system error messages if Natural Security is installed.

In environments in which different FNAT and FUSER files are used, error messages are stored in the files according to the type of error: Natural system error messages are stored in the FNAT file; user error messages are stored in the FUSER file of the corresponding library.

This section covers the following topics:

- Error Message Menus
- Language Parameter Considerations
- Renumber an Error Message
- Processing Status
- Direct Commands for Error Messages
- Error Messages in Batch Mode

## Error Message Menus

The subfunction menu displayed depends on the function selected. The COPY subfunction menu is shown as the example in this documentation; however, menus for other functions are similar in format.

```

18:43:21          ***** NATURAL SYSMAIN UTILITY *****          1999-11-18
User SAG          - Copy Error Message Texts -          Library SYSMAIN

                Code  Function

                A    Copy Short and/or Extended Texts
                E    Copy only Extended Texts
                S    Copy only Short Texts
                ?    Help
                .    Exit

                Code ..... A          Selection List ..... Y

Error   No. From .. ____   No. To .. ____
Source  Library ... _____   Lang. ... *_____   DBID .. 10___   FNR .. 50___
Target  Library ... _____   Lang. ... *_____   DBID .. 10___   FNR .. 60___
Options Replace ... N

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Menu Exit Copy Del Find List Move Ren Fsec Fnat
    
```

**Note:**

For the processing of Natural system error messages, a library name must not be specified. If the source or target library name contains a question mark (?), an asterisk (\*), a greater than (>) or less than (<) sign, a selection list is displayed with libraries containing error messages (except if the FIND function was used).

The functions listed on the menus for error messages have the following meanings:

Function	Explanation
A	Process Short and/or Extended Texts Process any error message which exists in short and/or extended form.
E	Process only Extended Texts Process any error message which exists in extended form. Only extended error messages with corresponding short error messages are processed.  <b>Note:</b> An extended error message cannot be transferred to a target environment if there is no corresponding short error message in the target environment.
S	Process only Short Texts Process any error message which exists in short form.

If selective processing has been selected, a selection list is displayed. If there are more error messages in a library than can be displayed on a single screen, the additional messages can be displayed by pressing ENTER.

Select specific error messages for processing by entering the desired option in column C (code) to the left of the error number. A question mark (?) in this column invokes a window listing the options available for the specific function. These options make it possible to perform additional functions before actively processing the object.

The options which can be entered on an error message selection list have the following meanings:

Option	Explanation
A	Process Extended and Short Text Process all error message types listed in the corresponding Type column, which means, short (S), extended (E) or short and extended (S/E) messages.
E	Process only Extended Error Text Process only the extended form of the error message. If there is no corresponding short error message in the target environment, the extended error message cannot be processed completely. If E is specified for a message which exists only as a short message, an error is returned.
S	Process only Short Error Text Process only the short form of the error message.
L	List Extended and Short Text Review an error message before processing it. The short and/or extended text is displayed, depending on the subfunction previously specified. The default language corresponds to the codes or values in the Source Language parameter or, if no match is found, to the setting of the system variable *LANGUAGE.

## Language Parameter Considerations

Attention must be given to the setting of the Source and Target Language parameters. These have three possible "logical" settings:

1. If an asterisk (\*) is specified, all languages are treated as one unit. All languages are processed.  
For example, if the source error message exists only in languages 1, 2 and 3, and for the target error message only languages 1, 4 and 6 are defined, after a COPY function, the resulting target message exists only in languages 1, 2 and 3.
2. If the language parameters are specified as individual codes, each occurrence of language in the parameter is processed individually.  
For example, if the source error message contains languages 1, 2 and 3, and the parameter is set to123, and if the target error message contains languages 1, 4 and 6, and the parameter is also set to123, after a COPY function, the resulting target error message contains languages 1, 2, 3, 4 and 6, but only the English target error message is overwritten by the English text of the source error message.
3. If a single language code is specified for the source error message, and multiple language codes are specified for the target error message, the resulting target error message is in the language specified for the source upon completion of the COPY function.  
For example if the source error message contains language 1 and the target error message contains languages 23456, the resulting error message is in language 1 for all the specified languages. This allows the use of the \*LANGUAGE parameter.

## Renumber an Error Message

Error messages can be moved from one library to another, or the languages for each error message can be copied, moved or replaced with the MOVE and COPY functions. In addition, it is possible to renumber a single error message or renumber a range of error messages. This can be done with the RENAME function.

When specifying the ranges, the number of error messages specified by the range "From - To" range of the Source Library must be equal in number to the "From - To" range of the Target Library. In addition, if renumbering error messages within a single library, there must be no overlapping in the New Number range.

**Examples:**

The following examples are valid:

```
REN ERROR 1 THRU 100 AS 101 THRU 200 IN CLAIMS
```

```
REN ERROR 101 THRU 200 AS 1 THRU 100 IN CLAIMS
```

The following examples are invalid, because the number ranges are overlapping:

```
REN ERROR 1 THRU 100 AS 51 THRU 150 IN CLAIMS
```

```
REN ERROR 101 THRU 200 AS 51 THRU 150 IN CLAIMS
```

When large ranges of error messages are being processed, the processing of error messages may require significant resources. In such cases, batch-mode processing may be preferable.

## Processing Status

After the individual error messages have been selected on the selection list, they are processed by SYSMAIN and an appropriate message is displayed in the message column.

### Option L

If the **L** option was selected, the message returned upon completion of processing is Listed.

An object can be processed again using the **A**, **E** or **S** options after the **L** option processing has been completed.

### Options A, E and S

If the **A**, **E** or **S** options were selected, the messages returned upon completion of normal processing are Copied, Moved, Renamed or Deleted, depending on the function selected.

Other messages which may be returned are:

Message	Explanation
Replaced	The replace option was set to <b>Y</b> and the target error message was deleted before the COPY, MOVE or RENAME function was completed.
Not Replaced	The replace option was set to <b>N</b> and an error message with the same error number already exists in the target environment; function not completed.
Exit: <i>nnn</i>	A user exit routine was active and a non-zero return code was returned by the exit ( <i>nnn</i> = the return code); function not completed. See also User Exit Routines.
No Short Err	An extended error message was selected for further processing, but the target error message number has no corresponding short error message; function not completed.
Not Found	An error in the update logic occurred during processing and the requested error message could not be found. This implies that the message was deleted during interim between selection and update.
Ext Exists	The function required a short error message to be deleted, which would have resulted in an extended error message with no corresponding short message; function not completed.
Updated	The text in the specified language did not previously exist for the error number selected, and SYSMAIN has updated the error number with the new language. An error message existed and has now been updated with a new language text.
No Lang 1	Only one language ( <b>E</b> or <b>1</b> ) is available for Natural System extended error message text. An attempt has been made to copy extended error message text, and language <b>E</b> or <b>1</b> has not been included in the LANGUAGE parameter; function not completed.

## Direct Commands for Error Messages

The direct command syntax for processing error messages is shown in this section. (Since the *where-clause* and *with-clause* syntax are identical for each command, they are only shown once with the COPY and MOVE command syntax below.)

For Natural system error messages, Natural-SYSTEM or Natural-SYS has to be specified as library name.

### COPY and MOVE Direct Command Syntax

```

[COPY
MOVE] ERROR number [THRU number]
FM [LIBRARY ] lib-name [where-clause]
TO [LIBRARY ] lib-name [where-clause] [with-clause]
  
```

#### where-clause

```

[WHERE] [DBID dbid] [FNR file-nr] [NAME name] [CIPHER cipher]
[
PASSWORD
PSW
] password
[LANGUAGE language ] [SEC (dbid, fnr, psw, ciph)]
  
```

#### with-clause

```

[WITH] [TYPE type] [REPLACE] [HELP] [RCOP] [MON]
  
```

#### Note:

Commas must be used as separators between the values following the SEC keyword; or if a value is missing. For example: SEC (10,,secret,2a). If the Natural session parameter ID has been set to a comma, use a slash (/) sign as the separator between values.

#### Examples:

```

COPY ERROR 1 FROM ACCOUNTS TO ACCOUNTS1 REP WITH TYPE A
C ERROR 1 THRU 50 FROM ACCT WHERE DBID 1 FNR 10 LANG 123456
TO ACCT WHERE DBID 5 FNR 26 LANG 23456 WITH REP HELP
  
```

```

MOVE ERR 200 THRU 10 FM ACCT FNR 10 LANG 123 TO ACCT LANG 1 TYPE S
M ERR 376 TYPE E FM ACCT LANG E TO ACCT LANG FGSDI
  
```

## DELETE Direct Command Syntax

```
DELETE ERROR number [THRU number]  
[IN [LIBRARY] lib-name] [where-clause] [with-clause]
```

### Examples:

```
DELETE ERROR 1 THRU 10 IN LIBRARY ACCT  
WHERE DBID 1 FNR 2 PSW GUESS CIPH 137561 WITH TYPE E MON HELP  
DEL ERR 100 IN ACCT
```

## FIND Direct Command Syntax

```
FIND ERROR number [IN [LIBRARY] lib-name] [where-clause] [with-clause]
```

### Examples:

```
FIND ERR 4280 IN A* MON  
FIND ERROR 10 IN LIB ACCT WHERE DBID 1 FNR 3 WITH TYPE E
```

## LIST Direct Command Syntax

```
LIST ERROR number [THRU number] [IN [LIBRARY] lib-name]  
[where-clause] [with-clause]
```

### Examples:

```
LIST ERR 1 THRU 10 IN ACCT  
LIST ERROR 100 THRU 150 IN LIB ACCT WHERE DBID 12 FNR 5
```

## RENAME Direct Command Syntax

```
RENAME ERROR number [THRU number] AS new-number  
[THRU new-number] [with-clause]  
IN [LIBRARY] lib-name [where-clause]  
TO [LIBRARY] lib-name [where-clause]
```

### Examples:

```
RENAME ERROR 1 THRU 50 AS 11 THRU 60 WITH TYPE A REP HELP MON RCOP  
IN LIBRARY ACCT WHERE DBID 1 FNR 2 TO LIB ACCOUNT WHERE FNR 3  
RENAME ERR 1 AS 101 IN ACCT
```

## Error Messages in Batch Mode

Direct commands must be used to process error messages in batch mode.

A report is provided to show the status of error messages processed in batch mode.

### Note:

By using Online Report Mode, you can obtain the SYSMAIN batch report online. If required, you can also obtain a hardcopy of the report using the %H option.

# SYSMAIN Profiles

All SYSMAIN functions except the FIND function can be performed on profiles.

The profile environment specification must always correspond to the relevant DBID and FNR of the FNAT file.

This section covers the following topics:

- Profile Menus
- Processing Status
- Direct Commands for Profiles
- Profiles in Batch Mode

## Profile Menus

The Profiles menu contains all of SYSMAIN functions for the processing of profiles (there are no subfunction menus for profiles):

```

18:34:24          ***** NATURAL SYSMAIN UTILITY *****          1999-11-25
User SAG                - Profiles -                               Library SYSMAIN

                                Code  Function
                                C    Copy   Profiles
                                D    Delete Profiles
                                L    List   Profiles
                                M    Move   Profiles
                                R    Rename Profiles
                                ?    Help
                                .    Exit

                                Code ..... C                      Sel. List .. Y
Profile Name ..... *_____ Type ... ____ New Name ... _____
Source Database ... 10____ File ... 32____ Name ..... _____
Target Database ... 10____ File ... 32____ Name ..... _____
Options Replace .... N

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Menu Exit Copy Del          List Move Ren Fsec          Fnate
    
```

If you leave the field Type blank or enter an asterisk (\*), editor profiles, map profiles and device profiles will be processed. To process parameter profiles (as created with the SYSPARM utility), you enter a **P**.

If selective processing has been selected, a selection list is displayed. If there are more profiles in a library than can be displayed on a single screen, the additional profiles can be displayed by pressing ENTER.

Select specific profiles for processing by entering the desired options in column **C** (code) to the left of the profile name (the column contains three bytes, and the options can be entered in any order). A question mark (?) in this column invokes a window listing the options available for the specific function. These options make it possible to perform additional functions before actively processing the object.

In a profile selection list, you can enter the following options:

Process All Profiles	If option <b>A</b> is entered, all profile types listed in the corresponding Type column are processed.
Process Editor Profile	Option <b>E</b> is used to process only the editor profile, even if there are map and device profiles listed as well. If <b>E</b> is specified for an object which does not exist as an editor profile, an error is returned.
Process Device Profile	Option <b>D</b> is used to process only the device profile, even if there are editor and map profiles listed as well. If <b>D</b> is specified for an object which does not exist as a device profile, an error occurs.
Process Map Profile	Option <b>M</b> is used to process only the map profile, even if there are editor and device profiles listed as well. If <b>M</b> is specified for an object which does not exist as a map profile, an error is returned.
Process Parameter Profile	Option <b>P</b> is used to process only the parameter profile (as created with the SYSPARM utility).
List Parameter Profile	Option <b>L</b> is used to display the contents of the parameter profile.

## Processing Status

After the individual profiles have been selected on the selection list, they are processed by SYSMAIN and an appropriate message is displayed in the message column.

Message	Explanation
Exit: <i>nnn</i>	A user exit routine was active and a non-zero return code was returned by the exit ( <i>nnn</i> = the return code); function not completed. See also User Exit Routines.
Dev exists	Replace option was set to N and a device profile already exists in the target environment; all profiles with this name are rejected.
Map exists	Replace option was set to N and a map profile already exists in the target environment; all profiles with this name are rejected.
Edt exists	Replace option was set to N and an edit profile already exists in the target environment; all profiles with this name are rejected.
Replaced	Replace option was set to Y and the target profile was deleted before the COPY, MOVE or RENAME function was completed.
Not Replaced	Replace option was set to N and a profile with the same profile name already exists in the target environment; function not completed.
Not Found	An error in the update logic occurred during processing and the requested profile could not be found; implies that profile was deleted during interim between selection and update.

## Direct Commands for Profiles

The direct command syntax for processing of profiles is shown in this section. (Since the *where-clause* and *with-clause* syntax are identical for each command, they are only shown once with the COPY and MOVE command syntax below.)

### COPY and MOVE Direct Command Syntax

```
{ COPY  
MOVE } PROFILE name [with-clause] FM [where-clause] TO [where-clause]
```

#### where-clause

```
[WHERE] [DBID dbid] [FNR file-nr] [NAME name]  
[CIPHER cipher] [PASSWORD  
PSW] pass word
```

#### with-clause

```
[WITH] [TYPE type] [REPLACE] [HELP] [RCOP] [MON]
```

**Examples:**

```
COPY PROF USER1 TYPE E FM DBID 1 FNR 5 TO DBID 2 FNR 5
COPY PROFILE USER TYPE MED REP FM FNR 6 TO FNR 7
COPY PROF USER1 TYPE P FM DBID 10 FNR 44 TO DBID 3 FNR 7
```

```
MOVE PROF USER1 TYPE E FM DBID 1 FNR 5 TO DBID 2 FNR 5
MOVE PROFILE USER1 TYPE MED REP FM FNR 6 TO FNR 7
```

**DELETE Direct Command Syntax**

```
DELETE PROFILE name [IN [where-clause]] [with-clause]
```

**Examples:**

```
DEL PROF U* TYPE DM
DEL PROF TEST* IN DBID 177 FNR 205
```

**LIST Direct Command Syntax**

```
LIST PROFILE name [IN [where-clause]] [with-clause]
```

**Examples:**

```
LIST PROF USER* IN DBID 1 FNR 5
LIST PROF DT* TYPE E
```

**RENAME Direct Command Syntax**

```
RENAME PROFILE name AS new-name
                        [IN where-clause]
                        [TO where-clause] [with-clause]
```

**Examples:**

```
R PROFILE USER1 AS USER2 RCOP
REN PROF USER1 AS USER2 DBID 1 FNR 4 TO DBID 1 FNR 5
REN PROF USER1 AS NEWUSER IN FNR 4 TO FNR 5 REPLACE RCOP
```

**Profiles in Batch Mode**

Direct commands must be used to process profiles in batch mode.

A report is provided to show the status of profiles processed in batch mode.

**Note:**

By using Online Report Mode, you can obtain the SYSMAIN batch report online. If required, you can also obtain a hardcopy of the report using the %H option.

# SYSMAIN Rules

All SYSMAIN functions can be performed on processing rules.

The processing rule environment specification must always correspond to the relevant DBID and FNR of the FDIC files.

If Predict is installed, it is recommended that you use Predict instead of SYSMAIN for the processing of rules.

This section covers the following topics:

- Rule Menus
- Processing Status
- Direct Commands for Rules
- Rules in Batch Mode

## Rule Menus

The Rules menu contains all of SYSMAIN functions for the processing of rules (there are no subfunction menus for rules):

```

19:12:03          ***** NATURAL SYSMAIN UTILITY *****          1999-11-18
User SAG                - Rules -                                Library SYSMAIN

                                Code  Function
                                C    Copy   Rules
                                D    Delete Rules
                                L    List   Rules
                                M    Move   Rules
                                R    Rename Rules
                                ?    Help
                                .    Exit

Rule      Code ..... C                Sel. List ... Y
Name ..... _____ Type ... ____
New Name .. _____
Source   Database .. 10___ File .. 50__ Name ..... _____
Target   Database .. 10___ File .. 60__ Name ..... _____
Options  Replace ... N

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Menu Exit Copy Del          List Move Ren Fsec Fdic
    
```

The Rule Type is used as part of the selection criteria. Since processing rules are either automatic rules or free rules, only types **A**, **F** or **AF** apply. Only rules which were cataloged with the specified type are selected. With type **AF**, both automatic and free rules are selected. If the Type field is left blank, no verification takes place.

If selective processing has been selected, a selection list is displayed with the following items of information:

Rule Name	The name of the rule that meets the selection criteria and therefore appears on the list.
Type	The type of the selected rule; that is, free rule or automatic rule.
Ver. Type	The verification type is an attribute of the rule which is directly taken from the verification as defined in Predict. A value of Unknown indicates that there is no corresponding verification information in Predict available. For valid values, see the relevant Predict documentation.
Format	The format type is an attribute of the rule which is directly taken from the verification as defined in Predict. A value of Unknown indicates that there is no corresponding verification information in Predict available. For valid values, see the relevant Predict documentation.
Message	The message which indicates the current processing status of the selected rule. See also the following section.

If there are more rules selected than can be displayed on a single screen, the additional rules can be displayed by pressing ENTER.

## Processing Status

A help character (?) in column **C** (code) to the left of the rule name invokes a window that lists the options available for the specific function.

If you want to select a specific rule for processing, enter option **A** (Process Rule). After the individual rules have been selected on the selection list, they are processed by SYSMAIN and one of the following processing status messages is displayed in the message column.

Message	Explanation
Replaced	Replace option was set to <b>Y</b> and the target rule was deleted before the COPY or MOVE function was completed.
Not Replaced	Replace option was set to <b>N</b> , a rule with the same rule name already exists in the target environment, and no new name has been specified; function not completed.
Err: NAT4852	A Natural Security violation occurred; function not completed.
Exit: <i>nnn</i>	A user exit routine was active and a non-zero return code was returned by the exit ( <i>nnn</i> = the return code); function not completed. See also User Exit Routines.

The other options which can be entered on a COPY, DELETE, LIST, MOVE or RENAME selection list have the following meanings:

List DDMs and Fields for this Rule	All DDMs and fields which have the specified rule assigned are displayed. Upon completion of processing the message File Listed is returned.  This option only applies to automatic rules.
Hardcopy of Saved Rule	The rule(s) specified are printed and listed on the screen. Upon completion of processing the message Printed is returned.  This option only applies to saved rules.
List Directory Information	This option lists the source directory of the specified rule(s). Upon completion of processing the message Directory is returned.
List Saved Rule	The source code of the rule specified is listed. Upon completion of processing the message Listed is returned.  This option only applies to saved rules.

## Direct Commands for Rules

The direct command syntax for processing of rules is shown in this section. (Since the *where-clause* and *with-clause* syntax are identical for each command, they are only shown once with the COPY and MOVE command syntax below.)

### COPY and MOVE Direct Command Syntax

```
{ COPY }
{ MOVE } RULE name [AS new-name] FM [where-clause] TO [where-clause] [with-clause]
```

#### where-clause

```
[WHERE] [DBID dbid] [FNR file-nr] [NAME name] [CIPHER cipher ]
      { PASSWORD } password
      { PSW }
      [DIC (dbid, fnr, psw, ciph)]
      [SEC (dbid, fnr, psw, ciph)]
```

#### Note:

Commas must be used as separators between the values following the DIC and SEC keywords; or if a value is missing. For example: DIC(10,,secret,2a). If the Natural session parameter ID has been set to a comma, use a slash (/) sign as the separator between values.

#### with-clause

```
[WITH] [TYPE type] [REPLACE] [HELP] [MON]
```

#### Examples:

```
C RULE TESTRULE FM FNR 20 TO FNR 24 REPLACE
COPY RULE C< FM FNR 20 TO FNR 24
```

```
M RULE TESTRULE FM FNR 20 TO FNR 24 REPLACE
MOVE RULE C< FM FNR 20 TO FNR 24
```

## DELETE Direct Command Syntax

```
DELETE RULE name [with-clause] [IN [where-clause]]
```

### Example:

```
DEL RULE DEMORULE IN DBID 12 FNR 27
```

## LIST Direct Command Syntax

```
LIST RULE name [IN [where-clause]] [with-clause]
```

### Example:

```
L RULE * DBID 1 FNR 5
```

## Rules in Batch Mode

A report is provided to show the status of rules processed in batch mode.

### Note:

By using Online Report Mode, you can obtain the SYSMAIN batch report online. If required, you can also obtain a hardcopy of the report using the %H option.

# SYSMAIN DDMs

All SYSMAIN functions except the FIND and RENAME functions can be performed on DDMs.

The DDM environment specification must always correspond to the relevant DBID and FNR of the FDIC files.

This section covers the following topics:

- DDM Menus
- Processing Status
- Direct Commands for DDMs
- DDMs in Batch Mode

## DDM Menus

The DDMs menu contains all SYSMAIN functions for the processing of DDMs (there are no subfunction menus for DDMs):

```

18:37:11          ***** NATURAL SYSMAIN UTILITY *****          1999-11-18
User SAG                - DDMs -                                Library SYSMAIN

                                Code  Function
                                C    Copy    DDMs
                                D    Delete DDMs
                                L    List    DDMs
                                M    Move    DDMs
                                ?    Help
                                .    Exit

      Code ..... C      Sel. List ... Y
DDM   Name ..... _____
      DDM DBID ... _____ DDM FNR ..... _____
Source Database ... 10___ File ..... 50__ Name ... _____
Target Database.... 10___ File ..... 60__ Name ... _____
Options Replace .... N    Del.NSC-Def. N

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Menu Exit Copy Del          List Move          Fsec Fdic
    
```

The DBID and FNR of the DDM are used as part of the selection criteria. Only DDMs which were cataloged with the DBID and/or FNR specified are selected. If these fields are set to zero, no verification takes place.

If selective processing has been selected, a selection list is displayed.

If there are more DDMs in a library than can be displayed on a single screen, the additional DDMs can be displayed by pressing ENTER.

**Note:**

If a DDM is deleted with SYSMAIN, the corresponding Natural Security file profile is automatically deleted, too.

Select specific DDMs for processing by entering option **A** (Process DDM) in column **C** (code) to the left of the DDM name. A question mark (?) in this column invokes a window listing all options available for the specific function.

The options which can be entered on a COPY, DELETE or MOVE selection list have the following meanings:

Process DDM	Process all DDMs specified. Option <b>A</b> is not available with the LIST function.
List DDM	List the specified DDM before performing any action on it.
List Automatic Rules	List all automatic rules linked to the specified DDM(s). If one or more DDMs have been selected for listing of their automatic rules, the message Rules Listed is displayed.

The field Del. NSC-Def. indicates deletion of Natural Security definitions. If a DDM is deleted from a source environment or moved to a new environment, and different FSEC files have been specified, you can use this parameter to specify whether or not to delete the DDM definition in the source FSEC file. Indicate **N** to keep the DDM definition in the source FSEC. Indicate **Y** to delete the DDM definition in the source FSEC.

**Note:**

This field only appears in a Natural Security environment.

## Processing Status

After the individual DDMs have been selected on the selection list, they are processed by SYSMAIN and an appropriate message is displayed in the message column.

Message	Explanation
Replaced	Replace option was set to <b>Y</b> and the target DDM was deleted before the COPY or MOVE function was completed.
Not Replaced	Replace option was set to <b>N</b> and a DDM of the same name already exists in the target environment; function not completed.
Err: NAT4852	A Natural Security violation occurred; function not completed.
Exit: <i>nmn</i>	A user exit routine was active and a non-zero return code was returned by the exit ( <i>nmn</i> = the return code); function not completed. See also User Exit Routines.

## Direct Commands for DDMs

The direct command syntax for processing DDMs is shown in this section. (Since the *where-clause* and *with-clause* syntax are identical for each command, they are only shown once with the COPY and MOVE command syntax below.)

**Note:**

For compatibility reasons, instead of the keyword DDM you can use the keyword VIEW (its short form V) in direct commands for DDMs.

## COPY and MOVE Direct Command Syntax

```
{COPY}  
{MOVE} DDM name FM [where-clause] TO [where-clause] [with-clause]
```

### where-clause

```
[WHERE] [DBID dbid] [FNR file-nr] [NAME name] [CIPHER cipher ]  
      [ {PASSWORD}  
      PSW password ]  
      [DIC (dbid, fnr, psw, ciph)  
      [SEC (dbid, fnr, psw, ciph)
```

#### Note:

Commas must be used as separators between the values following the DIC and SEC keywords; or if a value is missing. For example: DIC(10,,secret,2a). If the session parameter ID has been set to a comma, use a slash (/) sign as the separator between values.

The DBID, FNR, CIPHER and PASSWORD specifications can be used instead of the corresponding DIC specifications, or vice versa. If an item is specified twice, the one specified last will be used.

### with-clause

```
[WITH] [DDMDBID DDM-dbid] [DDMFNR DDM-fnr] [REPLACE] [HELP] [MON]
```

#### Examples:

```
C DDM PERSONNEL FM FNR 20 TO FNR 24 REPLACE  
COPY DDM C< FM FNR 20 TO FNR 24
```

```
M DDM PERSONNEL FM FNR 20 TO FNR 24 REPLACE  
MOVE DDM C< FM FNR 20 TO FNR 24
```

## DELETE Direct Command Syntax

```
DELETE DDM name [with-clause] [IN [where-clause]
```

#### Example:

```
DEL DDM FINANCE IN DBID 12 FNR 27
```

## LIST Direct Command Syntax

```
LIST DDM name [IN [where-clause]] [[with-clause]]
```

### Example:

```
L DDM * IN DBID / FNR 5
```

## DDMs in Batch Mode

A report is provided to show the status of DDMs processed in batch mode.

By using Online Report Mode, you can obtain the SYSMAIN batch report online. If required, you can also obtain a hardcopy of the report using the **%H** option.

# SYSMAIN DL/I Subfiles

This section covers the following topics:

- General Information
- Processing Status
- Direct Commands for DL/I Subfiles

## General Information

If the Natural interface to DL/I is installed, SYSMAIN can also be used to process DL/I subfiles.

Subfiles are:

- **NDBs** = DL/I DBDs (Database Descriptions) defined to Natural.
- **NSBs** = DL/I PSBs (Program Specification Blocks) defined to Natural.
- **UDFs** = DL/I User-Defined Fields defined to Natural.

For more information on NDBs, NSBs, DBDs, PSBs and UDFs, see the Natural for DL/I documentation.

The DL/I Subfiles menu contains all SYSMAIN functions for the processing of DL/I Subfiles (there are no subfunction menus for DL/I Subfiles):

```

11:42:45          ***** NATURAL SYSMAIN UTILITY *****          1999-11-22
User SAG          - DL/I Subfiles -                               Library SYSMAIN

                                Code  Function
                                C    Copy   Subfiles
                                D    Delete Subfiles
                                L    List   Subfiles
                                M    Move   Subfiles
                                ?    Help
                                .    Exit

                                Code ..... C                      Sel. List ... Y

Subfile Name ..... *_____ Type ... D
Source Database .. 10___ File ... 50__ Name ..... _____
Target Database .. 10___ File ... 60__ Name ..... _____
Options Replace ... N

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Menu Exit Copy Del           List Move           Fsec Fdic
    
```

The SYSMAIN handling of DL/I subfiles is analogous to the handling of DDMs, with the difference that in direct commands you use the keyword DL1 instead of DDM.

In addition, instead of the DDM DBID and FNR, with DL/I subfiles, the subfile type can be specified as part of the selection criteria. Possible types are **D** for NDBs and UDFs, and **P** for NSBs. If the Type field is left blank, an error message is returned.

If selective processing has been selected, a selection list is displayed.

If there are more DL/I subfiles in a library than can be displayed on a single screen, the additional DL/I subfiles can be displayed by pressing ENTER.

Select specific DL/I subfiles for processing by entering option A (see below) in column C (code) to the left of the DL/I subfile name. A question mark (?) in this column invokes a window listing the options available for the specific function.

The options which can be entered on a COPY, DELETE or MOVE selection list have the following meanings:

Option	Explanation
A	Process NSB Subfile or NDB and UDF Subfile Process all DL/I subfiles specified. Option A is not available with the LIST function.
L	List NSB Subfile or NDB and UDF Subfile List the specified DL/I subfile before you perform any action on it.

## Processing Status

After the individual DL/I subfiles have been selected on the selection list, they are processed by SYSMAIN and an appropriate message is displayed in the message column.

Message	Explanation
Replaced	Replace option was set to <b>Y</b> and the target DL/I subfile was deleted before the COPY or MOVE function was completed.
Not Replaced	Replace option was set to <b>N</b> and a DL/I subfile with the same subfile name already exists in the target environment; function not completed.
Err: NAT4852	A Natural Security violation occurred; function not completed.
Exit: <i>nmn</i>	A user exit routine was active and a non-zero return code was returned by the exit ( <i>nmn</i> = the return code); function not completed. See also User Exit Routines.

## Direct Commands for DL/I Subfiles

The direct command syntax for processing DL/I subfiles is shown in this section. (Since the *where-clause* and *with-clause* syntax are identical for each command, they are only shown once with the COPY and MOVE command syntax below.)

### COPY and MOVE Direct Command Syntax

<div style="border: 1px solid black; padding: 5px; display: inline-block;"> <b>COPY</b> <b>MOVE</b> </div>	<b>DL1</b> <i>subfile-name</i> [ <i>with-clause</i> ] <b>FM</b> [ <i>where-clause</i> ] <b>TO</b> [ <i>where-clause</i> ]
--	---

**where-clause**

```
[WHERE] [DBID dbid] [FNR file-nr] [NAME name]
      [CIPHER cipher] [ {PASSWORD} password ]
                        [PSW
```

**with-clause**

```
[WITH] [TYPE type] [REPLACE] [HELP] [MON]
```

**Examples:**

```
COPY DL1 SUBFILE1 TYPE D FM DBID 1 FNR 5 TO DBID 2 FNR 5
COPY DL1 SUBFILE REP FM FNR 6 TO FNR 7 TYPE D
COPY DL1 SUBFILE1 TYPE P TO DBID 3 FNR 7
```

```
MOVE DL1 SUBFILE1 TYPE D FM DBID 1 FNR 5 TO DBID 2 FNR 5
MOVE DL1 SUBFILE1 REP FM FNR 6 TO FNR 7 TYPE D
```

**DELETE and LIST Direct Command Syntax**

```
{DELETE}
LIST } DL1 subfile-name [IN [where-clause]] [with-clause]
```

**Examples:**

```
DEL DL1 S* TYPE D
DEL DL1 TEST* IN DBID 177 FNR 205 TYPE D
```

```
LIST DL1 SUBFILE* IN DBID 1 FNR 5 TYPE D
LIST DL1 SF* TYPE P
```

# Commands Issued to SYSMAIN

Several commands can be issued to the SYSMAIN utility. These commands perform special functions related to the operation of the utility itself.

Command	Function
ADAON / NOADA	Traps abnormal database errors (only applicable online with programming objects) for debugging purposes.
<u>BATCH</u> / <u>NOBATCH</u>	Switches the SYSMAIN utility into batch mode, whereby all processing is done as if SYSMAIN was running in batch. NOBATCH switches the SYSMAIN utility back to online mode.
CLEAR	Clears the current work area. This function can be useful if a large program is in the work area and the SYSMAIN utility therefore requires a larger ESIZE.
<u>DISPLAY</u>	Displays the extended text for the error which has occurred (needs not be preceded by SET).
<u>FINDFIRST</u>	Stops the FIND for programming objects when the first library is found that contains the specified object.
<u>FINDALL</u>	Causes the FIND for programming objects to search in all libraries.
MON / NOMON	Traces the current activity in SYSMAIN. During processing, you are informed as to which object is being read, deleted, updated, added, and whether an error occurs. With programming objects, you are also informed about the action taken with the XREF data. This function is effective only with TP environments which can run in non-conversational mode.
PROMPT / <u>NOPROMPT</u>	Enables or disables the SYSMAIN prompts. For example, before any deletion, SYSMAIN prompts you for confirmation. With NOPROMPT, no confirmation screen is displayed.
SET	Displays a window which explains all special SYSMAIN commands.
SET FDIC	Specifies Adabas security information for the Dictionary/Predict system file. This refers to the Natural parameter FDIC or the keyword DIC in batch mode.
SET FNAT	Specifies Adabas security information for the SYSMAIN source and target system files. This is specified in the <i>where-clause</i> in batch mode.
SET FSEC	Specifies Adabas security information for the Natural Security system file. This refers to the Natural parameter FSEC or the keyword SEC in batch mode.
SET PC	Verifies whether the device is a personal computer (PC). This setting can be intermittently changed with the %+ and %- terminal commands. SET PC then results in SYSMAIN re-verifying the status of the PC parameter.
STATUS	Displays the current values of certain SYSMAIN variables that are important for Software AG Support.

Command	Function
<u>TOTAL</u>	<p>Verifies the actual processing of the last SYSMAIN function executed. The following information is displayed:</p> <p>Read            Total number of objects which were actually read, based on the Object Name specification.</p> <p>Rejected        Total number of objects read which were then rejected, based on the selection criteria specified.</p> <p>Processed      Total number of objects which satisfied the selection criteria.</p> <p>Added          Total number of new objects added to the target environment.</p> <p>Updated        Total number of existing objects updated. (Where possible, SYSMAIN attempts to update existing objects instead of deleting and adding new ones.)</p> <p>Deleted        Total number of objects deleted from either the source or target environment, depending on the function and Replace option.</p> <p>Replaced       Total number of objects which were replaced in the target environment.</p> <p>Not Repl.      Total number of objects which were <b>not</b> replaced in the target environment.</p>
.	Terminates SYSMAIN (SET is not required).

# SYSMAIN PF Keys

You can also use PF keys to perform SYSMAIN functions. PF keys which are not valid within certain menus, are not displayed for this menu. The PF keys are summarized in the table below.

PF Key	Name	Function
PF1	Help	Displays online help depending on current cursor position. If the cursor is positioned in the Object Code or Function Code field, SYSMAIN general help is displayed. If the cursor is in another field, field-specific help is displayed.
PF2	Menu	Displays the SYSMAIN Main Menu.
PF3	Exit	Returns to the previous screen. If you press PF3 on the SYSMAIN Main Menu, SYSMAIN is terminated.
PF4	Copy	Performs the COPY function for the object specified.
PF5	Del	Performs the DELETE function for the object specified.
PF6	Find	Performs the FIND function for the object specified. (Only with programming objects, error messages and rules.)
PF7	List	Performs the LIST function for the object specified.
PF8	Move	Performs the MOVE function for the object specified.
PF9	Ren	Performs the RENAME function for the object specified.
PF10	Fsec	Invokes the security screen for specifying the security information of the FSEC file to Natural Security.
PF11	Fdic	Invokes the security screen for specifying the dictionary information of the Natural FDIC file. (Only applicable for programming objects and for rules and DDMs.)
PF12	Fnat	Invokes the security screen for specifying the system file information of the Natural FNAT and/or FUSER files. (Only applicable for programming objects, debug environments, error message texts and profiles.)
CLEAR	Exit	Returns to the previous screen. If you press CLEAR on the SYSMAIN Main Menu, the SYSMAIN utility is terminated.

# SYSMAIN - Data Rejected

This section covers the following topics:

- Object Rejection and Reasons
  - SYSMAIN Error Notification
- 

## Object Rejection and Reasons

If, during the execution of a SYSMAIN function, one or more objects were found to satisfy the specified selection criteria, but some or all of these objects were then rejected for further processing, an error or NAT4893 occurs. The possible reason for these errors are:

- An object was selected and then rejected because the object type was not valid for the type of processing specified. For example, all maps are rejected if processing is for programs or subroutines.
- An object was selected and then rejected because the date on which it was saved or cataloged did not fall within the range specified by the Date/Time From, Date/Time To, User ID and Terminal ID parameters.
- A cataloged programming object with type **S** (subroutine) was selected and then rejected because the external name was identical to the name of another subroutine in the target library.
- A cataloged programming object with type **4** (class) was selected and then rejected because the external name or the GUID was identical to the name or GUID of another class in the target library.
- An object was selected and then rejected because the target environment already contained an object identified by the same name, and the Replace option was set to **N**.
- A cataloged object was selected and then rejected because the Recat option was **ON** and there was no saved object corresponding to the cataloged object.
- A saved (only) object was selected and then rejected because the Recat option was **ON** and the target environment already contained a cataloged object with the same name.
- The XREF indicator was not set to **N** and there were no XREF data for the programming object specified.
- A user exit routine was active, and a non-zero return code was returned during processing of the object.
- An error message extended text was selected and then rejected because there was no corresponding short message text in the source library.
- An extended error message text was selected but could not be processed because there was no short error message text in the target environment.
- A short error message text was selected to be moved, deleted or renamed, but could not be processed because the corresponding extended error text was not included in the selection criteria. An extended error message must always have a corresponding short error message text.
- The library is controlled by PAC/PAA, and the object can be handled by using the NATLOAD utility only.
- A protected library under the control of Natural Security includes restrictions on object types, which are therefore ignored by SYSMAIN.

You can use the SYSMAIN utility command **TOTAL** to review the specific status of a request.

## SYSMAIN Error Notification

SYSMAIN always attempts to recover in the event of an error during processing. This feature is automatically activated and uses the Natural system variable **\*ERROR-TA**. This feature is deactivated when SYSMAIN is terminated normally.

If the terminal command control character is used to terminate SYSMAIN, this is considered an abnormal termination, and the system variable **\*ERROR-TA** is not reset. It can be reset by re-invoking SYSMAIN and terminating it normally. In the event that you have set the **\*ERROR-TA** system variable, SYSMAIN resets it to its previously assigned value upon termination.

If invalid data have been specified with respect to the selection criteria, an error message is displayed in the message line. If you are uncertain as to the meaning of the short error message, the SYSMAIN command DISPLAY, DISP or DIS can be entered to activate a display of the corresponding extended error message text.

## Data Entry Errors

If invalid data have been specified with respect to the selection criteria, an error message is displayed in the message line. In some situations the online help facility for particular entries is invoked. This feature provides you with more detailed information about the error.

If an error occurs in batch mode, an error message and corresponding error number are printed and the SYSMAIN utility is terminated.

## Processing Errors

If you make a request which causes an error in processing, the SYSMAIN utility ERROR module is invoked, and a window is displayed:

```

11:29:36          *** SYSMAIN Error Report ***          1999-11-25

The following internal error occurred while processing the
SYSMAIN function: XXXXXX (CC)

    Error in field specification for IF SELECTION statement.

Error Number .. EEEE
Program ..... PPPPPPPP
Status Code ... S           Status ..... Unknown Error Type
Line ..... LLLL           Level ..... VV
Library ..... SYSMAIN     Steplib .....
Startup .....             Device ..... PC
User Id ..... SAG         User Name ... USER1

```

The information contained in the window is useful for analyzing the cause of the error.

The values in the window above have the following meanings:

<i>XXXXXX</i>	SYSMAIN function.
<i>CC</i>	An internal status code useful for Software AG support personnel. The following codes can be displayed:  A Automatic processing D XREF data are being deleted E Error in processing (flag for SYSMAIN) F Status setting when XREF data are being processed G Status setting when XREF data are being processed H Selection list processing I Option is being processed S Single object processing T SYSMAIN termination by command processor V Status setting when XREF data are being processed X SYSMAIN termination by command processor Y Validation error has occurred, redisplay should follow Z Validation error has occurred, redisplay should follow
<i>EEEE</i>	Corresponds to the system variable *ERROR-NR.
<i>LLLL</i>	Corresponds to the system variable *ERROR-LINE.
<i>S</i>	C Command processing error. L Logon error. O Object time error. S Non-correctable Syntax error.
<i>VV</i>	Corresponds to the system variable *LEVEL.
<i>PPPPPPP</i>	Corresponds to the system variable *PROGRAM.

If a processing error occurs, note the information in the window and press ENTER. The SYSMAIN utility attempts to recover to the last active menu screen, leaving the data values of the parameters unchanged.

If DISPLAY, DISP or DIS is entered in the window, the extended error message for the error incurred is displayed.

If a processing error occurs during batch processing, the SYSMAIN utility prints the relevant error message and terminates.

Certain user errors can also cause the window to be displayed. Although SYSMAIN attempts to trap all errors during evaluation, this may not always be entirely successful. For example, if a user requests that a DDM be copied from one environment to another, but specifies an invalid DBID, SYSMAIN attempts to access this database. An Adabas response code of 148 is returned, and the SYSMAIN ERROR module is invoked and the window displayed. Similarly, an invalid file can result in a number of errors being sent from the database.

In situations in which an Adabas response code 9 is returned, SYSMAIN writes a message informing you of the error and restart processing from the last function or subfunction menu. If a particular request had not been completed, you can assume that the response code 9 resulted in a BACKOUT TRANSACTION to the last non-completed transaction.

# Special Considerations for Administrators

This section covers the following topics:

- SYSMAIN Security
- User Exit Routines

## SYSMAIN Security

The security aspects of the SYSMAIN utility can be divided into two categories:

- File Security
- Natural Security

### File Security

The file security relates to the security which users can define for their Natural system files, FUSER, FNAT, FDIC and FSEC. This type of security can be used in both Adabas and VSAM environments.

To define file security to the SYSMAIN utility (that is, passwords and cipher codes), invoke the SYSMAIN utility security screens and enter the appropriate security profiles. The security screens are invoked as shown in the following section:

File	Special Command	Key	Objects Affected
FUSER, FNAT	SET FNAT	PF12	programming objects debug environments error messages profiles
FDIC	SET FDIC	PF11	rules DDMs XREF information DL/I subfiles
FSEC	SET FSEC	PF10	Natural Security profile

The security screens are windows similar to the one shown below:

```

--- Security for the NATURAL System Files ---

Specify the password(s), cipher(s) and VSAM FCT
name(s) for the source/target file(s) below:

    - Source -                - Target -
Library .... OLDLIB          Library .... NEWLIB
Database ... 10              Database ... 10
File ..... 32                File ..... 51

Password ...                  Password ...
Cipher .....                  Cipher .....
VSAM Name .. _____      VSAM Name .. _____

```

The FUSER and FNAT files relate to the source and target environments since these environments must relate to the appropriate Natural system file.

**Note:**

No validation is performed for any of the information. The field or entry Library is applicable only when processing programming objects.

If the security information is not provided and the Natural system file requires security, the Adabas security for the appropriate Natural system file is taken as the default.

You can override this feature by invoking the appropriate security screen as mentioned above. Once file security is defined, the SYSMAIN utility uses this security information for all subsequent processing. If a user then requires that the default security information (obtained at initialization of the session) be used, the corresponding security screen must be re-invoked and the password and cipher code fields set to blank. (The password and cipher codes are non-display, so even though they appear to be blank, they should be set to blank again.)

## Natural Security

Two aspects must be considered when using the SYSMAIN utility within a Natural Security environment:

- Defining the Natural Security Environment
- Restricting Use of SYSMAIN under Natural Security

### Defining the Natural Security Environment

The source and target libraries can be within one Natural Security environment or within two different Natural Security environments. These environments must be defined to the SYSMAIN utility.

The definition of the Natural Security environment(s) to be used is specified with the command SET FSEC.

By default the current FSEC setting assigned at the start of the Natural session is used. If you change these settings (on the screen Security for Natural Security), they remain in effect until they are changed by the next SET FSEC process. In batch mode, the SEC parameter should be used to specify the file security and assignments of the request.

Once the source and target environments have been determined, SYSMAIN verifies both the source and target libraries with Natural Security. (The source and/or target database and file must correspond with the DBID and FNR specified in the library security profile; if these values are not specified, default values are taken from the security profile.)

### Restricting Use of SYSMAIN under Natural Security

The use of the SYSMAIN utility itself can be restricted, or the use of the SOURCE and TARGET libraries to be handled with the SYSMAIN utility can be restricted. See Protecting Utilities in the Natural Security documentation for details.

## User Exit Routines

The user exit routines of the SYSMAIN utility provide information about each object being processed.

The user exit routines are Natural subprograms invoked with CALLNAT statements. All subprograms (as well as the data areas they use) are provided in source form in library SYSMAIN.

The source codes of the subprograms are stored under the names SM-UX-*nn* (*nn* = 01 to 11) in library SYSMAIN. To make a user exit routine available, you have to stow the corresponding source under the name MAINEX*nn*, either in the library SYSMAIN or in one of its steplib.

**Note:**

The names of the user exit routines' sources and objects are different to ensure that the overwriting of the sources by an update installation does not affect the objects.

You can change or expand any of the user exit routines as necessary.

Use of these exits results in additional overhead to the SYSMAIN utility, depending on the code logic. It is necessary, however, always to return control to SYSMAIN when exit processing is completed.

As the SYSMAIN utility uses ET logic with Adabas files, the use of user exit routines can lengthen the transaction time limit (Adabas parameter TT). Furthermore, the definition of the Adabas transaction should not be altered, which means that you should not issue any ET/BT commands or END/BACKOUT TRANSACTION statements. SYSMAIN is responsible for the issuing of all END TRANSACTION statements. The exception to this rule is in a situation where a user terminates the normal completion of any SYSMAIN function with the user exit routines. If this is the case you **must** issue a BACKOUT TRANSACTION before terminating.

If the return code is set to a non-zero value, this overrides any error given by SYSMAIN. (See the sections relating to specific object types for a discussion of message field settings.) When an error is received from an exit, it is placed in the message field and displayed or printed as appropriate. The exception is automated processing, because processing is completed with minimum terminal I/O.

The individual user exit routines are described below:

- MAINEX01 - First User Exit Routine for Object Interrogation
- MAINEX02 - Second User Exit Routine for Object Interrogation
- MAINEX03 - User Exit Routine for Request Interrogation
- MAINEX04 - User Exit Routine for Modification of File Assignments
- MAINEX05 - User Exit Routine for Verification of Direct Commands
- MAINEX06 - User Exit Routine for SYSMAIN Initialization
- MAINEX07 - User Exit Routine for SYSMAIN Termination
- MAINEX08 - User Exit Routine for Nothing Found in Batch Mode
- MAINEX09 - User Exit Routine for Abnormal Termination in Batch Mode
- MAINEX10 - User Exit Routine for Command Errors in Batch Mode
- MAINEX11 - User Exit Routine for Setting Special Flags to SYSMAIN

## MAINEX01 - First User Exit Routine for Object Interrogation

Function	Interrogate the current value settings of the data elements associated with an object <b>before</b> the object is processed by SYSMAIN.
Remarks	Any object passed to MAINEX01 can be rejected by setting the RESP-CODE parameter to a non-zero value. If any additional logic is to be performed, the transaction may <b>not</b> be at end-of-transaction status and so no END TRANSACTION or BACKOUT TRANSACTION statement should be issued. Control must be returned to SYSMAIN.
Parameters	PARM-AREA1 (A250) /* SYSMAIN parameter area (fixed values) PARM-AREA2 (A250) /* SYSMAIN parameter area (variable values) RESP-CODE (B1) /* Response code to be returned to SYSMAIN  <b>Note:</b> Only the RESP-CODE parameter can be modified.
Local Data Area	SM-UX-L

## MAINEX02 - Second User Exit Routine for Object Interrogation

Function	Interrogate the current value settings of the data elements associated with an object <b>after</b> the object has been processed by SYSMAIN.
Remarks	Any object passed to MAINEX02 can be rejected by setting the RESP-CODE parameter to a non-zero value. If any additional logic is to be done, the transaction may <b>not</b> be at end-of-transaction status and so no END TRANSACTION or BACKOUT TRANSACTION statement should be issued. Control must be returned to SYSMAIN.
Parameters	PARM-AREA1 (A250) /* SYSMAIN parameter area (fixed values) PARM-AREA2 (A250) /* SYSMAIN parameter area (variable values) RESP-CODE (B1) /* Response code to be returned to SYSMAIN  <b>Note:</b> Only the RESP-CODE parameter can be modified.
Local Data Area	SM-UX-L

## MAINEX03 - User Exit Routine for Request Interrogation

Function	Interrogate any request made to SYSMAN in terms of a direct command or information entered via the online menu system. MAINEX03 obtains control <b>before</b> SYSMAN processes the command.
Remarks	Any command passed to MAINEX03 can be rejected by setting the RESP-CODE parameter to a non-zero value. Additional logic can be added, but it is your responsibility to issue any necessary END TRANSACTION requests to the database. Control must be returned to SYSMAN.
Parameters	<p>PARAM-AREA (A250) /* Command string passed in three parts RESP-CODE (B1) /* Response code to be returned to SYSMAN</p> <p><b>Note:</b> Only the RESP-CODE parameter can be modified.</p>

## MAINEX04 - User Exit Routine for Modification of File Assignments

Function	Override the database, file, password and cipher codes for the Natural system file(s).
Remarks	MAINEX04 is invoked <b>before</b> any request is processed or validated by SYSMAN. When control is passed to MAINEX04, you are at end-of-transaction status; therefore you have to set the RESP-CODE parameter to a non-zero value if you wish to reject the request. Control must be returned to SYSMAN.
Parameters	<p>PARAM-AREA (A250) /* SYSMAN parameter area RESP-CODE (B1) /* Response code to be returned to SYSMAN</p>
Local Data Area	SM-UX-L4

## MAINEX05 - User Exit Routine for Verification of Direct Commands

Function	Verify any direct command entered during online processing of SYSMAN. In addition, the special characters used to indicate a system command can be overwritten.
Remarks	<p>MAINEX05 is invoked <b>before</b> any direct command issued within SYSMAN is processed. For example, MAINEX05 enables you to interrogate any of the special SET commands and also prevent them from being issued. You can verify these commands and reject them by returning a non-zero value in the RESP-CODE parameter. You are at end-of-transaction status when control is passed to MAINEX05.</p> <p>A Natural system command entered within SYSMAN has to be preceded by two slashes (//); see Command Line. With MAINEX05, you can define two other special characters for this purpose; to do so, you assign the desired characters to the parameter CMD-DEL. If CMD-DEL is set to blanks, SYSMAN uses the default value of two slashes (//). Control must be returned to SYSMAN.</p>
Parameters	<p>COMMAND (A68) /* Actual command issued in SYSMAN CMD-DEL (A3) /* Special character for system commands RESP-CODE (B1) /* Response code to be returned to SYSMAN</p>

## MAINEX06 - User Exit Routine for SYSMAN Initialization

Function	Obtain control at initialization of a SYSMAIN session.
Remarks	MAINEX06 is invoked at the start of the SYSMAIN session, where you can override some of the SYSMAIN default settings, as for example, prompts for confirmation of a request like deleting, moving or replacing an object. All parameters are verified. If they are invalid, the default settings are used. Control must be returned to SYSMAIN.
Parameters	MAINEX06 uses the parameter data area SM-UX-L6.

### MAINEX07 - User Exit Routine for SYSMAIN Termination

Function	Obtain control at termination of a SYSMAIN session.
Remarks	MAINEX07 is invoked at termination of a SYSMAIN session to decide whether control is to be kept by SYSMAIN or not.
Parameters	USER-AREA (A50) /* Area for free usage

### MAINEX08 - User Exit Routine for Nothing Found in Batch Mode

Function	Determine further processing if no objects are found for a command in batch mode.
Remarks	MAINEX08 is invoked if no objects are found that meet the specified criteria for a specific command executed in batch mode. If this is the case, control may, but need not, be returned to SYSMAIN. If control is returned to SYSMAIN, SYSMAIN will continue processing with the next command.
Parameters	CMD (A250) /* Command string passed in three parts

### MAINEX09 - User Exit Routine for Abnormal Termination in Batch Mode

Function	Determine action to be taken in case of error in batch mode.
Remarks	MAINEX09 is invoked if SYSMAIN processing in batch mode leads to an error. If this is the case, control may, but need not, be returned to SYSMAIN. If control is returned to SYSMAIN, SYSMAIN will be terminated with condition code 45.  Note: Errors NAT4810, NAT4818, NAT4867, NAT4868 and NAT4893 cannot be handled by this user exit routine.
Parameters	CMD (A250) /* Command string passed in three parts ERROR-CODE (N4) /* Number of error which caused termination

**MAINEX10 - User Exit Routine for Command Errors in Batch Mode**

Function	Determine action to be taken in case of command error in batch mode.
Remarks	MAINEX10 is invoked if an error is detected in a SYSMAIN command in batch mode. If this is the case, control may, but need not, be returned to SYSMAIN. If control is returned to SYSMAIN, SYSMAIN will continue processing with the next command.
Parameters	CMD (A250) /* Command string passed in three parts ERROR-CODE (N4) /* Number of error which caused termination

**MAINEX11 - User Exit Routine for Setting Special Flags to SYSMAIN**

Function	Special settings user exit routine.
Remarks	MAINEX11 is invoked at the start of the SYSMAIN session, where you can set some special SYSMAIN flags, as for example, display of MAINUSER messages in batch. See the source of the user exit routine (SM-UX-11) for the available flags. Control must be returned to SYSMAIN.
Parameters	FLAGS (A250) /*Flag string (redefined)

# SYSNCP Utility - Overview

The utility SYSNCP is used to define command-driven navigation systems for Natural applications.

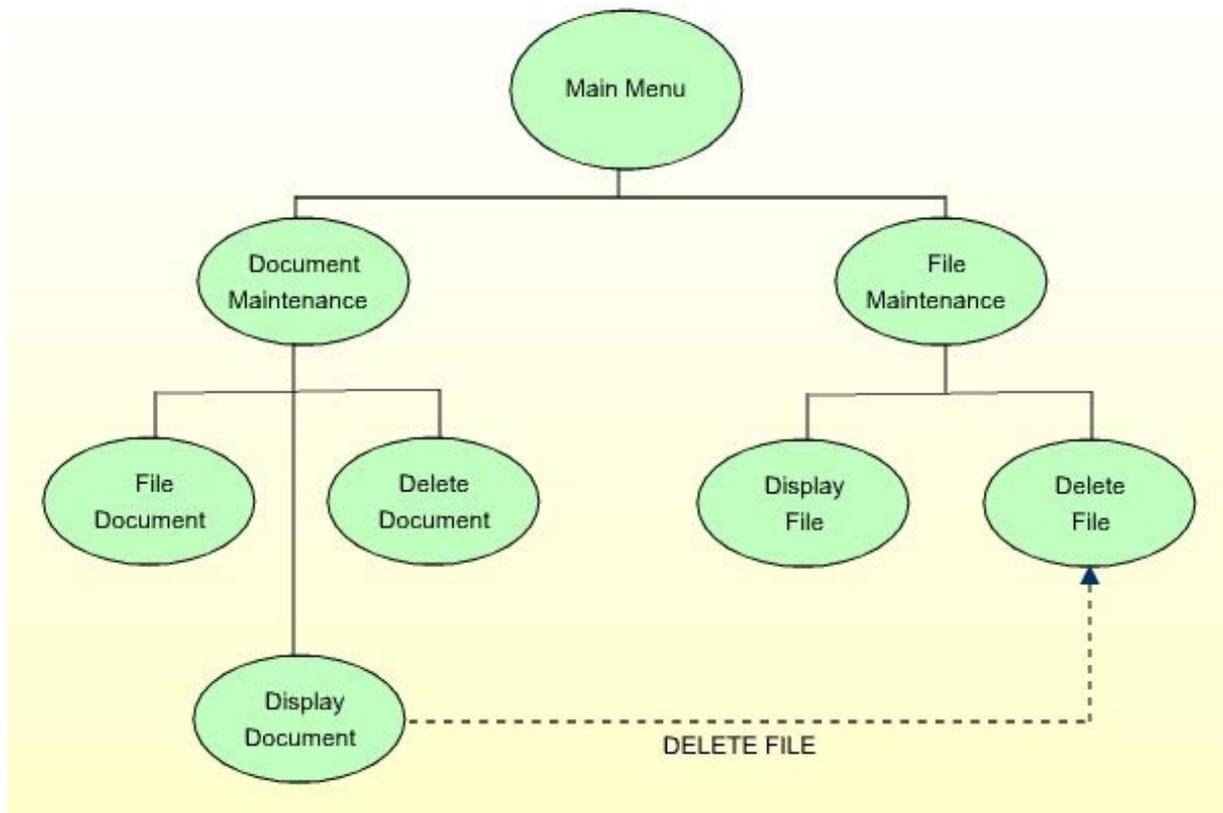
The Natural Command Processor (NCP) consists of two components: maintenance and runtime. The utility SYSNCP is the maintenance part which comprises all facilities used to define and control navigation within an application. The PROCESS COMMAND statement (see the Natural Statements documentation) is the runtime part used to invoke Natural programs.

The SYSNCP utility documentation covers the following topics:

- Introduction Components, features and functionality of the Natural Command Processor.
- Prerequisites Prerequisites for installing the command processor under Windows and (not applicable to mainframes) UNIX.
- Invoking SYSNCP Invoking the SYSNCP utility.
- Processor Selection Invoking the Select Processor function.
- Header Records Defining global settings for a command processor.
- Keyword Maintenance Defining components for functions.
- Function Maintenance Defining functions.
- Runtime Actions Combining functions with runtime actions.
- Processor Cataloging Generating a Natural object of type Processor.
- Administrator Services Administration tasks such as, deleting, loading and securing a processor source.
- Session Profile Defining SYSNCP utility standards for screen layouts and functions such as, color and key assignments.

# SYSNCP - Introduction

Applications which enable users to move from one activity to another activity by using direct commands far exceed in usability the ones which force the user to navigate through menu hierarchies to a desired activity.



The figure above illustrates the advantage of using direct commands. In an application in which menu hierarchies form the basis for navigation, a user wishing to advance from the Display Document facility to the Delete File facility would have to return to the Main Menu via the document branch and then enter the file branch. This is clearly less efficient than accessing the Delete File facility directly from the Display Document facility.

Below is information on:

- Object-Oriented Data Processing
- Features of the Command Processor
- Components of the Command Processor
- What is a Command?
- Creating a Command Processor

## Object-Oriented Data Processing

The Natural command processor is used to define and control navigation within an application. It could be used, for example, to define a command DISPLAY DOCUMENT to provide direct access to the Display Document facility. When a user enters this command string in the command line of a screen (for which this command is allowed), the Natural command processor processes the input and executes the action(s) assigned to the command.

In contrast to menu-driven applications, the command-driven applications implemented with the Natural command processor take a major step toward object-oriented data processing. This approach has the following advantages:

- The design of an application need not depend on the way in which a certain result can be reached, but only on the desired result itself. Thus, the design of an application is no longer influenced by the processing flow within its components.
- The processing units of an application become independent of one another, making application maintenance easier, faster and much more efficient.
- Applications can be easily expanded by adding independent processing units. The resulting applications are, therefore, not only easy to use from an end-user's view, but also easier to create from a programmer's view.

The Natural command processor has the following additional benefits:

- **Less Coding**  
Instead of having to repeatedly program lengthy and identically structured statement blocks to handle the processing of commands, you only have to specify a PROCESS COMMAND statement that invokes the command processor; the actual command handling need no longer be specified in the source code. This considerably reduces the amount of coding required.
- **More Efficient Command Handling**  
As the command handling is defined in a standardized way and in one central place, the work involved in creating and maintaining the command-processing part of an application can be done much faster and much more efficiently.
- **Improved Performance**  
The Natural command processor has been designed with particular regard to performance aspects: it enables Natural to process commands as fast as possible and thus contributes to improving the performance of your Natural applications.

## Features of the Command Processor

The Natural command processor provides numerous features for efficient and user-friendly command handling:

- **Flexible Handling of Commands**  
You can define aliases (that is, synonyms for keywords), and abbreviations for frequently used commands.
- **Automatic Check for Uniqueness of Abbreviated Keywords**  
The command processor automatically compares every keyword you specify in SYSNCP with all other keywords and determines the minimum number of characters in each keyword required to uniquely identify the keyword. This means that, when entering commands in an application, users can shorten each keyword to the minimum length required by the command processor to distinguish it from other keywords.
- **Local and Global Validity of Commands**  
You can specify in SYSNCP whether the action to be performed in response to a specific command is to be the same under all conditions or situation-dependent. For example, you can make the action dependent on which program was previously issued. In addition, you can define a command to be valid under one condition but invalid under another.
- **Error Handling for Invalid Commands**  
You can attach your own error-handling routines to commands or have error input handled by Natural.
- **Functional Security**  
With Natural Security, library-specific and user-specific conditions of use can be defined for the tables generated with SYSNCP. Thus, for your Natural applications you can allow or disallow specific functions or keywords for a specific user. This is known as functional security.  
See also the section Functional Security in the Natural Security documentation.
- **Help Text**  
In SYSNCP, you can attach help text to a keyword or a command. Then, by specifying a PROCESS COMMAND ACTION TEXT statement, you can return command-specific help text to the program.

- **Online Testing of Command Processing**

If the execution of a command does not produce the intended result, you can find out why the command was not processed correctly by using the `PROCESS COMMAND` statement (see the Natural Statements documentation) and the `EXAM*` sample test programs (source form) provided in the Library `SYSNCP`. The endings of the `EXAM-*` program names appear as abbreviations at the top border line of the relevant action windows (for example, `EXAM-C` appears as `C`).

▶ **To test a command processor at runtime**

1. Enter the direct command `EXAM` to list all test programs.  
The "Demonstrate `PROCESS COMMAND` Statement" window is displayed.
2. Enter Function Code `O` to open a processor.
3. Enter the name of the processor.
4. Choose any of the Functions Codes listed (for example, `C` for `CHECK`) to apply command actions.
5. Enter Function Code `Q` to close the processor.

## Components of the Command Processor

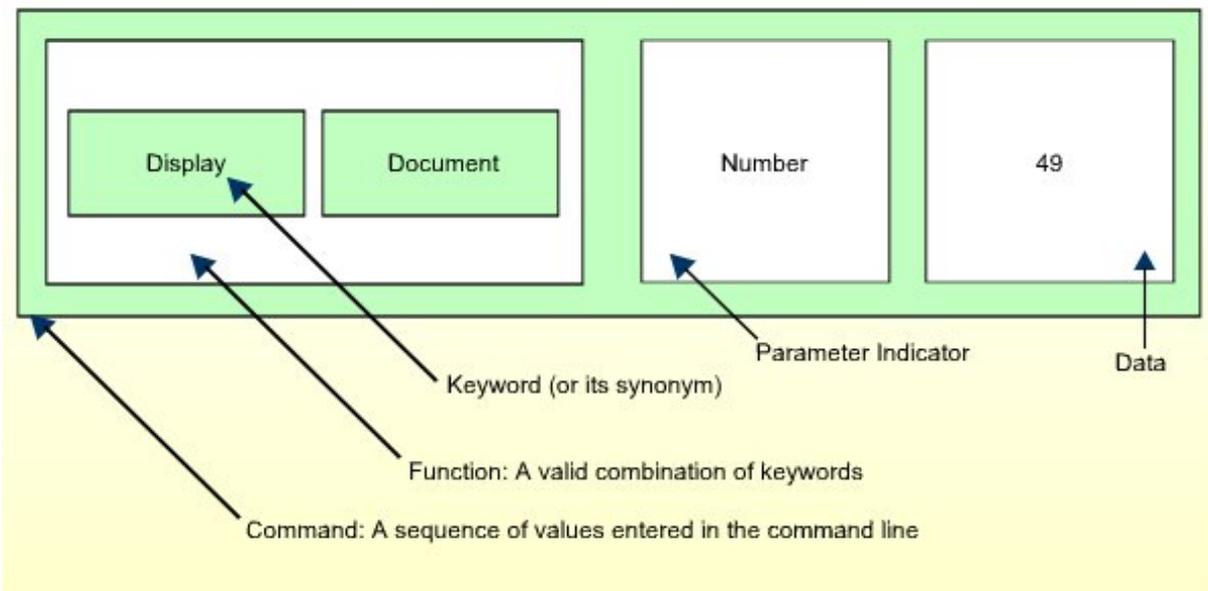
The Natural command processor consists of two parts: a development part and a runtime part:

- The development part is the utility `SYSNCP`, which is described in this section. With the utility `SYSNCP` you define commands (as described below) and the actions to be performed in response to the execution of these commands. From your definitions, `SYSNCP` generates decision tables which determine what happens when a user enters a command. These tables are contained in a Natural member of type `Processor`.
- The runtime part is the statement `PROCESS COMMAND`, which is described in the Natural Statements documentation. This statement is used to invoke the command processor within a Natural program. In the statement, you specify the name of the processor to be used to handle the command input by a user at this point.

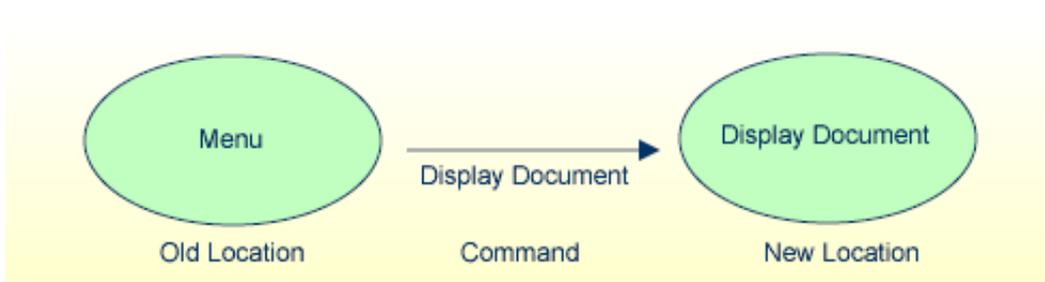
## What is a Command?

A command is any sequence of values entered in the command line which is recognized and processed by an application. Commands can contain up to three elements:

- **Function:**  
One or more valid keywords. For example, MENU or DISPLAY DOCUMENT.
- **Parameter Indicator:**  
Optional. A keyword which introduces command data.
- **Command Data:**  
Information to be sent to a function. Command data can be alphanumeric or numeric, for example, the name or the number of the file to be displayed.



Commands are always executed from a situation within an application; the position where this situation is reached is referred to as a location. Commands take the user from one location to another location; thus, each command can be viewed as a vector:

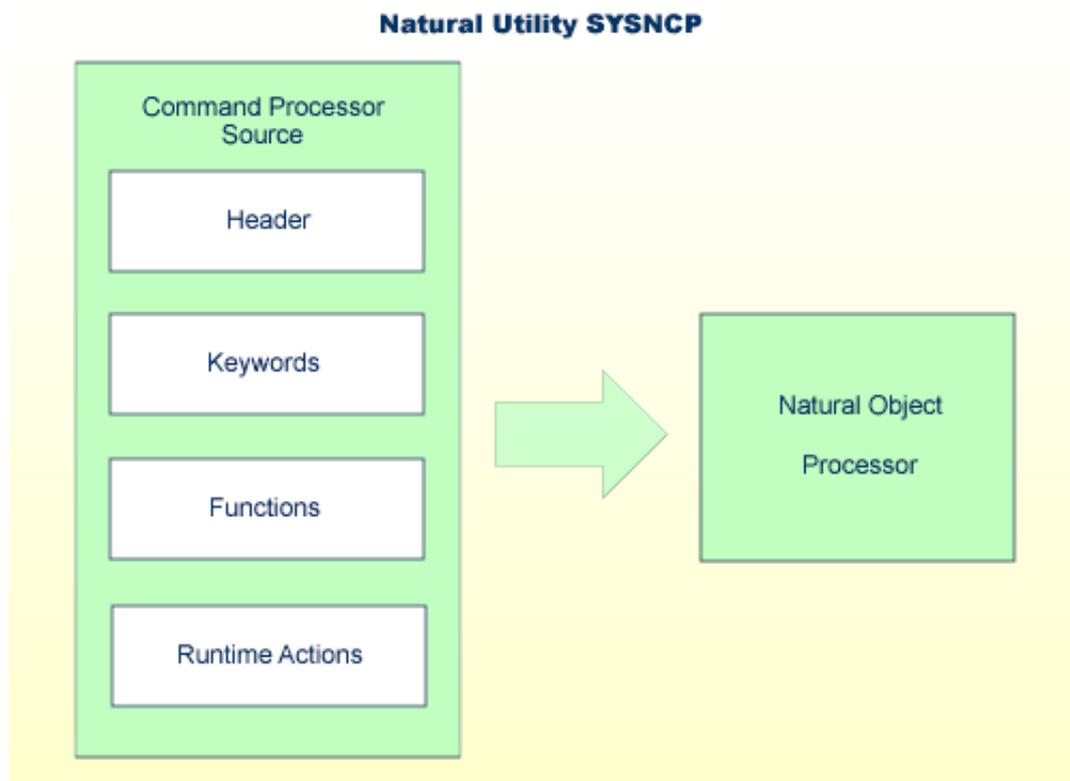


The location from which a certain command can be issued can be restricted on a system-wide or user-specific basis. On a system-wide basis, for example, the functions specified within commands can be local or global. A global function can be issued from **any** location while a local function can only be issued from specified locations. Restrictions can be placed on keywords and functions, however, if Natural Security is active in your environment.

## Creating a Command Processor

The utility SYSNCP is used to create and maintain command processors. A command processor contains decision tables which determine what happens when a user enters a valid command.

The creation of a command processor is a cumulative operation involving several steps, from header definition, which establishes general defaults for the processor, to keyword definition, function definition and the linking of actions to functions. Special editors are provided by SYSNCP for the purpose of specifying keywords, functions and actions.



The end product of command processor development is a complex command processor source, which, when cataloged, generates a Natural object of type Processor. Whenever this object is referenced by the Natural statement `PROCESS COMMAND`, the runtime system of the Natural command processor is triggered.

## Steps in Command Processor Creation

The following is a summary of the steps necessary to create a command processor.

### To create a command processor

1. Verify/Modify the Session Profile.  
SYSNCP itself uses a Session Profile which contains various parameters which control how SYSNCP is to perform certain actions and how information is to be displayed. Desired modifications can be made and the resulting profile can be saved with a given user ID. See the section Session Profile.
2. Initialize the Command Processor.  
The name of the command processor and the library into which it is to be stored are specified.
3. Define Global Settings (Header).  
Various global settings for the command processor are defined. For example, descriptive text for keywords during editing, minimum and maximum length for keywords, in which sequence keywords are to be processed at runtime, runtime error-handling, and whether PF keys can be used at runtime to invoke functions. See the section Header Records.
4. Define Keywords.  
Each keyword which is to be processed by the command processor is defined together with an indication as to whether the keyword is to be entered as the first, second or third entry of a command.  
Keyword synonyms can also be defined as well as parameter indicators. User text can be defined for each keyword. This text can subsequently be read at runtime using the PROCESS COMMAND ACTION TEXT statement. See the section Keyword Maintenance.
5. Define Functions.  
Functions are defined by validating keyword combinations. A function can be defined as local (can only be invoked from a specific location within an application) and/or global (can be invoked from anywhere within an application). See the section Function Maintenance.
6. Define Runtime Actions.  
The actions to be taken by the command processor when a command is issued at runtime are specified.  
Example actions are: fetch a Natural program, place a command at the top of the Natural stack, place data at the top of the Natural stack, change contents of the command line. See the section Runtime Actions.
7. Catalog Command Processor.  
The resulting source is cataloged as a Natural object (type Processor) in the designated Natural library. The command processor can now be invoked by a Natural program using the PROCESS COMMAND statement. See the section Processor Cataloging.

# SYSNCP - Prerequisites

- Windows
  - UNIX
- 

## Windows

- Adabas Version 2.2.3 and above.
- Logical file (LFILE) 190 (NCP Command Proc);  
this file is set as default in the Natural parameter file, do not modify it.
- FDT "SYSTEM-NCP"; see the README file in the folder "demodb".
- Depending on the size of the command processor, the values of the Adabas parameters LP, LQ and NH may have to be adjusted. If these values are too small, you receive error NAT3145 (see the Adabas documentation). With Adabas 3.1.1 for Windows, the profile parameter LQ is no longer used.

## UNIX

- Logical file (LFILE) 190 (NCP Command Proc).
- FDT "SYSTEM-NCP" must be loaded during installation, see the section Installing and Setting Up Natural on UNIX in the Natural for UNIX Installation documentation.

# Invoking SYSNCP

## ▶ To invoke the SYSNCP utility

- Enter the system command SYSNCP.  
The Processor Source Maintenance menu is displayed:

```

18:22:53          ***** NATURAL SYSNCP UTILITY *****          2000-05-22
User SAG          - Processor Source Maintenance -

                Code  Function
                S    Select Processor
                N    Create New Processor
                H    Modify Header
                K    Define Keywords
                F    Define Functions
                R    Define Runtime Actions
                C    Catalog Processor
                A    Administrator Services
                ?    Help
                .    Exit

                Code .. _   Name .. SAGTEST_   Library .. SYSNCP__

Logon to SYSNCP accepted.
Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                Help  Cmd   Exit  Last  List  Flip                                Canc

```

From this menu, you can invoke all functions necessary to create and maintain a command processor. To invoke a function, enter the code letter in the Code field.

### Note:

When you invoke the SYSNCP utility or restart SYSNCP, the user exit NCP-USR1 is invoked for dynamic customization purposes: see the program NCP-USR1 delivered in the Natural system library SYSNCP.

## Help

For help on individual input fields (and also on some output fields) in SYSNCP, place the cursor on the field and press PF1.

# SYSNCP - Processor Selection

The Select Processor function results in a list of all existing command processor sources with related information. If Natural Security is installed, only those sources are listed which can be cataloged to a library to which you are allowed to log on. These restrictions do not apply to those users who have administrator status.

## To invoke the Select Processor function

1. On the Processor Source Maintenance menu, enter Function Code **S**.
2. Press ENTER.

The following information is provided for each processor:

Name	The name of the command processor.
Library	The name of the Natural library for which a processor is created. When the processor is cataloged, it is stored in this library.
User ID	The ID of the user who created the processor.
Date	The date the processor was created.
Status	The stage of development of the processor. For possible status values, see Current Status in the section Header Records.
Cat	Indicates if the processor has been cataloged.

### Note:

With the user exit NCP-SELX (delivered in the Natural system library SYSNCP), you can limit the display to certain processors.

3. In the **Ac** field, enter any character to select a processor.  
The Processor Source Maintenance menu is displayed, where the name of the selected processor is automatically placed in the Name field.  
If you enter a question mark (?) in the Ac field, a window is displayed, listing other possible options.

The name and library name of a command processor can be one to eight characters long. It can consist of upper-case alphabetical characters (A - Z), numeric characters (0 - 9) and the special characters: "-", "/", "\$", "&", "#", "+" and "\_".

# SYSNCP - Header Records

The header maintenance facility defines various global settings for a command processor. These definitions are collectively referred to as a header. Seven header maintenance screens are provided for creating and modifying headers. Header settings for a command processor can be updated at any stage of development (see the following section). After the settings have been modified, the status of a command processor is always set to Header (see also Current Status).

Below is information on:

- Create New Processor
  - Modify Header - General Explanations
  - Keyword Runtime Options - Header 1
  - Keyword Editor Options - Header 2
  - Miscellaneous Options - Header 3
  - Command Data Handling - Header 4
  - Runtime Error Handling - Header 5
  - Statistics - Header 6
  - Status - Header 7
- 

## Create New Processor

### To create a new command processor

- On the Processor Source Maintenance menu, enter Function Code N (Create New Processor), the name of the command processor to be created and the name of the Natural library in which the command processor is to be later cataloged.
- Press ENTER.  
The first header maintenance screen is displayed.

The first header maintenance screen and the following ones are filled with default values that can be edited.

## Modify Header - General Explanations

The Modify Header function is used to maintain an existing header; that is, to modify the various header settings for a given command processor.

### ▶ To modify an existing header

1. On the Processor Source Maintenance menu, enter Function Code **H** (Modify Header), the name of the corresponding command processor and the name of the library into which this command processor has been cataloged.
2. Press ENTER.  
The first header maintenance screen is displayed.
3. Modify any input field in the header maintenance screens described below.
4. Press ENTER to confirm modifications.

Seven different screens are available for the definition and maintenance of a processor header (for the definition of a header, see the previous section).

### ▶ To navigate between the header maintenance screens

- Use PF8 (forward) or PF7 (backward).

Each of the screens contains the following information:

Name	The name of the command processor.
Library	The name of the library into which the resulting command processor object is to be placed after being cataloged.
DBID, FNR	The database ID and file in which the specified library is located.
Created by	The user ID of the Natural user who initialized this command processor.
Date	The date the command processor was initially created.
Current Status	The command processor status: <ul style="list-style-type: none"> <li>Init        The command processor has been initialized.</li> <li>Header     The header for the command processor has been created/modified.</li> <li>Keysave    Keywords have been defined and saved.</li> <li>Keystow    Keywords have been checked and stowed.</li> <li>Function    Keyword combinations have been defined.</li> <li>Action     Runtime actions have been defined.</li> <li>Object     An object form of the command processor has been created.</li> <li>Frozen     The command processor has been frozen.</li> <li>Copied     The command processor has been copied.</li> <li>Error      An error has been detected.</li> </ul>

# Keyword Runtime Options - Header 1

When you select the Modify Header function (as described above), the "Processor Header Maintenance 1" screen is displayed:

```

16:40:19          ***** NATURAL SYSNCP UTILITY *****          2000-05-04
User SAG          - Processor Header Maintenance 1 -

Modify Processor          Name SAGTEST  Library SYSNCP  DBID 10    FNR 32
Created by SAG          Date 2000-04-29          Current Status Init

Keyword Runtime Options:
-----
First  Entry used as ..... Action_____
Second Entry used as ..... Object_____
Third  Entry used as ..... Addition_____

Minimum Length ..... _1
Maximum Length ..... 16
Dynamic Length Adjustment .. -

Keyword Sequence ..... 123_____
Alternative Sequence ..... _____
Local/Global Sequence ..... LG_____

Processor Header with name SAGTEST for library SYSNCP has been added.
Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Cmd  Exit  Last  List  Flip  -      +      Canc
    
```

Various attributes which are to apply for the keywords defined for the command processor are entered on this screen.

Field	Explanation
First Entry used as	<p>A descriptive text which is to be associated with all keywords which are entered as the first entry (entry type 1) when defining a keyword sequence.</p> <p>For example, if the first keyword of a keyword sequence is to represent the action to be performed (DISPLAY, DELETE, etc.), the descriptive text "Action" could be entered in this field.</p> <p>The first four characters of the text entered in this field appear under the column heading Use in the Keyword Editor as described in the section Keyword Maintenance.</p>
Second Entry used as	<p>A descriptive text which is to be associated with all keywords which are entered as the second entry (entry type 2) when defining a keyword sequence.</p> <p>If, for example, the second keyword of a keyword sequence is to represent the object to be used (DOCUMENT, FILE, etc.), the descriptive text "Object" could be entered in this field.</p> <p>The first four characters of the text entered in this field appear under the column heading Use in the Keyword Editor as described in the section Keyword Maintenance.</p>
Third Entry used as	<p>A descriptive text (TITLE, PARAGRAPH, etc.) which is to be associated with all keywords which are entered as the third entry (entry type 3) when defining a keyword sequence.</p> <p>The first four characters of the text entered in this field appear under the column heading Use in the Keyword Editor as described in the section Keyword Maintenance.</p>
Minimum Length	The minimum length permitted when defining a keyword. Valid values are 1 - 16 characters. The default is one character.
Maximum Length	The maximum length permitted when defining a keyword. Valid values are 1 - 16 characters. The default is 16 characters.
Dynamic Length Adjustment	<p>The following values are permitted:</p> <ul style="list-style-type: none"> <li>+ At runtime, each keyword must be entered in its entirety.</li> <li>- At runtime, each keyword can be abbreviated provided that it retains uniqueness with respect to other keywords.</li> <li>S The number of characters which must be entered for a given keyword is to be specified during keyword definition in the ML field of the Keyword Editor as described in the section Keyword Maintenance.</li> </ul>
Keyword Sequence	<p>The sequence in which keyword entries are to be processed at runtime. Possible values are 1, 2, 3 and P (for parameter indicator); the default sequence is 12, which means first the first keyword entry and then the second keyword entry. See also the field E as described in the section Keyword Maintenance.</p> <p><b>Note:</b> A maximum of four entries can be specified; the field is eight characters long to permit expansion at a later release level.</p>
Alternative Sequence	An alternative sequence in which keywords are to be processed at runtime in the event that the default sequence (specified above) results in an error during runtime.

<b>Field</b>	<b>Explanation</b>
Local/Global Sequence	<p>This option specifies the order of command validation to be performed at runtime. Possible values are:</p> <ul style="list-style-type: none"><li>L Command is to be validated as a local command.</li><li>G Command is to be validated as a global command.</li></ul> <p>The default validation sequence is LG, which means that the command is to be validated first as a local command and then (if necessary) as a global one.</p>

## Keyword Editor Options - Header 2

Further keyword attributes can be entered on the "Processor Header Maintenance 2" screen:

Field	Explanation
Header 1 for User Text	These two fields are used to enter a descriptive text which appears in the Keyword Editor above the column reserved for user text. This text is also output during runtime when the TEXT option is specified with the PROCESS COMMAND statement as described in the Natural Statements documentation.
Header 2 for User Text	
Prefix Character 1	This field and the next three are used to attach a hexadecimal prefix to keywords. This enables the processing of internal keywords which cannot be represented by a normal keyboard. When the command processor is cataloged, all prefix characters in keywords are replaced by the hexadecimal values specified.  If a non-blank character is entered in one of the Prefix Character fields, the specified character is replaced by the hexadecimal value specified in the Hexadecimal Replacement field.
Hex. Replacement 1	The value specified in this field replaces the character specified in the field Prefix Character and is used as a prefix for a keyword at runtime.
Prefix Character 2	See above Prefix Character 1.
Hex. Replacement 2	See above Hex. Replacement 1.
Keywords in Upper Case	This option specifies whether keywords are to be translated to upper case in the Keyword Editor and the application:  Y Keywords entered in the Keyword Editor are automatically converted to upper case. In the application, end-users can enter the keywords in upper or lower case.  N Keywords entered in the Keyword Editor are not converted to upper case. In the application, end-users must enter the keywords <b>exactly</b> as they appear in the Keyword Editor.
Unique Keywords	This option specifies whether keywords within the processor must be unique.  Y Each keyword defined must be unique within this processor, regardless of its type.  N Each keyword defined for a given keyword type (1, 2, 3 or P) must be unique.

## Miscellaneous Options - Header 3

Miscellaneous options can be entered on the "Processor Header Maintenance 3" screen:

Field	Explanation
Invoke Action Editor	<p>This option specifies whether the Runtime Action Editor is to be activated from the Function Editor (see the sections Runtime Action Editor and Define Functions).</p> <p>Y The Runtime Action Editor is invoked whenever a valid keyword combination is defined in the Function Editor.</p> <p>N The Runtime Action Editor is suppressed in the Function Editor.</p> <p><b>Note:</b> If you use the user exit NCP-REDM (delivered in the Natural system library SYSNCP), you should set this option to Y; otherwise, invalid runtime action values cannot be detected in time and can lead to runtime errors.</p>
Catalog User Texts	<p>This option specifies whether user texts are to be cataloged with the command processor.</p> <p>Y Text portions of the edit line (Keyword Editor; see the section Define Keywords) and the user text portion of the action line (Runtime Action Editor) are bound to the associated keyword or function when the command processor is cataloged. This text can then be read at runtime using the TEXT option of the PROCESS COMMAND statement.</p> <p>N Texts are not cataloged with the command processor and cannot be read at runtime.</p>
Security Prefetch	<p>This option specifies whether security checking is to be performed when the command processor is initially invoked during runtime or at each command evaluation.</p> <p>Y If Natural Security is installed, security checking is performed for all keywords when the processor is invoked.</p> <p>N If Natural Security is installed, security checking is performed with the evaluation of each keyword.</p> <p>If option Y is selected, security checking is performed only once for all keywords when the command processor is invoked. Since the checking procedure takes time, evaluation of the first command is comparatively slow at runtime, while the evaluation of all remaining commands is comparatively fast. Conversely, if option N is selected, the evaluation time for each command is always the same because security is checked for each keyword individually before it is evaluated.</p>
Command Log Size	<p>Commands processed at runtime can be stored in a command log area by the command processor. Specify in the input field the number of KBs storage space allocated to command logging:</p> <p>0 No storage space is allocated to command logging. Command logging is inactive.</p> <p>1 1 KB of storage space is allocated to command logging. Command logging is active.</p>

Field	Explanation
Implicit Keyword Entry	<p>This option specifies whether a keyword of type 1 is to be retained as an implicit keyword for all subsequent commands.</p> <p>1 If a command is entered which only contains a keyword of type 2, the command processor assumes the most recently entered keyword of type 1 as implicit keyword.</p> <p>N Option is disabled.</p>
Command Delimiter	<p>This option specifies the character used to separate commands if more than one command is specified in the command line. At runtime, only the first command will be executed. For example:  DISPLAY CUSTOMER; MODIFY CUSTOMER; PRINT.</p>
PF-Key may be Command	<p>This option specifies whether commands can be allocated to PF keys: if the command processor receives at runtime a command line which contains all blanks, it checks if a PF key has been pressed by the user.</p> <p>Possible values are:</p> <p>A The identifier for this PF key (system variable *PF-NAME) is used as the command.</p> <p>K The content of the *PF-KEY system variable is used as the command.</p> <p>Y If *PF-NAME is empty, the content of the *PF-KEY system variable is used instead.</p> <p>N PF keys cannot be used as command, Natural error NAT6913 is issued with message "Command line not accepted".</p> <p>For more information on the system variables *PF-NAME and *PF-KEY see the Natural System Variables documentation.</p>

## Command Data Handling - Header 4

The attributes to be entered on the "Processor Header Maintenance 4" screen specify how command data are handled for a function; command data are optional.

Options are:

Field	Explanation
Data Delimiter	Specifies the character to be used to precede data. Default data delimiter is "#". Example: ADD CUSTOMER #123
Data Allowed	Specifies if data input is allowed at runtime.  N A runtime error occurs if data is found. D Data is dropped if present. S Data is placed at the top of the Natural stack. No verification is performed. Y Data is checked and keyword entries of type P (parameter indicator) are evaluated.  Example of Y: DISPLAY CUSTOMER NAME=SMITH
More than one Item Allowed	Only applies if the option Data Allowed is set to Y. Specifies whether more than one data string is permitted.  N A runtime error occurs if more than one data string is found. D All data after the first data string are dropped. Y More than one data string is permitted.  Example: ADD ARTICLE #111 #222  As long as uniqueness is guaranteed, the data delimiter can be omitted.  Example: ADD ARTICLE 123
Maximum Length of one Item	Only applies if the option Data Allowed is set to Y. Specifies the maximum number of characters allowed for a data string. If the specified maximum is exceeded, a runtime error occurs. Valid range: 1 - 99.
Item Must be Numeric	Only applies if the option Data Allowed is set to Y. Specifies whether each data value must be an integer value.  Y Data input must be a positive integer value. If not, a runtime error occurs. N Data can be of any type.
Put to Top of Stack	Only applies if the option Data Allowed is set to Y. Specifies where data is to be placed.  Y Data is placed at the top of the Natural stack. 1-9 Data is placed in the <i>n</i> th occurrence of the DDM field RESULT-FIELD. If the occurrence has already been filled as a result of a runtime action, it is overwritten.

<b>Field</b>	<b>Explanation</b>
If Error, Drop all Data	Only applies if the option Data Allowed is set to Y or N. Specifies the reaction to a data evaluation error:  Y If an error occurs during evaluation of the data, data is discarded and processing continues.  N If an error occurs during data evaluation, control is given to the error handler as described below.

## Runtime Error Handling - Header 5

The attributes to be entered on the "Processor Header Maintenance 5" screen specify how to handle runtime errors:

Field	Explanation
General Error Program	<p>The name of the program which is to receive control when an error is detected during runtime processing by the command processor. The Natural stack contains the following information when this program is invoked:</p> <p style="margin-left: 40px;">Error Number (N4) Line Number (N4) Status (A1) Program Name (A8) Level (N2)</p> <p>If no error program and no specific error handling is specified (see below), the program with the name as contained in the Natural system variable *ERROR-TA is invoked; otherwise, a Natural system error message is issued.</p>
Keyword not found	Indicates whether an action has been specified that is to be performed if a keyword could not be found.
Keyword missing	Indicates whether an action has been specified that is to be performed if the keyword type is missing.
Keyword Sequence Error	Indicates whether an action has been specified that is to be performed in the case of a keyword sequence error.
Command not defined	Indicates whether an action has been specified that is to be performed in the case of an undefined command.
Data disallowed	Indicates whether an action has been specified that is to be performed in the case of disallowed data.
Data Format/Length Error	Indicates whether an action has been specified that is to be performed in the case of a format/length error.
General Security Error	Indicates whether an action has been specified that is to be performed if an error is detected during a general security check.
Keyword Security Error	Indicates whether an action has been specified that is to be performed if an error is detected during a keyword security check.
Command Security Error	Indicates whether an action has been specified that is to be performed if an error is detected during a command security check.

## Statistics - Header 6

The "Processor Header Maintenance 6" screen contains only output fields which report statistical data about the keywords specified for a command processor.

The following statistical information is provided:

Field	Explanation
Entry <i>n</i> Keywords	The number of keywords of type <i>n</i> defined in the command processor (not including synonyms).
Entry <i>n</i> Keywords + Synonyms	The sum of keywords of type <i>n</i> and their assigned synonyms.
Highest IKN for Entry <i>n</i>	The largest Internal Keyword Number for the keyword of type <i>n</i> .
Possible Combinations	The number of possible combinations for keywords defined.
Cataloged Functions	The number of keyword combinations currently cataloged.

## Status - Header 7

The "Processor Header Maintenance 7" screen contains only output fields which report the time and the date when parts of the command processor were executed or modified.

# SYSNCP - Keyword Maintenance

Keywords are the basic components for defining functions. Before it is possible to define keywords, the header maintenance records must be created (see the section Header Records).

- Define Keywords
- Editor Commands
- Positioning Commands
- Line Commands

## Define Keywords

Keywords used in commands are created with the Define Keywords function and the Keyword Editor. The Keyword Editor is similar to existing Natural editors except that lines of the editor are broken up into separate fields. Most of the editor commands (see the relevant section) and the line commands (see the relevant section) which are used in the Natural program editor can also be used in the Keyword Editor.

 **To invoke the Keyword Editor**

1. On the Processor Source Maintenance menu, enter Function Code **K** (Define Keywords).
2. Press ENTER.  
The Keyword Editor screen is displayed.

The Keyword Editor screen is shown below. Several keywords have already been defined to serve as examples for this section.

```

09:42:39                - SYSNCP Keyword Editor -                2000-05-04
Modify Keywords          Name SAGTEST   Library SYSNCP   DBID 10   FNR 32

I Line E Use  Keyword          IKN   ML Comment
-----
   1 1 Acti MENU                1004   1
   2 1 Acti DISPLAY            1002   2
   3 S Syno SHOW                1002   1
   4 1 Acti DELETE             1001   2
   5 S Syno PURGE              1001   1
   6 S Syno ERASE              1001   1
   7 1 Acti FILE                1003   4
   8 P Parm NAME                4002   2
   9 2 Obje FILE                2001   4
  10 P Parm NUMBER             4001   2
  11 2 Obje DOCUMENT           2003   2
  12 1 Acti INFORMATION        1005   1
  13
  14
----- All -----

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      Help  Cmd  Exit  Last  List  Flip  -1   +1   Top  Bot  Info  Canc
    
```

Enter in the Keyword Editor all the keywords which you want to have in your command language. These can be entered in any order desired, except synonyms, which must immediately follow the keywords they are related to. To each keyword you assign a type which specifies to which part of command syntax the keyword belongs. Rules of command syntax for a command processor are specified in the processor header; see Keyword Runtime Options - Header 1 in the section Header Records. For example, you can specify whether a keyword is to be of type 1 (entered in first position in a command), type 2, type 3, a synonym for another keyword or a parameter indicator.

**Note:**

A command language requires a strict syntax because, to date, no computer is capable of understanding semantics. Word type is, therefore, the only practical way to communicate meaning in a command language.

In the example above, the keywords DELETE and DISPLAY are defined as keywords of type 1. As specified in the processor header, these keywords denote actions. The keyword DOCUMENT is defined as a keyword of type 2 and it denotes an object. The keyword FILE, however, is defined as both type 1 and 2, and it can, therefore, denote an action or an object, depending on where it is positioned in the command. It is possible to compose the two keyword types to make commands, such as DELETE FILE and FILE DOCUMENT.

You can save the keywords you have entered by issuing the SAVE or STOW command from the command line. In addition to saving the keyword definitions in source form, the STOW command performs a consistency check on them. Once a keyword is stowed successfully, it is given an internal keyword number (IKN) which is used at runtime to evaluate a command. Synonyms are always linked to a master keyword and always take the IKN of their master.

Each line in the Keyword Editor contains the following fields:

Field	Explanation
I	<p>Output field. An information field which can contain the following values:</p> <ul style="list-style-type: none"> <li>E Indicates that a definition error has been detected.</li> <li>X Line is marked with X.</li> <li>Y Line is marked with Y.</li> <li>Z Line is marked with both X and Y.</li> <li>S Scan value found in this line.</li> </ul>
Line	Output field. The line number of the editor.
E	<p>Specifies the entry type for a keyword; that is, the position the keyword is to be entered in a command: first, second or third position, synonym or parameter indicator.</p> <p>For instance, in the Keyword Editor screen example above the keyword DELETE is of entry type 1 and DOCUMENT of type 2. Using these keywords, the command DELETE DOCUMENT can be defined.</p> <p>The field takes any of the following characters as input:</p> <ul style="list-style-type: none"> <li>1 The keyword defined in this line is to be used as the first entry in a command sequence.</li> <li>2 The keyword defined in this line is to be used as the second entry in a command sequence.</li> <li>3 The keyword defined in this line is to be used as the third entry in a command sequence.</li> <li>S The keyword defined in this line is to be used as a synonym for the preceding keyword with entry type 1, 2, 3 or P.</li> <li>P The keyword defined in this line is to be used as a parameter indicator in a command sequence.</li> <li>* No keyword is to be defined in this line. Instead, the line is to be used solely as a comment line.</li> <li>? This symbol is an output value which indicates an invalid keyword specification.</li> </ul>
Use	<p>Output field. The value displayed is determined by the value entered in the preceding field E:</p> <ul style="list-style-type: none"> <li>1 - 3 The first four characters of the user text specified in the processor header for the first, second and third keyword entries respectively are displayed. See also Keyword Editor Options - Header 2 in the section Header Records.</li> <li>S SYNO, the abbreviation for synonym, is displayed.</li> <li>P PARM, the abbreviation for parameter indicator, is displayed.</li> </ul>
Keyword	<p>Enter the keyword to be defined. Embedded blanks are not permitted. If you have specified in the processor header that keywords can only be upper case, then keywords are always translated to upper case, regardless of how they are entered. Otherwise, the case remains as entered. The maximum and minimum length of keywords depends on the settings specified in the header (default: 1 - 16 characters). Keywords must be unique unless specified otherwise in the header. Keyword prefixes can be used as described in Keyword Editor Options - Header 2 in the section Header Records.</p>
IKN	<p>Output field. The Internal Keyword Number (IKN) is an identifier assigned to each valid keyword. IKNs are useful for testing and debugging. They are allocated only when a keyword is successfully stowed (see also the STOW command under Editor Commands). Each keyword is assigned a unique IKN, except synonyms, which take the IKN of their master term (see the Keyword Editor screen example above: DISPLAY and SHOW).</p>

Field	Explanation
ML	<p data-bbox="320 344 1396 533">Input and output field indicating the minimum length of a keyword. The field is an input field if <b>S</b> is specified in the Dynamic Length Adjustment field of the processor header as described in "Keyword Runtime Options - Header 1", Header Records. In this case, you must specify the number of characters which must be entered for the keyword. For all other input, this field contains the minimum number of characters of a keyword a user must specify to avoid ambiguity with other keywords.</p> <p data-bbox="320 564 1385 622">For instance, in the Keyword Editor screen example above, keyword MENU requires only input of <b>M</b> while keyword DISPLAY requires input of <b>DI</b> to avoid ambiguity with keyword DELETE.</p>
Comment	<p data-bbox="320 642 1385 828">Enter free text for a keyword. There are no input restrictions. The user text is included in the cataloged command processor if the field Catalog User Texts is set to Y in the header definition as described in "Miscellaneous Options - Header 3", Header Records. It can be read at runtime using the TEXT option of the PROCESS COMMAND statement. The header text appearing at the top of this column is controlled by the header definition fields "Header for User Text 1" and "Header for User Text 2".</p>

## Editor Commands

In the command line of the Keyword Editor, you can enter the following commands:

Command	Function
ADD	Adds ten empty lines to the end of the editor.
CANCEL	Returns to Processor Maintenance Menu.
CHECK	Tests the keyword source for consistency.
EXIT	Returns to Processor Maintenance Menu.
HELP	Displays valid escape characters and other useful processor settings.
INFO	Displays information on the keyword on which your cursor is positioned.
LET	Undoes all modifications made to the current screen since the last time ENTER was pressed.
POINT	Positions the line in which a line command .N is entered to the top of the current screen.
RECOVER	Returns keyword source that existed before last SAVE/STOW.
RESET	Deletes the current X and Y line markers.
SAVE	Keyword source is saved.
SCAN	Scans for the next occurrence of the scan value.
STOW	Keyword source is stowed and Internal Keyword Numbers (IKNs) are generated for valid keywords.

## Positioning Commands

Editor positioning commands are the same as the ones provided for the Natural program editor. For more information, see the description of the program editor in the Natural Editors documentation.

The last line of the editor contains an output field which informs you of where your display is located in the editor. The following output values are displayed:

Top	Editor is currently positioned at the top of the keyword source.
Mid	Editor is currently positioned at the center of the keyword source.
Bot	Editor is currently positioned at the bottom of the keyword source.
Emp	Editor is currently empty.
All	The entire source is contained on the current screen.

## Line Commands

Line commands in the Keyword Editor are the same as in the Natural program editor with the exception of the commands .J and .S, which cannot be used.

Each command is entered beginning in the **E** field; the remaining part of the command is entered in the Keyword field, as illustrated in the screen below:

09:42:39		- SYSNCP Keyword Editor -				2000-05-04	
Modify Keywords		Name	SAGTEST	Library	SYSNCP	DBID	10 FNR 32
I	Line	E	Use	Keyword	IKN	ML	Comment
-	-	-	-	-	-	-	-
	1	1	Acti	MENU	1004	1	
	2	1	Acti	DISPLAY	1002	2	
	3	S	Syno	SHOW	1002	1	
	4	.	Acti	<b>i(3)</b> TE	1001	2	
	5	S	Syno	PURGE	1001	1	

### Attention:

When you move (.M) or copy (.C) lines, ensure that individual keywords are always moved or copied together with their synonyms.

When you delete (.D) lines, the corresponding keywords and any functions containing these keywords will not be deleted from the database until you issue the STOW editor command. As long as you do not issue the STOW command, these functions will still be displayed within the Function Editor.

# SYSNCP - Function Maintenance

Functions are composed of the keywords entered in the Keyword Editor. Before it is possible to define functions, the keywords must be successfully stowed (see the section Keyword Maintenance).

- Define Functions
- Editor Commands
- Direct Command QUICK-EDIT
- Local and Global Functions
- Procedure for Validating Functions

## Define Functions

Use the Define Functions function and the Function Editor to specify functions and compose valid commands which can be accessed from a specific location.

 **To invoke the Function Editor**

1. On the Processor Source Maintenance menu, enter Function Code **F** (Define Functions).
2. Press ENTER.  
The Function Editor screen is displayed.

The Function Editor displays all possible combinations of the keywords stowed in the Keyword Editor.

The screen below, shows the Function Editor with keywords used as examples in the Keyword Editor screen in the section Keyword Maintenance:

```

09:45:53          ***** NATURAL SYSNCP UTILITY *****          2000-05-04
User SAG          - Function Editor -
Edit Global Combinations   Name SAGTEST  Library SYSNCP   DBID 10   FNR 32

Global
I Ac  Action          Object          Addition          Global Local Any Loc
-----
      DELETE          DOCUMENT          Yes
      DELETE          FILE             Yes
      DISPLAY         DOCUMENT          Yes
      DISPLAY         FILE             Yes
      FILE            DOCUMENT          Yes
      FILE            FILE             Yes
      INFORMATION    Yes
      INFORMATION    DOCUMENT
      INFORMATION    FILE

Repos: _____

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Cmd  Exit  Last  List  Flip      +      Top   Loc  Loc+  Canc
    
```

You have to validate each keyword combination that you want to designate as a valid function in your application. A keyword combination can be validated as a global function, local function or both. A global function can be invoked from anywhere in an application, whereas a local function can only be invoked from a

specific location within an application.

Two fields in the upper left corner of this screen indicate the current validation mode (local or global) and the location for which keyword combinations can currently be validated. In the screen above, the text "Edit Global Combinations" indicates that global mode is active. If the local mode were active, the text "Edit Local Combinations" would appear here. In the screen above, the text "Global" appears below this text. This indicates that global validation can be performed for all of the combinations listed. In local mode, in this field the name of the location appears for which local validation can be performed (for example, "Local DISPLAY FILE").

The Function Editor contains the following columns:

Column	Explanation
I	Output field. The following values are output as a result of function editing.  E Runtime action edited. D Referenced locations displayed. V Validation issued. R Validation removed.
Ac	Action to be taken. The following values can be entered:  VG Validate as global function. VL Validate as local function. RG Remove validation as global function. RL Remove validation as local function. DL Display all functions which reference the specified function as a local function. EG Invoke the Runtime Action Editor for a global function (see Runtime Action Editor in the section Runtime Actions). EL Invoke the Runtime Action Editor for a local function (see Runtime Action Editor in the section Runtime Actions). +G Invoke global mode, so that you can maintain any global functions. +L Invoke local mode for the current line, so that you can maintain local functions for this line. IN Information about keywords in this line.
Action	These three columns are used to display all possible combinations of currently defined keywords.
Object	The text which appears at the top of each keyword column is controlled by the fields "First Entry used as", "Second Entry used as" and "Third Entry used as" as specified in the processor header (see
Addition	Keyword Runtime Options - Header 1 in the section Header Records).
Global	If the function has been defined as a global command, Yes appears in this field.
Local	If the function has been defined as a local command, Yes appears in this field for the current location (only displayed in local mode).
Any Loc	Any Location. If the function has been defined as a local command anywhere else within the processor, Yes appears in this field for any other location.

## Editor Commands

In the command line of the Function Editor, you can enter the following commands:

Command	Function
ANY ON	Enable the column Any Loc.
ANY OFF	Disable the column Any Loc (the column will be filled with question marks). This allows for faster scrolling in the Function Editor. Moreover, the third repositioning field is available. Also, processing-in-progress information windows will not be displayed.
FIELD	Display keyword-specific combinations.
GLOBAL	Activate global mode.
LOC	Position to next location group.
LOC+	Position forward by one location.
SINGLE ON	Display only single-word functions.
SINGLE OFF	Display all possible combinations.
TOP	Position to top of list.

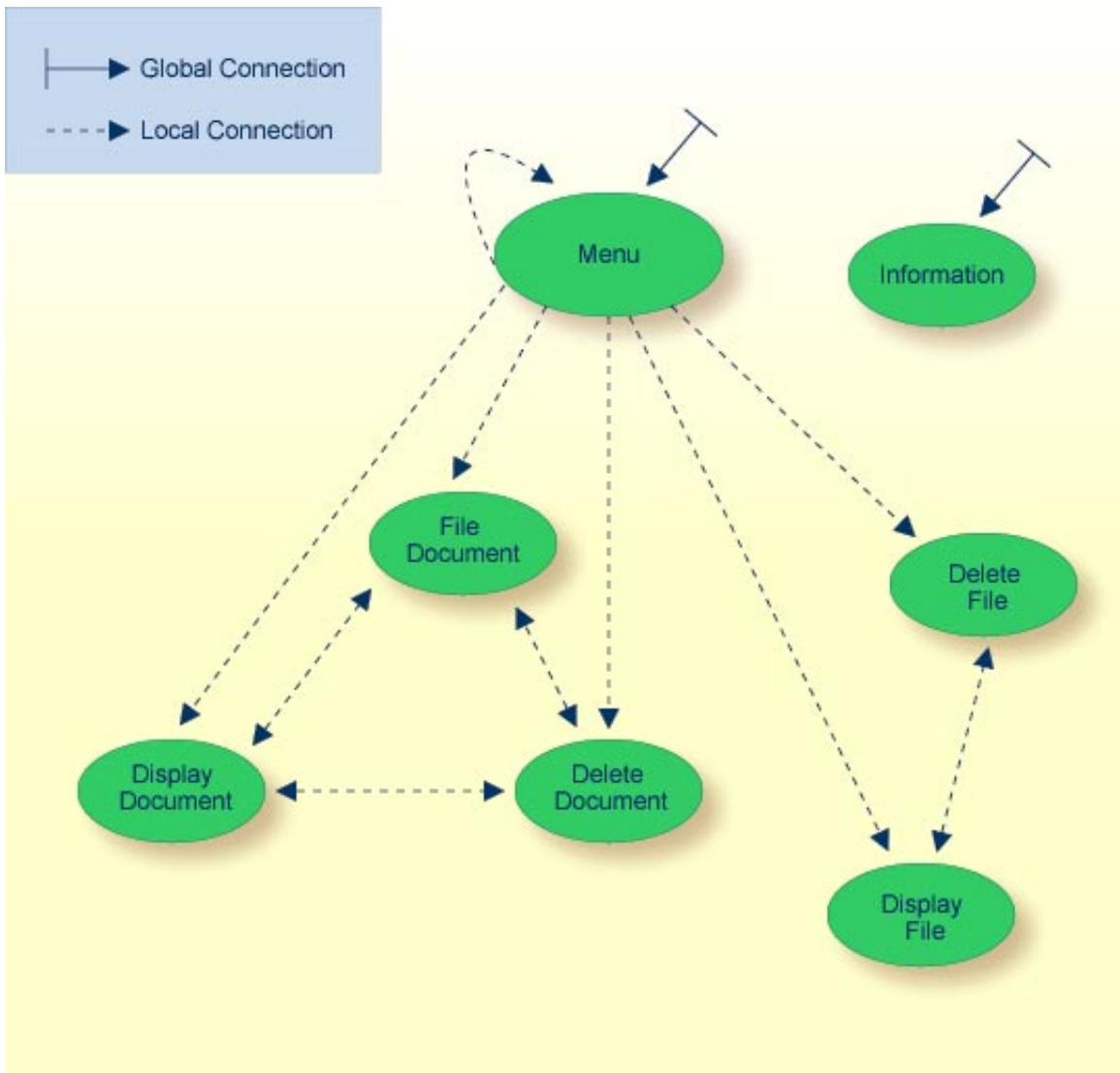
## Direct Command QUICK-EDIT

The direct command QUICK-EDIT enables you to quickly define local/global functions, as well as the corresponding runtime actions, by entering keywords or IKNs directly. This may be helpful for extremely large command processors. Note, however, that the location from which the command can be issued is not verified and navigation may not function correctly at runtime.

## Local and Global Functions

To understand the concept of local and global functions, you have to picture each valid keyword combination as a location in your application (for example, a location called Display File). In the Function Editor, you specify the commands which can be issued from this location, as well as from which locations this location can be reached using the command DISPLAY FILE.

Local and Global Connections within a Sample Application:



In the sample application above, the Menu and Information locations are the only locations which have been designated as global. Thus, they can be accessed directly from all of the remaining locations in the application. All locations have been designated as local to the location Menu, except Information. The only way to get from the location Display File to Display Document is via Menu.

## Procedure for Validating Functions

The Function Editor operates in two modes: global and local. From global mode you can validate global functions and from local mode you can validate global and local functions. Global mode is the default mode. You can determine whether the editor is in global or local mode by the output field above the **I** field in the editor. If the editor is in global mode, then Global is displayed. If the editor is in local mode, then the location for which local functions are to be validated is displayed. Below is a general procedure for validating global and local functions for an application.

### To validate global and local functions

1. With the Function Editor in global mode, enter **VG** (validate global) in the Ac field next to the corresponding action to validate all global functions.  
Press ENTER.  
The Runtime Action Definition screen appears.
2. Press PF3 to return to the Function Editor.  
Yes appears under the column heading Global beside the validated functions.
3. Enter **+L** in the Ac field for each global function validated in the previous step, to switch to local mode.  
Press ENTER.
4. Enter **VL** (validate local) in the Ac field for each function that is to serve as a location for this global function.  
Press ENTER.  
The Runtime Action Definition screen appears.
5. Press PF3 to return to the Function Editor.  
Yes appears under the column heading Local beside the validated functions.
6. To validate local functions for a **local** location:  
Enter **+L** (invoke local mode) in the Ac field for each location validated in the previous step, to validate all local functions which are to be used from this location.  
Press ENTER.
7. Enter **VL** (validate local) in the Ac field for each function that is to serve as a local function for the current location.
8. Press PF3 to return to the Function Editor.  
Yes appears under the column heading Local beside the validated functions.

#### **Note:**

If in the command processor header (Processor Header Maintenance 3) the field Invoke Action Editor is set to Y, in addition, the window Runtime Action Definition (see Runtime Action Editor in the section Runtime Actions) is displayed for each action.

# SYSNCP - Runtime Actions

Once valid keyword combinations have been identified as either local or global functions in the Function Editor, it is possible to link each function with one or more runtime actions. Runtime actions consist of one or more steps which are to be carried out whenever a function is issued.

Below is information on:

- Define Runtime Actions
- Runtime Action Editor

## Define Runtime Actions

There are two different locations in SYSNCP from which you can define runtime actions: the Function Editor (see the section Function Maintenance) and the Result Editor. The Result Editor is explained in this section, including how to specify runtime actions for a function.

 **To invoke the Result Editor**

1. On the Processor Source Maintenance menu, enter Function Code **R** (Define Runtime Actions).
2. Press ENTER.

The Result Editor screen is displayed:

```

09:47:03          ***** NATURAL SYSNCP UTILITY *****          2000-05-04
User SAG          - Result Editor -
List defined combinations   Name SAGTEST   Library SYSNCP   DBID 10   FNR 32

I Ac Location          Command          Result
-----
  < Global >          MENU          KR
  < Global >          INFORMATION   SF
  DELETE FILE          DISPLAY FILE   SF
  DELETE DOCUMENT      DISPLAY DOCUMENT SF
  DISPLAY FILE          DELETE FILE    SF
  DISPLAY DOCUMENT      DELETE DOCUMENT SF
  DISPLAY DOCUMENT      FILE DOCUMENT  SF
  FILE DOCUMENT          DELETE DOCUMENT SF
  FILE DOCUMENT          DISPLAY DOCUMENT SF
  MENU          DELETE FILE    KCS
  MENU          DELETE DOCUMENT KCCS
  MENU          DISPLAY FILE   KRCS
Repo _____

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Cmd  Exit Last List Flip      +      Top  Loc-- Loc+ Canc
    
```

The Result Editor contains all of the local and global functions specified in the Function Editor. Each line in the editor represents the location from which a command can be issued (Location field), the command itself (Command field) and an abbreviated summary of the action to be carried out when the command is issued (Result field).

The fields of the screen are explained in detail in the table below:

Field	Explanation
I	Output field. Information on the last action carried out on this line.
Ac	Action to be taken. The following values can be entered:  DI Display the runtime action definitions for this function. ED Edit the runtime action definitions for this function. PU Purge this function.
Location	Output field. The location within the application from which the command (see Command field below) can be issued. If the function is global, then < Global> appears in this field (the command can be issued from any location).
Command	Output field. The command.  The contents of the Location and Command fields may be truncated if very long keywords are used.
Result	Output field. Contains an abbreviated summary of the action to be performed when the command is issued. The first character represents the Keep Location information (see the following section); for all other characters, see the Runtime Action Definition table below.

## Runtime Action Editor

The Runtime Action Editor is used to define the actions to be taken when a command is issued from a specific location. The editor can only be invoked for functions which have been defined as global or local functions. The editor can be invoked either from the Function Editor or the Result Editor.

### ▶ To invoke the Runtime Action Editor from the Function Editor

1. In the Ac field, enter **EG** (edit global) for global functions.  
Or  
In the Ac field, enter **EL** (edit local) for local functions.
2. Press ENTER.

### ▶ To invoke the Runtime Action Editor from the Result Editor

1. In the Ac field, enter **ED**.
2. Press ENTER.  
The Runtime Action Definition window is displayed:

Runtime Action Definition

Location .... DISPLAY DOCUMENT  
Command ..... DELETE DOCUMENT

Keep Location .... S  
Data allowed ..... Y    More than one .... N    Max. Length ..... 99  
Numeric ..... N    TOP of STACK ..... Y    Error: Drop ..... Y

A Runtime Action Definition  
- -----  
F DE-PGM \_\_\_\_\_  
- \_\_\_\_\_  
- \_\_\_\_\_  
- \_\_\_\_\_  
- \_\_\_\_\_  
- \_\_\_\_\_  
- \_\_\_\_\_

Actions are always associated with an origin and a destination. The origin is the location from which the command is issued, and the destination is the command itself. Thus, it is possible to link different actions to a command based on the context in which it is used.

In the Runtime Action Editor, you also specify whether the location is to remain the same after the actions have been carried out, or whether the command itself is to become the new current location.

Actions are specified by entering a single-letter code in the left column of the editor. Enter any parameters accompanying an action in the field next to the code. If the characters "/"\* are entered in this field, all subsequent input is considered a comment. If you omit a required parameter, you will be prompted for input.

The sequence in which actions are performed at runtime is determined by the order of entry in the editor (from top to bottom). Thus, if a FETCH is specified, all of the actions specified below it are not to be performed.

The Runtime Action Editor contains the following fields:

Field	Explanation
Location	Output field. The location from which the command is issued. If the function is defined as global, the field shows < Global>.
Command	Output field. Command for which actions are to be specified.
Keep Location	<p>Specifies whether the current or a new location is to be active once the actions have been performed. A value in this field only affects commands with a specified EXEC option. Possible values are:</p> <p>K Keep current location. The actions to be performed affect the current location only.</p> <p>S Set new location (global/local). Once the actions are performed, the command processor makes the command the new current location. Every command entered subsequently has to be either a local command of this new location or a global command.</p> <p><b>Note:</b> The defined actions themselves have no influence on the location; that is, any action performed does <b>not</b> cause the current location to be changed.</p>
Other Options	<p>All other options are related to the handling of parameters provided with this command sequence. For further information, see Command Data Handling - Header 4 in the section Header Records.</p> <p>To activate the header defaults of these options, enter an asterisk (*).</p>

 **To define runtime actions**

1. Invoke the Runtime Action Definition window as described earlier.
2. In the field **A**, enter an action code and the corresponding action in the field opposite to it:

<b>Code</b>	<b>Runtime Action Definition</b>
V	Default value. No runtime action is specified.
T	Text which can be read at runtime using the TEXT or GET option of the PROCESS COMMAND statement.
M	Modify command line. The data are placed in the command line.
C	Command. This command is placed at the top of the Natural stack. If an asterisk (*) is specified here, the name of the program which issued this PROCESS COMMAND statement is put on top of the stack (STACK TOP COMMAND '*PROGRAM'). (*)
D	Data. These data are placed on top of the Natural stack. (*)
F	Natural program name. The program is invoked with a FETCH statement. (*)
S	Natural STOP statement. The statement is executed at runtime. (*)
E	The value specified in this line is to be moved immediately into the system variable *ERROR-NR.
R	A return code is entered in the DDM field RETURN-CODE as described in PROCESS COMMAND in the Natural Statements documentation.
1 to 9	A text string. This value is entered into the multiple DDM field RESULT-FIELD as described in PROCESS COMMAND in the Natural Statements documentation.
*	Comment line.

\* These actions are only performed with the EXEC option of the PROCESS COMMAND statement.

3. Press PF3 to leave the Runtime Action Definition window.

**Note:**

The user exit NCP-REAM allows you to use some or all of the above codes. The user exit NCP-REEM allows you to modify the line that follows the heading of the Runtime Action Definition table. The user exit NCP-REDM allows you to define default values for runtime action definitions (if you use this user exit, see also Invoke Action Editor in the section Header Records). All user exits mentioned above are delivered in the Natural system library SYSNCP

# SYSNCP - Processor Cataloging

Once you have specified runtime actions for all of the functions you want to use in your command processor, you should catalog the command processor. Cataloging a command processor generates a Natural object of type Processor.

## To catalog a command processor

1. On the Processor Maintenance menu, enter Function Code **C** (Catalog Processor), the name of the command processor to be cataloged, and the name of the Natural library in which the command processor is to be cataloged.
2. Press ENTER.

### **Note:**

If you have Natural Security installed, you have to allow the use of your command processor as described in the Natural Security documentation in the section Functional Security.

### **Note for Windows and UNIX:**

Unlike on mainframes, SYSNCP does not create a report when cataloging a command processor.

# SYSNCP - Administrator Services

SYSNCP provides facilities for the administration of command processors. Only system administrators, as defined in Natural Security, are authorized to access these services.

## ▶ To access the administrative services

1. On the Processor Source Maintenance menu, enter Function Code A (Administrator Services).
2. Press ENTER.

The Administrator Services screen is displayed:

```

09:49:11          ***** NATURAL SYSNCP UTILITY *****          2000-05-04
User SAG          - Administrator Services -

                Code  Function
                S    Select Processor
                C    Copy Processor Source
                D    Delete Processor Source
                P    Print Source/Object/NCP-Buffer
                U    Unload Processor to Work File 3
                L    Load Processor from Work File 3
                F    Freeze Processor Source
                R    References from Natural Security
                ?    Help
                .    Exit

                Code .. _      Name .. SAGTEST_  Library .. SYSNCP__

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Cmd  Exit Last List Flip                               Canc

```

### Note:

If you do not have Natural Security installed, be aware that all other users have administrator status.

Below is information on:

- Select Processor
- Copy Processor Source
- Delete Processor Source
- Print Source/Object/NCP Buffer
- Unload Processor
- Load Processor
- Freeze Processor Source
- References from Natural Security

## Select Processor

See the section Processor Selection.

## Copy Processor Source

In copying processor sources, you have the choice of copying the entire processor or only selected sources (header, keywords, functions, runtime action definitions).

**▶ To copy a command processor**

1. On the Administrator Services menu, enter Function Code **C**.
2. Press ENTER.

The Copy Processor Source window is displayed to provide source and target information:

Copy Processor Source		
	Source	Target
Name .....	SAGTEST_	_____
Library .....	SYSNCP__	SYSNCP__
DBID .....	10__	10__
FNR .....	32__	32__
Password .....		
Cipher Key ..		
Replace .....	NO_	

3. In the Source fields, enter the name of the processor to be copied, and the library, database ID (DBID) and file number (FNR) in which the processor is stored.  
 The default values correspond to the processor specified on the Administrator Services menu.  
 In the Target fields, enter the name of the processor to be copied to, and the library, database ID (DBID) and file number (FNR) into which the processor is to be copied.  
 In the Cipher Key field, enter the appropriate password and/or cipher key if the source and/or target file is protected by a password and/or cipher key.  
 In the Replace field, enter YES if you want to overwrite a processor in the target environment.  
 The default for this field is NO.

4. Press ENTER.

The following window is displayed to select sources:

Copy Processor Source					
Mark	Copy		Source		Target
---	-----		---		-----
-	Header .....		yes		no
-	Keywords .....		yes		no
-	Functions .....		yes		no
	Runtime Action Definitions ..		no		no
Source Name	SAGTEST	Library	SYSNCP	DBID 10	FNR 32
Target Name	TEST2	Library	SYSNCP	DBID 10	FNR 32
Replace ...	NO				

5. In the appropriate Mark fields, enter any character to select the sources you want to copy.

6. Press ENTER.

# Delete Processor Source

This function is used to delete processor sources.

**▶ To delete a command processor**

1. On the Administrator Services menu, enter Function Code **D**.
2. Press ENTER.  
The Delete Processor Source window is displayed.
3. Specify the name of the processor to be deleted, and the library, database ID and file number in which the processor is stored. If the file is protected by a password and/or cipher key, you also have to enter the appropriate password and/or cipher key.
4. Press ENTER.  
The following window is displayed to select the sources to be deleted:

Delete Processor Source					
Mark	Delete				Available
----	-----				-----
-	Header .....				yes
-	Keywords .....				yes
-	Functions .....				yes
-	Runtime Action Definitions ..				yes
Name SAGTEST    Library SYSNCP    DBID 10    FNR 32					

To the right of each processor source (header, keywords, functions, runtime action definitions) is a field which indicates whether the source exists. As command processor creation is a cumulative activity, you cannot delete a source without deleting all sources which are based on it. Thus, for example, in the screen above, you cannot delete the source of the functions without also deleting the source of the runtime action definitions.

5. In the appropriate Mark fields, enter any character to select each source indicated as Available.
6. Press ENTER.

## Print Source/Object/NCP Buffer

In addition to processor sources, you can also print the processor object and the NCP.

**▶ To print a command processor item**

1. On the Administrator Services menu, enter Function Code **P**.
2. Press ENTER.  
The "Print Source/Object/NCP-Buffer" window is displayed.
3. Specify the name of the processor to be printed, and the library, database ID and file number in which the processor is stored. If the file is protected by a password and/or cipher key, you also have to enter the appropriate password and/or cipher key.
4. Press ENTER.
5. The following window is displayed to select items for printing:

Mark	Print	Available
----	-----	-----
-	Header .....	yes
-	Keywords .....	yes
-	Functions .....	yes
-	Runtime Action Definitions ..	yes
-	Processor Object .....	yes
	NCP-Buffer .....	no
	Printer .....	_____
Name SAGTEST    Library SYSNCP    DBID 10    FNR 32		

To the right of each processor source (header, keywords, functions, runtime action definitions) is a field which indicates whether the item exists.

Possible input values for the Printer field are the logical printer ID, VIDEO or SOURCE; see also DEFINE PRINTER in the Natural Statements documentation.

6. In the appropriate Mark fields, enter any character to select the items you want to have printed and enter the logical printer name or the value VIDEO or SOURCE in the Printer field.
7. Press ENTER.

# Unload Processor

**▶ To unload a command processor**

1. On the Administrator Services menu, enter Function Code U.
2. Press ENTER.

The "Unload Processor to Work File 3" window is displayed:

Unload Processor to Work File 3		
	Source	Target
Name .....	SAGTEST_	
Library .....	SYSNCP__	SYSNCP__
DBID .....	10__	
FNR .....	32__	
Password .....		
Cipher Key ..		
Report .....	NO_	

3. In the Source fields, enter the name of the processor to be unloaded, the library, database ID, and file number in which the processor can be found; the default value is the processor specified in the Administrator Services menu. Enter the appropriate password and/or cipher key if the file is protected by a password and/or cipher key.
4. In the Report field, enter YES if you want a report to be produced. Default is NO.  
You do not have to use a file extension. If you wish to use an extension, you must use the file extension ".sag".
5. Press ENTER.

When the processor is unloaded, all processor sources (header, keywords, functions, runtime action definitions) are written to Work File 3.

**Note:**

To transfer command processors from one hardware platform to another, use the utility SYSTRANS (mainframes) or the Natural Object Handler (Windows and UNIX).

## Load Processor

### ▶ To load a command processor

1. On the Administrator Services menu, enter Function Code **L**.
2. Press ENTER.

The "Load Processor from Work File 3" window is displayed for loading processors from Work File 3 to a Natural library:

```
Load Processor from Work File 3

Replace existing processors .. N
Produce load report ..... NO_
```

3. In the "Replace existing processors" field, enter **Y** or **N** (default is **N**) to specify whether existing processors with the same name are to be replaced by the processor to be loaded.
4. In the "Produce load report" field, enter **YES** (default is **NO**) if you want a report to be produced.
5. Press ENTER.

**Note:**

Input for the processor name and the library into which the processor is to be loaded is taken from the work file.

## Freeze Processor Source

You can freeze a processor in its current state to prevent users from modifying it further.

### To freeze a command processor

1. On the Administrator Services menu, enter Function Code **F**.
2. Press ENTER.  
The Freeze Processor Source window is displayed.
3. Specify the name of the processor to be frozen, and the library, database ID and file number in which the processor is stored. If the file is protected by a password and/or cipher key, you also have to enter the appropriate password and/or cipher key.
4. Press ENTER.
5. In the following window, specify with **Y** or **N** whether modification of the processor sources is to be allowed or not. Default is **Y**.
6. Press ENTER.

## References from Natural Security

This function is only available if Natural Security is active in your environment. It is used to delete functional security references from Natural Security.

If functional security is defined for a processor in Natural Security, references are created automatically. These references are stored in the FNAT/FUSER system files along with the processor sources, not in FSEC.

### To invoke References from Natural Security function

1. On the Administrator Services menu, enter Function Code **R**.
2. Press ENTER.  
The Delete References window appears.
3. Specify the name of the processor, and the library, database ID and file number in which the processor is stored. If the file is protected by a password and/or cipher key, you also have to enter the appropriate password and/or cipher key.
4. Press ENTER.
5. In the following window, you can delete main references, function references and auxiliary references.

For further information on functional security for command processors, refer to the section Functional Security in the Natural Security documentation.

# SYSNCP - Session Profile

A session profile is a collection of user-definable defaults which determine how the SYSNCP screens appear or how SYSNCP reacts to input. In a session profile, for example, you can determine which command processor you want as default for a session or which colors you want assigned to screen attributes. In SYSNCP, there is a standard session profile called STANDARD which is issued to all new users. You can create several different session profiles and activate them as required.

Administrators for SYSNCP can access and modify any session profile in SYSNCP. Other users can access all session profiles, but can modify only those session profiles which are created under their user ID or which have the same name as their user ID.

## To define or modify a session profile

- Issue the PROFILE command from the command line of the Processor Source Maintenance menu. The first of three session profile maintenance screens is displayed.

Below is information on:

- Session Profile Name
  - Session Parameters - Profile 1
  - Color Attributes - Profile 2
  - Miscellaneous Attributes - Profile 3
- 

## Session Profile Name

The standard profile STANDARD or the value of the system variable \*USER is taken as default for the profile name.

If you are defining a new session profile, the parameters/attributes are defaults. You can modify these defaults as required and save them by entering the new name and pressing PF5.

The field Session Profile Name on each profile screen is both an input and output field. Thus, it is possible to define, read or save another profile from any of these screens by entering its name in the Profile Name field and pressing PF5 or PF4, respectively.

## Session Parameters - Profile 1

On the first profile maintenance screen, you can modify the following fields:

Field	Explanation
Apply Terminal Control 1	These fields can be used to enter the parameters of a SET CONTROL statement to be issued by SYSNCP at startup.
Apply Terminal Control 2	For example, when you enter <b>Z</b> in any of the fields, SYSNCP issues the statement SET CONTROL 'Z'.
Default Processor Name	The default command processor name to be used for this session.
Default Processor Library	The Natural library to be used to store a command processor.
Cancel Reaction	Specifies whether a warning is to be issued whenever the requested modification is not completed and the CANCEL command is issued.  W Issue warning. B Back out and cancel without issuing warning.
Clear Key Allowed	Specifies whether clear key is allowed.  N Clear key disallowed. Y Clear key active and has same effect as CANCEL.
Default Cursor Position	Specifies placement of the cursor.  1 Cursor to be positioned in first field of the screen. C Cursor to be positioned in command line.
Exec/Display Last Command	Specifies action to be taken as a result of the LAST command:  E Execute last command issued in command line. D Display last command issued in command line.

## Color Attributes - Profile 2

On the second profile maintenance screen, you can assign colors to various screen attributes, or overwrite existing color assignments.

By specifying the following color codes, you can assign the following colors:

Code	color
BL	Blue
GR	Green
NE	Neutral
PI	Pink
RE	Red
TU	Turquoise
YE	Yellow

For color assignments to screen attributes, see also the terminal command %= in the Natural Terminal Commands documentation.

## Miscellaneous Attributes - Profile 3

The following attributes can be specified on the third profile maintenance screen:

Field	Explanation
Message Line Position	The line on which messages are to be displayed. The value 21 is recommended. See also the terminal command %M in the Natural Terminal Commands documentation for more information.
Text for PF5 Key	The PF5 function key is reserved for global (session-wide) use. The text to be displayed on the PF-key line for PF5 can be entered in this field.
Command for PF5 Key	The PF5 function key is reserved for global (session-wide) use. The command to be executed when PF5 is pressed can be entered in this field.

In addition, the screen displays when and by which user this profile was last modified.

# SYSPARM Utility

The utility SYSPARM is used for creating and maintaining a set of Natural profile parameters that is stored under a profile name.

When invoking Natural with dynamic profile parameters, you can specify individual parameters each time you invoke Natural. More comfortably, however, you can specify a set of parameters once in SYSPARM, store this set under a profile name, and then invoke Natural with only one dynamic parameter: `PROFILE=profile-name`. The parameters defined in this profile are then passed to Natural as dynamic parameters and must therefore comply with the syntax of PROFILE as described in the Natural Parameter Reference documentation.

For descriptions of the individual profile parameters that can be specified in a profile, refer to Profile Parameters in the Natural Parameter Reference documentation.

The profiles are stored in the Natural system file FNAT or FUSER. The database ID (DBID) and file number (FNR) of the current FNAT system file are shown on the SYSPARM Menu.

To restrict the use of a profile to specific users, you use the profile parameter USER (as described in the Natural Parameter Reference documentation).

The SYSPARM utility documentation covers the following topics:

- Invoking SYSPARM
  - List Profiles
  - Display Profile
  - Add New Profile
  - Modify Profile
  - Editing Profiles
  - Copy Profile
  - Delete Profile
  - Direct Commands and Batch Processing
- 

## Invoking SYSPARM

### To invoke the SYSPARM utility

- In the direct command line, enter SYSPARM.  
The Natural SYSPARM Utility Menu is displayed.

The SYSPARM Menu provides the following fields and functions:

Field/Function	Explanation
List Profiles	Displays a list of all profiles. From the list, you can select a profile for display, modification or deletion. See: List Profiles.
Display Profile	Displays a specific profile. See: Display Profile
Add New Profile	Creates a new profile. See: Add New Profile and Editing Profiles.
Modify Profile	Changes an existing profile. See: Modify Profile and Editing Profiles.
Copy Profile	Creates a new profile by copying an existing one. See: Copy Profile
Delete Profile	Deletes an existing profile. See: Delete Profile
Profile	Selects the profile specified. Enter a valid profile name or use asterisk (*) notation to select a range of profiles. Use asterisk (*) or a blank field to select all profiles available.
Copy to	Copies the profile specified.
DBID/FNR	Selects the database ID (DBID) and file number (FNR) of the Natural system file where the profile resides. Default is the current FNAT.
Password/Cipher	Specifies the password (8 characters) and the cipher code (8 digits) of the Adabas file.

The SYSPARM functions listed above are described in the remainder of this documentation.

## List Profiles

This function displays a list of all existing profiles.

### To invoke the List Profiles function

- On the SYSPARM Menu, enter Function Code **L** and, in the Profile field, enter the name of a profile or specify a range of names:
  - To list all profiles, enter an asterisk (\*) or leave the field blank.
  - To list all profiles with names that start with a specified value, use asterisk (\*) notation, for example, ABC\*
  - To list all profiles with names greater than or equal to a specified value, use the sign greater than (>), for example, ABC>.
  - To list all profiles with names with names less than a specified value, use the sign less than (<), for example, ABC<.

The List Profiles screen is displayed.

On the List Profiles screen, press PF7 or PF8 to scroll the list up or down.

Listed below are the PF keys and line commands available to select a profile for display, modification or deletion. Mark a profile with the cursor and press the appropriate PF key, or enter a line command in the Sel. column next to the profile and press ENTER:

Key	Line Command	Function
PF4	D	Invokes the Display Profile function for the profile specified.
PF5	M	Invokes the Modify Profile function for the profile specified.
PF6	X	Invokes the Delete Profile function for the profile specified.

## Display Profile

This function is used to display an existing profile.

### To invoke the Display Profile function

- On the SYSPARM Menu, enter Function Code **D** and the name of a profile.  
Or, from the List Profiles screen, select a profile as described in the relevant section.

The Display Profile screen appears displaying the profile specified.

To modify the profile, press PF5, which invokes the Modify Profile function (see below).

## Add New Profile

This function is used to create a new profile.

### To invoke the Add New Profile function

- On the SYSPARM Menu, enter Function Code **A** and the name of a profile.

On the screen displayed, enter the profile parameters to be included in the profile as described in Editing Profiles.

## Modify Profile

This function is used to change an existing profile.

### To invoke the Modify Profile function

- On the SYSPARM Menu, enter Function Code **M** and the name of a profile.  
Or, on the Display Profile screen, press PF5.  
Or, from the List Profiles screen, select a profile as described in the relevant section.

A screen (similar to the screen for adding new profiles) is displayed on which you can modify the profile selected: see Editing Profiles below.

## Editing Profiles

The screen provided for adding or modifying a profile contains 18 input lines with a maximum length of 72 characters each. The individual parameters entered must be separated from one another by (one or more) blanks or commas. You can spread the specification of the parameters over as many lines as you like.

If you need additional space for editing or want to combine different profiles or group profiles by categories, concatenate multiple profiles by entering the PROFILE parameter as the last entry in the profile(s) concerned. The first profile then evaluates the parameter strings specified in the second profile as part of the string of the first profile.

**Example:**

PROFILE=P2 entered at the end in Profile P1, and PROFILE=P3 entered at the end in Profile P2, firstly invokes P1 and then P2 followed by P3.

You can enter a commentary text in each editor line, or spread a comment over as many lines as you like. Comments must be preceded by /\* and followed by \*/.

Below is information on:

- Functions
- Help with Parameters

## Functions

The following functions are available on the screen provided for editing profiles:

Key	Function
PF4	Checks whether the parameter specifications within the profile are syntactically correct.
PF5	Stores the profile.
PF9	Inserts one blank line below the line containing the cursor.
PF10	Deletes the line containing the cursor.
PF11	Copies the line containing the cursor.

## Help with Parameters

 **To invoke help information on a specific profile parameter**

- In the field "Help with Parameters":
  - Enter the name of a parameter.
  - Or, enter a character string with asterisk (\*) notation.  
All parameters whose names begin with that character string are listed.  
Select a parameter by marking it with any character.
  - Or, enter an asterisk (\*).  
A list of all parameters is displayed.  
Select a parameter by marking it with any character.
- Press ENTER.  
Or, place the cursor at a certain editor line and press ENTER as described under Special Positioning below.

The screen that appears contains two sections:

- The upper section with help text on the parameter specified.  
If the text extends beyond the current screen, in the field More Help, enter a plus (+) sign to display the next screen.  
Enter a minus (-) sign to return to the previous screen or enter a period (.) to terminate the Help with Parameters function.
- The lower section "Enter your parameter specification".
- Under "Enter your parameter specification", enter a parameter specification.
- Press PF4 to have your parameter specifications checked.

- Press PF3 to terminate the Help with Parameters function and add the parameter specifications at the bottom of your profile or at the position marked earlier: see Special Positioning below.

### Special Positioning

To add new parameter specifications at a certain position in the profile:

- In the field "Help with Parameters", enter a value (or an asterisk) and do **not** press ENTER.
- Position the cursor at the editor line below which you want the new parameter specifications to be placed.
- Press ENTER to invoke the Help with Parameters function and enter a parameter specification.
- Press PF3.

The new parameter specification is added below the editor line marked earlier.

## Copy Profile

This function is used to create a new profile by copying an existing one.

With this function, you can copy profiles only within the same Natural system file. To copy a profile from one Natural system file to another, use the utility SYSMAIN as described in the relevant documentation.

### To invoke the Copy Profile function

- On the SYSPARM Menu:
  - Enter Function Code C.
  - In the field Profile, enter the name of an existing profile.
  - In the field Copy To, enter the name of the new profile.

## Delete Profile

This function is used to delete an existing profile.

### To invoke the Delete Profile function

- On the SYSPARM Menu, enter Function Code X.  
Or, from the List Profiles screen, select a profile as described in the relevant section.

The Delete a Profile window is displayed.

- In the Delete a Profile window:
  - Confirm the deletion by entering the name of the profile and pressing ENTER.
  - Or, press ENTER without entering anything to cancel the action.

## Direct Commands and Batch Processing

The SYSPARM utility functions can also be executed in direct-command mode or batch mode.

In addition to the functions provided via the SYSPARM utility menus, options are provided in batch mode for specifying commentary text, the input delimiter and the input assign character.

### Related Topics about Batch Mode:

Natural in Batch Mode (Natural Operations documentation)

Using the INPUT Statement in Non-Screen Modes (Natural Statements documentation)

Using the INPUT Statement in Batch Mode on Mainframe Computers (Natural Statements documentation)

The following section contains information on:

- Batch Processing
- Syntax of SYSPARM
- with-clause
- where-clause
- Example of SYSPARM in Batch

## Batch Processing

Described below are general instructions that apply when executing SYSPARM in batch mode:

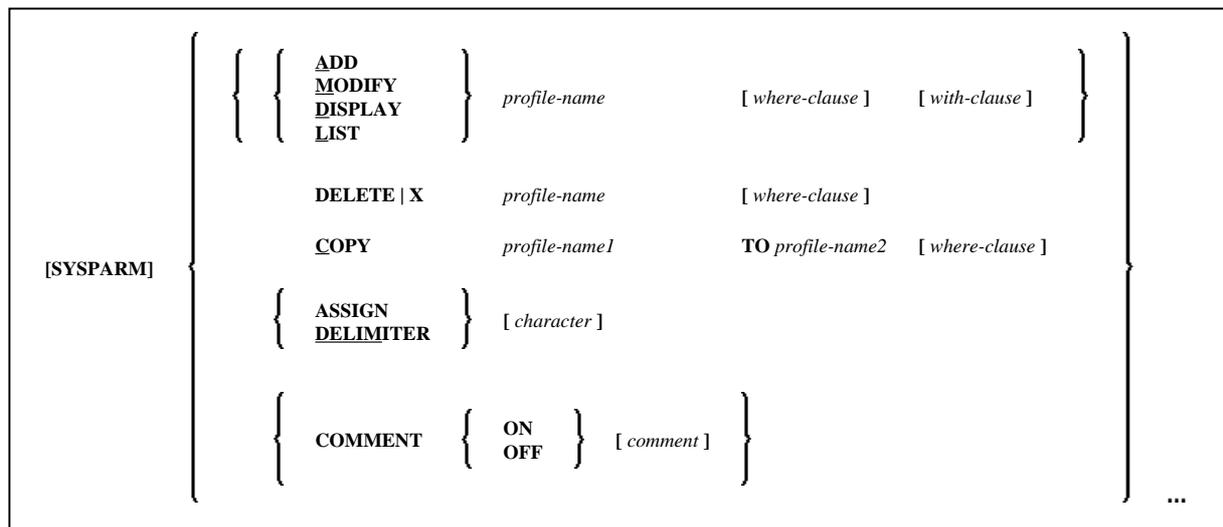
- To terminate SYSPARM, in a separate line, enter a period (.) or FIN, where FIN ends the Natural session.
- To execute more than one SYSPARM function, specify each function in a separate line.
- To extend a function over two lines, anywhere in the first line enter the character defined with the session parameter CF (default is %). This indicates continuation on the next line.

See also Example of SYSPARM in Batch below.

## Syntax of SYSPARM

Displayed below is the syntax that applies when processing SYSPARM direct commands or in batch mode. For the restrictions that apply when using direct commands, see the explanations of keywords below.

The symbols used in the diagram are explained in the section System Command Syntax in the Natural System Command Reference documentation.



### Explanation of Keywords

The following section contains a brief description of the keywords and clauses indicated in the SYSPARM syntax diagram above, along with special instructions, if required. For details on the keywords that represent commands, refer to the corresponding SYSPARM functions described earlier in this section.

In addition, Example of SYSPARM in Batch below demonstrates the use of the keywords and clauses.

Keyword	Explanation
SYSPARM	Invokes the SYSPARM utility. This keyword is only required when executing SYSPARM commands from the NEXT or MORE prompt.
<u>A</u> DD	Executes the Add Profile function.  Use the keyword WITH to clearly separate the command ADD from the input line(s) that follows and that makes up the contents of the profile.  For example:  <code>ADD <i>profile-name</i> WITH AUTO=ON</code>  To terminate an ADD command, enter a period (.) in a separate line.  See also Editing Profiles.
<u>M</u> ODIFY	Executes the Modify Profile function.  To terminate a MODIFY command, enter a period (.) in a separate line as shown in the example below.  To insert a commentary text in a profile, place the comment between the characters /* and */.
<u>D</u> ISPLAY	Executes the Display Profile function.
<u>L</u> IST	Executes the List Profiles function.
DELETE or X	Executes the Delete Profile function.
<u>C</u> OPY	Executes the Copy Profile function.
ASSIGN	Specifies or displays the input assign character. You need to change the input assign character before you start processing data that contains the input assign character defined for your current session. Otherwise, you will receive Natural error NAT1105 (Undefined keyword ... in input).  If no value is specified, the output contains the character defined for your current session.
<u>D</u> ELIMITER	Specifies or displays the input delimiter character. You need to change the input delimiter character before you start processing data that contains the input delimiter character defined for your current session. Otherwise, you will receive Natural error NAT1105 (Undefined keyword ... in input).  If no value is specified, the output contains the character defined for your current session.

Keyword	Explanation
SYSPARM	Invokes the SYSPARM utility. This keyword is only required when executing SYSPARM commands from the NEXT or MORE prompt.
<u>A</u> DD	Executes the Add Profile function.  Use the keyword WITH to clearly separate the command ADD from the input line(s) that follows and that makes up the contents of the profile.  For example:  <code>ADD <i>profile-name</i> WITH AUTO=ON</code>  To terminate an ADD command, enter a period (.) in a separate line.  See also Editing Profiles.
<u>M</u> ODIFY	Executes the Modify Profile function.  To terminate a MODIFY command, enter a period (.) in a separate line as shown in the example below.  To insert a commentary text in a profile, place the comment between the characters /* and */.
<u>D</u> ISPLAY	Executes the Display Profile function.
<u>L</u> IST	Executes the List Profiles function.
DELETE or X	Executes the Delete Profile function.
COMMENT	Not applicable to the command REPLACE.  Activates (ON) the comment option and writes a comment after each parameter that is modified, or deactivates (OFF) the comment option.  If activated, and if no individual <i>comment</i> has been specified (see below), the comment inserted for each parameter contains the following data: <ul style="list-style-type: none"> <li>● The ID of the user who last modified the profile as generated by the Natural system variable *USER and</li> <li>● The date and time when the profile was last modified as generated by the Natural system variables *DATV and *TIMX. (See also the relevant sections in the documentation Natural System Variables.)</li> </ul> To modify a comment, use the REPLACE command (see below).
<i>character</i>	Any special character: see the Natural session parameters ID (Input Delimiter Character) and IA (Input Assign Character) described in the section Session Parameters in the Natural Parameter Reference documentation.
<i>comment</i>	A commentary text that is to be placed between the characters /* and */.
<i>profile-name</i>	The name of a profile or a range of profiles:  Ranges of profiles are only allowed with the command LIST or, in batch mode, the command LIST or DISPLAY.

Keyword	Explanation
SYSPARM	Invokes the SYSPARM utility. This keyword is only required when executing SYSPARM commands from the NEXT or MORE prompt.
<u>A</u> DD	Executes the Add Profile function.  Use the keyword WITH to clearly separate the command ADD from the input line(s) that follows and that makes up the contents of the profile.  For example:  <code>ADD <i>profile-name</i> WITH AUTO=ON</code>  To terminate an ADD command, enter a period (.) in a separate line.  See also Editing Profiles.
<u>M</u> ODIFY	Executes the Modify Profile function.  To terminate a MODIFY command, enter a period (.) in a separate line as shown in the example below.  To insert a commentary text in a profile, place the comment between the characters /* and */.
<u>D</u> ISPLAY	Executes the Display Profile function.
<u>L</u> IST	Executes the List Profiles function.
DELETE or X	Executes the Delete Profile function.
<i>profile-name1</i> <i>profile-name2</i>	Only applies to the command COPY.  The source profile ( <i>profile-name1</i> ) from which to create a new profile ( <i>profile-name1</i> ), and the target profile ( <i>profile-name2</i> ) into which to copy the data.
WHERE	Optional keyword that indicates the start of a <i>where-clause</i> . See where-clause below.
WITH	Only applies to the commands ADD, DISPLAY, MODIFY and LIST.  When used in connection with ADD, the <i>with-clause</i> consists of the keyword WITH followed by the profile contents.  Optional keyword that indicates the start of a <i>with-clause</i> . See with-clause below.
***	Only applies in batch mode.  Illustrates that you can specify more than one SYSPARM function by placing each function in a separate line.

## where-clause

[WHERE] [DBID <i>dbid</i> ] [FNR <i>fnr</i> ] [PASSWORD <i>password</i> ] [CIPHER <i>cipher</i> ]
---

Below is a description of the keywords that are indicated in the *where-clause* and the values that apply:

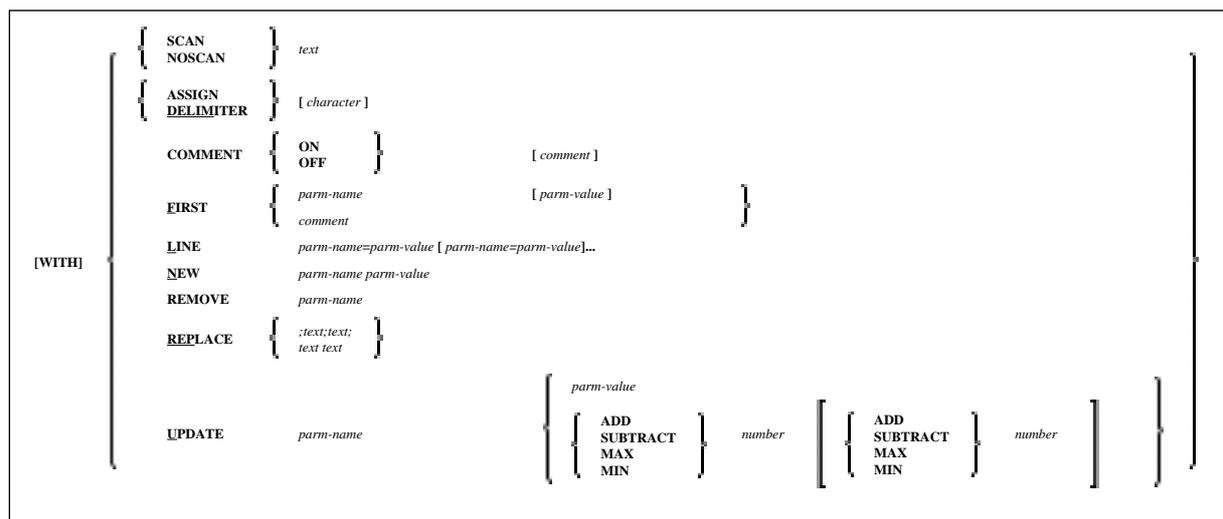
**Explanation of Keywords - where-clause**

The *where-clause* is optional. Its syntax is as follows:

Keyword	Valid Values
DBID	The database ID (DBID) of the Adabas file where the profile is stored.
FNR	The file number (FNR) of the Adabas file where the profile is stored.
PASSWORD	The 8-character Adabas password of the Adabas file where the profile is stored.
CIPHER	The 8-digit cipher code of the Adabas file where a profile is stored.

**with-clause**

The *with-clause* is optional and applies to the commands ADD, DISPLAY, MODIFY and LIST. Its syntax is as follows:



**Explanation of Keywords - with-clause**

Below is a description of the keywords that are indicated in the *with-clause* and the values that apply:

Keyword	Valid Values
SCAN	Only applies to the commands LIST and DISPLAY.  Scans one or more profiles available in the current Natural system file for a <i>text</i> string (for example, a parameter or a parameter value) and provides a list of the profiles that contain the <i>text</i> string specified.
NOSCAN	Only applies to the commands LIST and DISPLAY.  Exempts a <i>text</i> string from the scan over one or more profiles available in the current Natural system file and provides a list of the profiles that do <b>not</b> contain the text string specified.
ASSIGN	See ASSIGN above.
DELIMITER	See DELIMITER above.

Keyword	Valid Values
COMMENT	See COMMENT above.
<u>F</u> IRST	Moves a parameter and its value or a comment to the first position in a profile. This can be required, for example, if you want to specify the profile parameter PARM.  Comments contained in the first position of a profile will remain in this position. A parameter and its value will only be moved to the first position if no comment occupies this position. Otherwise, the parameter will be appended after the comment(s). A comment inserted with FIRST will always be placed in the first position, before any comment(s) that originally occupied this position.
<u>L</u> INE	Places the parameter(s) specified after LINE in a new line at the end of a profile.
<u>N</u> EW	Appends a parameter at the end of a profile but not necessarily in a new line. To place a parameter in a new line, use the parameter LINE.
<u>R</u> EMOVE	Removes a parameter and the value assigned (including parentheses) from a profile. The parameter to be removed can be a subparameter, such as the profile parameter DFS, which is a subparameter of the profile parameter RPC.
<u>R</u> EPLACE	Replaces an old <i>text</i> string with a new <i>text</i> string contained in a profile. See also the option COMMENT above.
<u>U</u> PDATE	Updates the value assigned to a parameter. If the parameter specified does not exist yet, it will be appended to the profile.
<i>parm-name</i>	The full name of a parameter.
<i>comment</i>	See <i>comment</i> above.
<i>parm-value</i>	The value assigned to a parameter.

Keyword	Valid Values
<i>text</i>	<p>Any text string contained in a profile.</p> <p>The following applies to the commands SCAN and NOSCAN: The text string must not contain any blank characters.</p> <p>The following applies to the REPLACE command:</p> <p>If neither the old <i>text</i> nor the new <i>text</i> contains blank characters, place a blank character between old and new <i>text</i>. If the old <i>text</i> or the new <i>text</i> contains blank characters, place the input delimiter character specified between old <i>text</i> and new <i>text</i> and around the entire old/new <i>text</i> string.</p> <p>For example:</p> <pre> ;This comment is old;This comment is new;</pre> <p>As an alternative to the delimiter character specified, you can choose any of the following characters:</p> <ul style="list-style-type: none"> <li>' A single quotation mark.</li> <li>, A comma.</li> <li>. A period.</li> <li>; A semicolon.</li> <li>/ A slash.</li> <li>\ A back slash.</li> <li>  A vertical bar.</li> </ul>
ADD	Increases the size of a parameter value by adding the number specified. The addition can be limited to a maximum or minimum value by specifying MAX or MIN (see below).
SUBTRACT	Decreases the size of a parameter value by subtracting the number specified. The subtraction can be limited to a maximum or minimum value by specifying MAX or MIN (see below).
MAX	Specifies the maximum of a parameter value.
MIN	Specifies the minimum of a parameter value.
<i>number</i>	A numeric value.

### Example of SYSPARM in Batch

The example profile and the example input below illustrate the use of SYSPARM commands and the result of profile modifications executed in batch:

- Example Profile TESTPROF - Before Job Execution
- Example Input
- Example Profile TESTPROF - After Job Execution
- Example Profile TESTPRO1 - After Job Execution

**Example Profile TESTPROF - Before Job Submission**

```

/* This is a test profile. */
AUTO=ON FNAT = (102,110,PASSWORD) FUSER=(1099,1100,PASSWORD,12345678)
RPC=(RPCSIZE=80,SRVNAME=MYSERV,SERVER=ON,DFS=(SRV2,NODE1,,ACI))
PRINT=((2,12,18),AM=STD,DEST='PRINT**',OPEN=INITOBJ,CLOSE=CMD)
PRINT=((1,3,6-11,15),AM=NAF)
ESIZE=90

```

**Example Input**

```

/*JCL
.
.
.
*/
SYSPARM
COPY TEST1 TO TESTPROF WHERE DBID 10 FNR 32
DISP TESTPROF
COMMENT ON /* TESTUSER 29.Jul.2003 */
MODIFY TESTPROF with REM DFS
  REPLACE ;test profile;test profile for SYSPARM in batch;
  NEW BPSIZE 4096
  UPDATE ESIZE ADD 20 MAX 100
  FIRST PARM INHOUSE
  DELIMITER $
  UPDATE FUSER (,6)
  ASSIGN :
  LINE IM=D, INTENS=1, AUTO=T, MT=0, MADIO=0
.
DISP TESTPROF
ADD TESTPRO1 WITH /* 106,210 */
  FNAT=(106,210,PASSWORD),FUSER=(,211)
.
DISPLAY TESTPRO1
DELETE TESTPRO2 WHERE DBID 10 FNR 32
X TESTPRO3 WHERE DBID 10 FNR 32
.
FIN

```

**Example Profile TESTPROF - After Job Execution**

```

/* This is a test profile for SYSPARM in batch. */
PARAM=INHOUSE /* TESTUSER 29.Jul.2003 */
AUTO=ON FNAT = (102,110,PASSWORD) FUSER=(,6) /* TESTUSER 29.Jul.2003 */
RPC=(RPCSIZE=80,SRVNAME=MYSERV,SERVER=ON, )
PRINT=((2,12,18),AM=STD,DEST='PRINT**',OPEN=INITOBJ,CLOSE=CMD)
PRINT=((1,3,6-11,15),AM=NAF)
ESIZE=100 /* TESTUSER 29.Jul.2003 */ BPSIZE=4096 /* TESTUSER 29.Jul.2003
*/
IM=D, INTENS=1, AUTO=T, MT=0, MADIO=0 /* TESTUSER 29.Jul.2003 */

```

**Example Profile TESTPRO1 - After Job Execution**

```

/* 106,210 */
FNAT=(106,210,PASSWORD),FUSER=(,211)

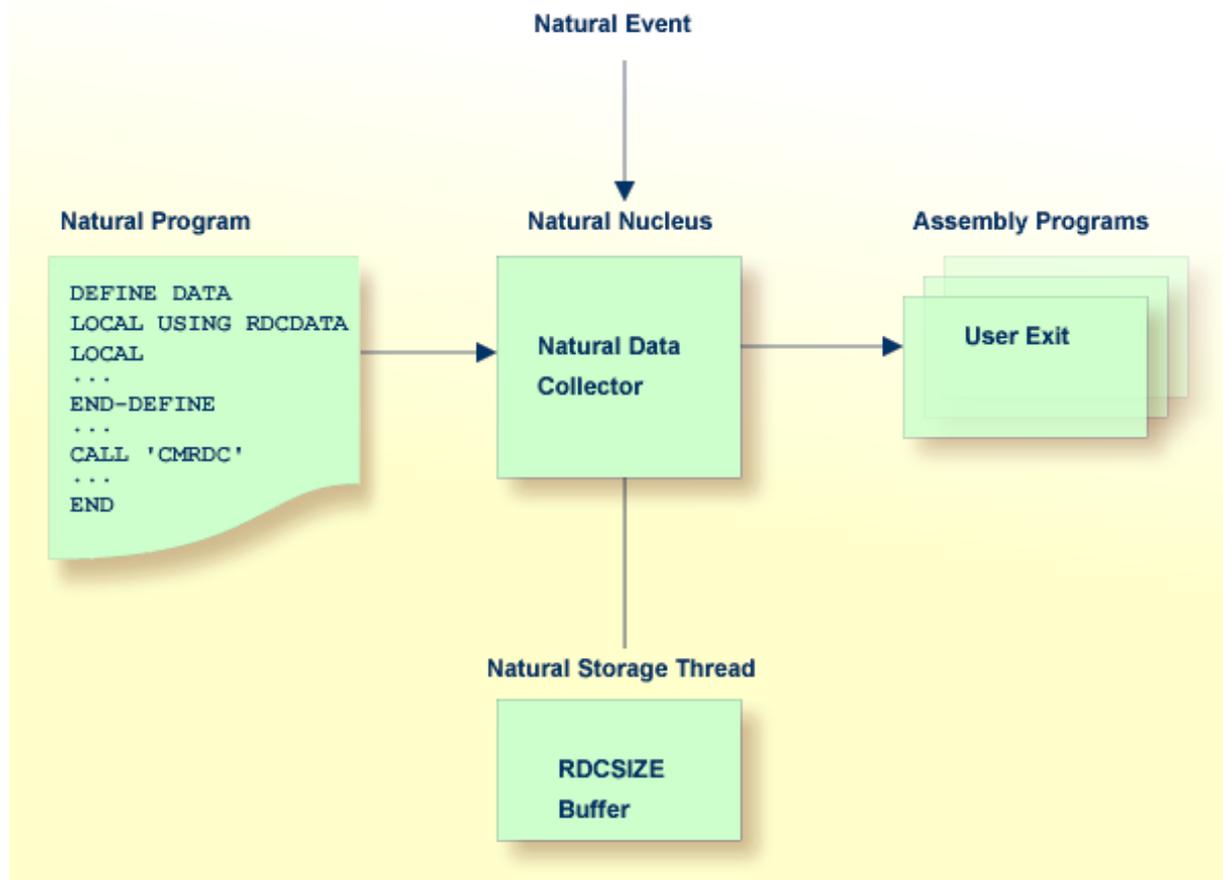
```

# SYSRDC Utility

The utility SYSRDC is used to record monitoring and accounting data on the internal processing flow within a Natural application. This data can be used for evaluating Natural session activities in external or Natural programs.

The data are collected at events performed within Natural. To collect data and supply it for further processing, as illustrated in the graphic below, the SYSRDC utility provides the following components:

- Natural Data Collector as part of the Natural nucleus:  
Collects data and controls data recording in the RDCSIZE buffer.
- User exits:  
Pass data to external monitoring and accounting programs.
- CMRDC interface:  
Used to evaluate data in a Natural program for the current Natural session.



The section below covers the following topics:

- Data-Collecting Events
- Data Collected
- Activating the Natural Data Collector
- Trace Recording
- User Exits for External Monitoring/Accounting
- Calling the CMRDC Interface

## Data-Collecting Events

Events specify activities within Natural. The table below lists the events at which data is collected within Natural and the types of event available.

Each type of event is assigned a one- or two-letter event code where the first letter represents the type of event and any second letter the subtype.

For example, in the event code **PL**, **P** represents the event type "program" and **L** the subtype "load".

Event	Event Code
At session initialization.	SI
At session termination.	ST
At program load.	PL
At program start.	PS
At program termination.	PT
Before a database call.	DB
After a database call.	DA
Before a terminal I/O.	IB
After a terminal I/O.	IA
Before a call of a non-Natural program.	CB
After a call of a non-Natural program.	CA
At a runtime error.	E
At an internal trace call.  The information internal traces provide are only intended for debugging purposes by Software AG personnel. See also "To activate an internal trace call event" below.	N
At a Natural statement.	NS
At a user-defined event	U

### To activate an internal trace call event

- Specify the Natural profile parameter ITRACE=ON.
- Define the Natural components that are to issue internal trace calls:
  - By using the NTTRACE macro of the Natural parameter module.
  - Or, by setting the dynamic Natural profile parameter TRACE.

The NTTRACE macro and the parameters ITRACE and TRACE are described in the Natural Parameter Reference documentation.

## Data Collected

The data the Natural Data Collector collects at events are described in the following section.

For the layout of the data, see the Natural source dataset NAMRDC, or the local data area RDCDATA that is provided in the Natural system library SYSRDC.

The data collected can be split into two categories: general and event-specific data:

- General Data
- Event-Specific Data

### General Data

The following general data is collected at every event:

- Software AG product name
- Product version
- Operating system
- TP monitor
- Run modes, such as addressing or residence mode of the Natural nucleus and the buffer pool.
- TP user or batch job name
- TP terminal ID
- Current Natural user ID
- Current Natural Security user group ID
- Current Natural library
- Current Natural program,
- Current program level,
- Line number of program statement currently executed.

### Event-Specific Data

The following data is only collected at the following events:

Event	Data Elements
Session initialization	none
Session termination	Termination return code. Natural termination message code NAT99nn. Name of back-end program.
Program load	Name of program to be loaded. Name of load library.
Program start/termination	Program type. Program name. Program library name. Database ID of program library.
Database call	Database type. Command code. Command ID. Database ID. Response code. Pointer that lists parameter addresses. (Only useful for user exits.)
Terminal I/O	Number of bytes sent. Number of bytes read. Total session storage allocated. Compressed session storage length.
Call of non-Natural program	Name of program called. Calling mode, such as dynamic or static mode. Program link location. Parameter type. Parameter address. Program entry address. Response code.
Runtime error	Natural system error message code. External abend code Name of error handling program as specified with the Natural profile parameter ETA (see the relevant section in the Natural Parameter Reference documentation).
Internal trace	Up to 250 bytes of information on Natural nucleus components. The information provided is only intended for debugging purposes by Software AG personnel.
User-defined event	Up to 250 bytes of user-defined information.

## Activating the Natural Data Collector

The Natural Data Collector is activated and controlled by the Natural profile parameter RDCSIZE (see also the relevant section in the Natural Parameter Reference documentation).

This parameter determines the size (in KB) of the RDCSIZE buffer for the Natural Data Collector and controls trace recording as described in the relevant section.

By default, the parameter is set to RDCSIZE=0, which means that the Natural Data Collector is deactivated.

### ▶ To activate the Natural Data Collector

- Set RDCSIZE=2.

The Natural Data Collector is activated and the data collected is supplied to the user exits. No data is recorded in the RDCSIZE buffer.

To record data in the RDCSIZE buffer, see Trace Recording below.

## Trace Recording

The event data collected is always supplied to user exits to be used by external monitoring/accounting programs as described in the relevant section. Independently of the user exists, you can record the event data of the current Natural session in the RDCSIZE buffer. This can be useful for testing purposes. In this section, the recording of data in the RDCSIZE buffer is referred to as trace recording.

Trace recording starts when initializing a session and ends when terminating a session. The data recorded in the RDCSIZE buffer can be accessed by any Natural program within the same session.

The RDCSIZE buffer is filled in wrap-around mode; that is, the oldest record is overwritten when the buffer becomes full. At the end of the session, the contents of the buffer are deleted.

Trace recording is activated with the profile parameter RDCSIZE (see also the relevant section in the Natural Parameter Reference documentation). This parameter also determines the size of the RDCSIZE buffer.

### ▶ To activate and use trace recording

1. Set the Natural profile parameter RDCSIZE to a value greater than 2.

Use the following formula to calculate the number of records that fit into the RDCSIZE buffer:

$$11 * (RDCSIZE - 2) = \text{number of records}$$

All event data collected for the current Natural session is recorded in the RDCSIZE buffer.

2. Retrieve the contents of the RDCSIZE buffer by using the CMRDC interface as described in Calling the CMRDC Interface.
3. If desired, use the CMRDC interface to stop or restart and trace recording in the RDCSIZE buffer, specify events for which data is to be recorded or execute additional functions described in Calling the CMRDC Interface.

## User Exits for External Monitoring/Accounting

The event data can be passed to external monitoring and accounting programs for evaluation of activities in Natural sessions. This is accomplished with the user exits and the examples of user-exit programs provided. User-exit programs are written in assembly language.

The Natural Data Collector supports any number of user exits. A user exit can be defined by the Natural profile parameter RDCEXIT (see the relevant section in the Natural Parameter Reference documentation). An external monitoring/accounting program can be attached to each user exit.

Three exit names are predefined: RDCEX1, RDCEX2 and RDCEX3. If you use one of these names as entry point for your exit (which is linked to the Natural nucleus), the Natural profile parameter RDCEXIT is not required.

At every event listed under Data-Collecting Events, the user exits take over control by using the standard linkage call conventions described in the table below:

Register	Contents
1	Points to a parameter address list that consists of two addresses: one address points to the general data and the other points to event-specific data. The layout of these areas is mapped by the DSECT RDCGDATA and RDCLDATA respectively. Both DSECTs are supplied in source form in the Natural macro NAMRDC.
13	Points to a 72-byte standard save area.
14	Contains the return address.
15	Contains either the entry point address or the return code of the user exit.

**Note:**

The user exits are called independently of the CMRDC interface.

A user-exit program must have the same attributes as Natural; that is, it must have the same addressing mode, and it has to be reentrant. It must be linked with the Natural nucleus according to the conventions of statically linked non-Natural programs. See also the profile parameters CSTATIC and RCA described in the Natural Parameter Reference documentation.

By default, a 400-byte exclusive work area (per session) is supplied for each user exit (the field RDCGWRKA). If a larger work area is required for a user exit, it can be specified after the user-exit name in the Natural profile parameter RDCEXIT (see the relevant section in the Natural Parameter Reference documentation). The work-area length is passed on to the user exits in the field RDCGWRKL and can be used for verification. The location of this work area can change during a session due to Natural relocation, but its contents are preserved.

In TP-monitor environments, the TP anchor address is supplied (the field RDCGANCH); for example, the CSA address under CICS. It can be used to access system information.

If a Program Check occurs during the execution of a user-exit program, further data collecting is disabled for the rest of the session to avoid recursive abend situations.

The section below contains information on:

- Return Codes
- Examples of User-Exit Programs

## Return Codes

Non-zero return codes are only supported for two events:

- Before a database call, where Register 15 can contain an Adabas response code, which is stored in the control block; the Adabas call will not be executed.
- At program start, where Register 15 can contain a Natural error message code; the program will not be executed, but an error condition will occur with the specified number.

## Examples of User-Exit Programs

The following examples of user-exit programs are provided in the following Natural source datasets:

Program	Dataset	Used for
NAMRDC	NAT $nnn$ .SRCE	DSECT macro for general and event-specific data.
XNATRDC1	NAT $nnn$ .SRCE	Natural TSO Interface and Natural batch interface under operating systems OS/390 and VSE/ESA.
XNCFRDC1	NAT $nnn$ .SRCE	Com-plete.
XNCIRDC1	NCI $nnn$ .SRCE	Natural CICS Interface.

## Calling the CMRDC Interface

The CMRDC application programming interface is used for retrieving and controlling the trace data recorded in the RDCSIZE buffer. This can be useful for testing purposes.

### To invoke the CMRDC interface

- In the Natural program, issue a CALL statement.

The functions provided by the CMRDC interface and the syntax that applies to the corresponding Natural CALL statements are described in the following section.

In addition, the Natural system library SYSRDC provides example programs and the appropriate local data area RDCDATA.

The section below contains information on:

- Retrieving Trace Records
- Stopping and Restarting Trace Recording
- Selecting Event Types for Trace Recording
- User-defined Events
- CMRDC Return Codes
- Example Programs in Library SYSRDC

## Retrieving Trace Records

To read the data from the RDCSIZE buffer, invoke CMRDC with the following Natural statement:

```
CALL 'CMRDC' function event-time gen-data event-data seq.-number
```

The following parameters are passed:

Parameter	Format/Length	Explanation
<i>function</i>	A1	Possible functions:  F Get first trace record.  G Get next trace record.  N Get record of sequence number specified.
<i>event-time</i>	N10	Time of event. A time must be specified in the format <i>HHMMSSXXXX</i> ( <i>HH</i> = hours, <i>MM</i> = minutes, <i>SS</i> = seconds, <i>XXXX</i> = fraction of a second).
<i>gen-data</i>	A252	General data.
<i>event-data</i>	A252	Event-specific data.
<i>seq.-number</i>	I4	Sequence number of record.  Only applies to Function N (see above).

The retrieval functions stop recording data in the RDCSIZE buffer implicitly. To restart recording, use Function S described below.

## Stopping and Restarting Trace Recording

To stop or restart trace recording data in the RDCSIZE buffer, invoke CMRDC with the following statement:

```
CALL 'CMRDC' function
```

The following parameter is passed:

Parameter	Format/Length	Explanation
<i>function</i>	A1	Possible functions:  S Clear RDCSIZE buffer and restart trace recording.  P Stop trace recording.

## Selecting Event Types for Trace Recording

By default, all events are selected for trace recording. Use this function if you wish only specific events to be recorded.

**Note:**

This function only selects the events at which data is to be recorded in the RDCSIZE buffer. It does not affect the data passed to the user exits. Neither does it affect the status (started/stopped) of trace recording.

To select the types of event to be recorded, invoke CMRDC with the following statement:

```
CALL 'CMRDC' function type...
```

The following parameters are passed:

Parameter	Format/Length	Explanation
<i>function</i>	A1	Possible function:  T Select events for trace recording.
<i>type</i>	A1, A2 or A3	The one- or two-letter event code for the type of event to be recorded as listed in Data-Collecting Events.  Specify any number of parameters for the event codes desired.  Alternatively, you can select a range of events or no event:  ALL All events.  All events with event codes that start with <i>value</i> .  <i>value</i> * For example, <b>P*</b> selects all events of the type program: PL, PS and PT.  blank A blank character selects no event.

## User-Defined Events

To specify a user-defined event, invoke CMRDC with the following statement:

```
CALL 'CMRDC' function record
```

The following parameters are passed:

Parameter	Format/Length	Explanation
<i>function</i>	A1	Possible function:  U User-defined trace event.
<i>record</i>	<i>Annn</i>	Trace record with a length ( <i>nnn</i> ) of up to 250 bytes.

## CMRDC Return Codes

Code	Meaning
0	Function executed successfully.
4	Last trace record Only applies to Functions <b>F</b> and <b>G</b> (see above).
8	Too few parameters for this function.
12	Invalid function code.
16	Natural Data Collector not active, for example, RDCSIZE=0.
20	Natural Data Collector disabled after an error.
24	No buffer space available for trace recording (RDCSIZE=2 or smaller).
28	Invalid parameter value. Only applies to Functions <b>T</b> and <b>N</b> (see above).

### Example Programs in Library SYSRDC

The Natural system library SYSRDC contains the following example programs:

Program	Function
RDCDISP	Display all records in the RDCSIZE buffer and show the fields as specified in the program. See also "Example Output of Program RDCDISP" below.
RDCSTART	Restart trace recording.
RDCSTOP	Stop trace recording.
RDCSET	Select events for trace recording.
RDCUSER	User-defined event.

### Example Output of Program RDCDISP

The example screen below shows the extract of an output report produced by the example program RDCDISP.

ETIME1	TY	GCUID	LV	GPGM	T	GCAPL	PRLIB	PRNAM	CO	DB	FN	RC
0.0000	SI	SAG				SYSTEM						
0.0002	DB	SAG				SYSTEM			OP	10		
0.0002	DA	SAG				SYSTEM			OP	10		
0.0000	DB	SAG				SYSTEM			S1	10	1640	
0.0005	DA	SAG				SYSTEM			S1	10	1640	
0.0001	PL	SAG				SYSTEM	SYSLIB	MAINMENU				
0.0001	PS	SAG	1	MAINMENU	F	SYSTEM	SYSLIB	MAINMENU		10	1640	
0.0001	PL	SAG	1	MAINMENU		SYSTEM	SYSLIB	NAT00029				
0.0000	DB	SAG	1	MAINMENU		SYSTEM			S1	10	1640	
0.0002	DA	SAG	1	MAINMENU		SYSTEM			S1	10	1640	
0.0000	PS	SAG	2	NAT00029	N	SYSTEM	SYSLIBS	NAT00029		10	1640	
0.0000	CB	SAG	2	NAT00029		SYSTEM		CMMPP	S			
0.0000	CA	SAG	2	NAT00029		SYSTEM		CMMPP	S			
0.0000	PT	SAG	2	NAT00029	N	SYSTEM	SYSLIBS	NAT00029				
0.0000	PL	SAG	1	MAINMENU		SYSTEM	SYSLIB	USR2003P				
0.0000	PS	SAG	2	USR2003P	N	SYSTEM	SYSLIB	USR2003P		10	1640	
0.0001	PT	SAG	2	USR2003P	N	SYSTEM	SYSLIB	USR2003P				
0.0000	CB	SAG	1	MAINMENU		SYSTEM		CMUB	S			

The following table describes the columns that are displayed on the example screen, and the variables to which they refer. For further information, see the comments in the program source and the local data area RDCDATA. For an explanation of the Natural system variables mentioned, refer to the Natural System Variables documentation.

Column	Explanation
ETIME1	Time interval in seconds between the execution of the current and the previous event.
TY	Type of event as listed in Data-Collecting Events.
GCUID	Current Natural user ID as assigned by the Natural system variable *USER.
LV	Program level.
GPGM	Name of the current program as assigned by the Natural system variable *PROGRAM.
T	Type of program.
GCAPL	ID of the current application library as assigned by the Natural system variable *APPLIC-ID.
PRLIB	ID of the library where the program is stored. Applies to events of the type program, for example, Event Code PL.
PRNAM	Name of the program to be loaded for the type of event.
CO	Database command.
DB	Database ID.
FN	File number of Database.
RC	Database response code.

# SYSRPC Utility - Overview

This documentation describes the Natural SYSRPC utility which is used to maintain remote procedure calls.

## Related topics:

- For information on how to apply the SYSRPC utility functions to establish a framework for communication between server and client systems, refer to the Natural RPC (Remote Procedure Call) documentation.
- For an explanation of expressions relevant to the SYSRPC utility, see also Definition of Terms in the overview page of the Natural RPC documentation.
- The use of SYSRPC can be controlled by Natural Security. See Protecting Utilities in the Natural Security documentation.

The SYSRPC utility documentation covers the following topics:

	Basic Functionality	Invoking and terminating SYSRPC and commands, functions and options provided.
	Service Directory Maintenance	Maintaining client/server connections.
	Stub Generation	Generating client stubs.
	Parameter Maintenance	Modifying Natural RPC settings.
	Server Command Execution	Ping and Terminate.
	Remote Directory Maintenance	Maintaining client/server connections for a remote directory server.

# SYSRPC - Basic Functionality

This section covers the following topics:

- Invoking SYSRPC
  - Terminating SYSRPC
  - Help
- 

## Invoking SYSRPC

There are two ways of invoking the SYSRPC utility:

### To invoke SYSRPC online from a Natural library

- Enter the direct command SYSRPC.  
The Client Maintenance screen of the SYSRPC utility appears.

### To invoke SYSRPC online from the Natural Main Menu

1. Select Maintenance and Transfer Utilities.  
The corresponding menu is displayed.
2. Select Maintain Remote Procedure Calls.  
The Client Maintenance screen of the SYSRPC utility appears.

From the Client Maintenance screen, you can invoke all functions available for RPC maintenance:

- Service Directory Maintenance
- Stub Generation
- Parameter Maintenance
- Server Command Execution
- Remote Directory Maintenance

See the corresponding sections for a description of these functions.

## Terminating SYSRPC

### To exit the SYSRPC main menu

- In the Code field, enter a period (.).  
Or choose PF3/Exit.

## Help

To invoke the online help function, choose PF1/Help.

# SYSRPC - Service Directory Maintenance

The Service Directory Maintenance function is used to maintain a service directory in order to connect the client's calling program to a subprogram on a server. The service information is stored in the subprogram NATCLTGS in the current library. Before invoking the Service Directory Maintenance function, logon to the application library (or one of its steplibs) used by the client at runtime.

**Attention:**

If NATCLTGS is stored in the library SYSRPC, we strongly recommend that you move NATCLTGS to the application library (or one of its steplibs) used by the client.

For further information on how to apply the Service Directory Maintenance function, refer to Specifying RPC Server Addresses as described in Operating a Natural RPC Environment in the Natural RPC documentation.

This section covers the following topics:

- Service Directory Concept
  - Invoking Service Directory Maintenance
  - Fields
  - Commands and PF Keys
  - Defining Logical Node Names and Logical Services
- 

## Service Directory Concept

The Service Directory Maintenance function has a hierarchical structure with a cascading list to assign subordinate to superior fields. The highest hierarchical level is node and the lowest is program. You cannot enter node, server, library and program in the same line. If you do so, a corresponding error message occurs. You need to enter the value of a subordinate field in the lines below the superior field. You can assign several servers to a node, several libraries to a server and several programs to a library.

The node and server names specified in the Service Directory are either physical names or logical names and logical services:

- Physical Nodes and Servers
- Location Transparency

## Physical Nodes and Servers

Physical node and server names denote the names of real nodes (valid TCP/IP or Entire Net-Work address) and servers.

In Example 1 below, two servers are defined for one node. Both servers are connected to the same node: ETB045. The remote CALLNAT to Subprogram SUB1 is executed on Server NRPC001, whereas Subprograms SUB2 and SUB3 are executed on Server NRPC002.

The server names specified here must be identical to the server names used in the Natural parameter module of the server tasks: see the profile parameter SRVNAME in the Natural Parameter Reference documentation. Analogously, the node name in the service directory must be identical to the node name specified for the server tasks: see the profile parameter SRVNODE in the Natural Parameter Reference documentation.

## Location Transparency

Location transparency is a concept where physical node names can be replaced by logical node names, and a combination of physical node and server names can be replaced by logical services.

Logical node names and logical services are defined with EntireX and are assigned to physical node and server names at Natural runtime.

In Example 1 below, \*LOCTRAN in the field Node indicates that the field Server contains the logical service NRPC001-LOGICAL. LOGBROKER=NODE in the field Node indicates the logical node name.

### Related Topics:

- Defining Logical Node Names and Logical Services.
- Using Location Transparency in Operating a Natural RPC Environment in the Natural RPC documentation.
- The relevant sections in the EntireX documentation.

## Invoking Service Directory Maintenance

### Attention:

The Service Directory Maintenance function invokes the Natural editor. As a result, data stored in the source work area may be lost when invoking Service Directory Maintenance. A corresponding message will warn you not to delete any existing entries unintentionally: choose PF12 to cancel the function or choose ENTER to confirm the action and clear the source work area.

The Service Directory Maintenance screen provides a maximum of 500 lines for input.

### To invoke the Service Directory Maintenance function

1. In the Code field of the Client Maintenance screen, enter **SM**.  
A window appears saying "Existing service definitions found" (does not appear if the Service Directory is empty).
2. In the Code field, enter **A** (default) to append a new definition,  
Or enter **I** to ignore existing services (all existing service definitions will be deleted).

The default view of the Service Directory is displayed as shown in Example 1 below.

### Example 1 - Default View of Service Directory

```

15:32:25          *** NATURAL Remote Procedure Call ***          2002-05-24
                    Service Directory                               SYSRPC

      Node          Tr.          Server          Logon  Library          Program
1     ETB045_____ B          _____          -      _____          _____
2     _____          -          NRPC001_____          N      _____          _____
3     _____          -          _____          -      SYSTEM_____          _____
4     _____          -          _____          -      _____          SUB1_____
5     _____          -          NRPC002_____          Y      _____          _____
6     _____          -          _____          -      SYSTEM_____          _____
7     _____          -          _____          -      _____          SUB2_____
8     _____          -          _____          -      _____          SUB3_____
9     *LOCTRAN_____          -          _____          -      _____          _____
10    _____          B          NRPC001-LOGICAL_          N      _____          _____
11    _____          -          _____          -      SYSTEM_____          _____
12    _____          -          _____          -      _____          SUB1_____
13    LOGBROKER=NODE          B          _____          N      _____          _____
14    _____          -          NRPC002_____          N      _____          _____
15    _____          -          _____          -      SYSTEM_____          _____
16    _____          -          _____          -      _____          SUB2_____

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  LocTr Exit  <  -H  +H  -P  +P  Top  Bot>  Canc
    
```

If you choose PF4 or enter the less than (<) sign in the direct command line, the extended node/server view of the Service Directory is displayed as shown in Example 2 below.

**Example 2 - Extended Node/Server View of Service Directory**

```

14:48:33          *** NATURAL Remote Procedure Call ***          2002-05-29
                    Service Directory                               SYSRPC

      Node          Tr.          Server          Logon
1     ETB045_____ B          _____          -
2     _____          -          NRPC001_____          N
3     _____          -          _____          -
4     _____          -          _____          -
5     _____          -          NRPC002_____          Y
6     _____          -          _____          -
7     _____          -          _____          -
8     _____          -          _____          -
9     *LOCTRAN_____          -          _____          -
10    _____          B          NRPC001-LOGICAL_          N
11    _____          -          _____          -
12    _____          -          _____          -
13    LOGBROKER=NODE          B          _____          N
14    _____          -          NRPC002_____          N
15    _____          -          _____          -
16    _____          -          _____          -

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  LocTr Exit  <  -H  +H  -P  +P  Top  Bot>  Canc
    
```

**Fields**

The Service Directory screen contains the following input fields (one entry per line):

Field	Description
Node	<p>The name of the node to which the remote CALLNAT is sent. See also Definition of Terms in the Natural RPC documentation.</p> <p>Maximum input:</p> <p style="margin-left: 40px;">Default view of the Service Directory: 16 characters,</p> <p style="margin-left: 40px;">Extended node/server view of the Service Directory: 32 characters,</p> <p style="margin-left: 40px;">Using the window Location Transparency (see PF2 below): 192 characters.</p>
Tr.	<p>Transport method:  <b>B</b> indicates the EntireX Broker ACI protocol.</p>
Server	<p>The name of the server on which the CALLNAT is to be executed. See also Definition of Terms in the Natural RPC documentation.</p> <p>Maximum input:</p> <p style="margin-left: 40px;">Default view of the Service Directory: 16 characters,</p> <p style="margin-left: 40px;">Extended node/server view of the Service Directory: 32 characters,</p> <p style="margin-left: 40px;">Using the window Location Transparency (see PF2 below): 192 characters.</p>
Logon	<p>Initiates a Natural logon to the server.</p> <p>This is possible on server or node level and applies to all definitions made on a hierarchically lower level.                      For example, if the LOGON option has been set for a certain server, it applies to all associated library and subprogram definitions.</p> <p>Possible values are:</p> <p>Y      If set to Y (Yes), the client initiates a Natural logon to the server with the library name of the current library on the client, regardless of the library specified in the Service Directory.</p> <p>N      If set to N (No) or, if no value is entered, no logon is initiated.</p> <p>blank</p> <p>After the remote CALLNAT has been executed (successfully or not), the server library is reset to its previous state. For more information, see Using the Logon Option in the Natural RPC documentation.</p> <p>See also Server Command Execution.</p>
Library	<p>SYSTEM or the name of the library to which your client application is logged on during the execution of the remote CALLNAT is running.</p>
Program	<p>The name of the remote subprogram to be accessed from the client.</p>

### Selection Criteria for Node and Server

At Natural runtime, the selection of a node and server depends on the input value of the fields Program and Library. Comply with the following conditions:

**Non-conversational CALLNAT**

1. The field Library must contain the name of the current application library or SYSTEM.
2. The name of the subprogram specified in the CALLNAT statement must be contained in the program(s) subordinate to the library as located under Step 1 above.

**Conversational CALLNAT**

1. The field Library must contain the name of the current application library or SYSTEM.
2. The name(s) of the subprogram(s) specified in the OPEN CONVERSATION statement must be contained in the program(s) subordinate to the library as located under Step 1 above.

## Commands and PF Keys

Below is information on:

- Line Commands
- Direct Commands and PF Keys

### Line Commands

The line commands available to edit the Service Directory screen are listed below.

Enter a line command at the beginning of a line, that is, overwrite the sequential number and choose ENTER.

See also To copy or move a block of lines below and the direct command RESET.

Line Command	Function
A	Copies/moves the block of line marked with CC or MM (see below) below the line in which the command was entered.
CC	Delimits a block of lines to be copied.
D	Deletes the line marked.
DD	Delimits and deletes a block of lines.  Delimit a block of lines by entering the command in the first and the last line of the block and choose ENTER to execute the command.
I	Inserts five empty lines below the line in which the command was entered.
MM	Delimits a block of lines to be moved.
P	Copies/moves the block of lines marked with CC or MM above the line in which the command was entered.

 **To copy or move a block of lines**

1. At the beginning of the line where the block starts, enter CC or MM.
2. At the beginning of the line where the block ends, enter CC or MM.
3. Choose ENTER.  
The line commands disappear, the sequence numbers are displayed again and the block has been marked.
4. Go to the beginning of the line where you want to place the block and enter A (after) to copy or move the block **below** this line.

Or enter P (prior) to copy or move the block **above** this line.

Note that you can only execute A or P on lines with at least one field filled.

5. Choose ENTER.

## Direct Commands and PF Keys

The following direct commands and PF keys are available in the Service Directory screen:

Direct Command	PF Key	Function
<u>EXPIRATION</u>		<p>The remote directory data are loaded at runtime. The expiration time determines the period of validity of this data. If directory data are requested after the expiration time set, they will automatically be reloaded. If expiration time is set to 0 seconds, the remote directory data will not be reloaded.</p> <p>With the direct command EXPIRATION, enter an expiration time in seconds, for example, EXPIRATION 86400. Maximum is an 8-digit number.</p> <p>If you do not provide a parameter with the command, the Expiration Time window appears where you can display or modify the current time.</p>
RESET		<p>Removes the marks set with the line commands CC, MM and DD as described in Line Commands above.</p> <p>Note that if blocks have been marked incorrectly, a corresponding message occurs and you must remove the erroneous line command before you enter RESET.</p>
	PF1	Invokes the editor online help.
	PF2	Invokes the Location Transparency window where you can define a logical node name or a logical service as described in Defining Logical Node Names and Logical Services below.
	PF3	Exit. Prompts you to save modifications and exit the Service Directory screen.
<	PF4	<p>Displays the extended view of the fields Node and Server. The extended node/server view does not display the fields Library and Program as shown in Example 2 above.</p> <p>Choose PF11 to switch back to the default view.</p>
-H	PF5	Scrolls half a page backward/forward.
+H	PF6	
-P	PF7	Scrolls one page backward/forward.
+P	PF8	
TOP	PF9	Scrolls to the beginning of the list.
BOT	PF10	Scrolls to the end of the list.
>	PF11	Switches back to the default view of the Service Directory (see Example 1) if the extended view of the fields Node and Server view was invoked earlier by choosing PF4.
<u>CANCEL</u>	PF12	Exits the Service Directory screen without saving any modification.

## Defining Logical Node Names and Logical Services

Logical node names or logical services can only be defined for node or server fields that already contain any values.

Note that defining a logical service, the original (physical) node name will be replaced by \*LOCTRAN and it is **not** possible to automatically convert back logical node names or logical services. To remove logical names and services, see below.

### To define a logical service

- Place the cursor on a Server field and choose PF2/LocTr.  
The window "Location Transparency - Logical Service" appears.
- If desired, modify the existing values.
- Choose ENTER.  
The window Server Type Conversion appears and provides the following two options:
  - To confirm the conversion, enter Y .  
The value in the field Node that relates to the specified server is replaced by \*LOCTRAN. This indicates that a node/server combination was converted into a logical service.
  - To cancel the function, enter any character except Y (Yes) or do not enter any value.  
The physical node and server names are retained.

### To define a logical node name

- Place the cursor on a Node field and choose PF2/LocTr.  
The window "Location Transparency - Logical Node Name" appears with the preset value of LOGBROKER=*name*  
where *name* denotes the logical EntireX Broker name.  
If desired, modify *name* (but do not modify LOGBROKER=).
- Execute or cancel the function:
  - To confirm the conversion, choose ENTER.  
The physical node name is converted into a logical name.
  - To cancel the function, choose PF12.  
The physical node name is retained.

### To remove a logical node name or logical service

- For a logical node name: in the Node field, remove the string LOGBROKER=.

For a logical service: delete the logical service and insert a physical server(s) by using the line commands **D** and **I** as described in the relevant section above.

# SYSRPC - Stub Generation

The Stub Generation function is used to generate client stub subprograms. Though stubs are actually not required if automatic Natural RPC execution is used, it can be advantageous to generate them anyway.

For more details, see Stubs and Automatic RPC Execution in the section Operating a Natural RPC Environment in the Natural RPC documentation.

This section covers the following topics:

- Invoking Stub Generation
  - Parameter Specification
  - Example of Stub Generation
- 

## Invoking Stub Generation

### Attention:

The Stub Generation function invokes the Natural editor. As a result, data stored in the source work area can be lost when invoking Stub Generation. A corresponding message will warn you not to delete any existing entries unintentionally: choose PF12 to cancel the function or choose ENTER to confirm the action and clear the source work area.

The stubs are generated in the current library. We strongly recommend that you log on to the application library (or one of its steplibs) used by the client at execution time of the remote CALLNAT.

### Invoking and using the Stub Generation function

1. In the Code field of the Client Maintenance screen, enter **SG**.  
The Generate Client Stub Routine window appears.
2. Enter the name of the stub subprogram to be generated.  
The name of the stub subprogram must be the same as the name of the remote CALLNAT program.  
The name of the library is preset with the name of the current library and cannot be changed.
3. Choose Compression Type 0, 1 or 2 (default is 1); see Using Compression as described in Operating a Natural RPC Environment in the Natural RPC documentation.
4. Choose ENTER.
  - If the subprogram specified in Step 2 already exists in the library assigned, a corresponding window is displayed:  
  
Specify **N** (No) and choose ENTER if you do not want to generate a new stub.  
You will be returned to the Client Maintenance screen.  
  
Specify **Y** (Yes) and choose ENTER.  
The parameter data area of the existing subprogram is displayed in the Stub Generation screen.
  - If the subprogram specified in Step 2 does not exist, an empty Stub Generation screen is displayed.
5. Add or modify the parameters to be used in the stub subprogram: see Parameter Specification below.  
On the Stub Generation screen, the same editor and line commands apply that are valid for the Service Directory Maintenance function (see the relevant section).
6. Choose ENTER to generate the stub subprogram and to exit.  
The stub subprogram is generated in the library assigned in Step 2.

A window appears which indicates the size the stub requires for sending data from the client to the server or vice versa. The size includes internal RPC information used for the stub. The indication of the size helps you configure the middleware layer used, for example, the EntireX Broker attribute file.

Below is the text that appears in the window when you generate a stub from the example subprogram TESTS5 (see Example of Stub Generation below):

```
Stub TESTS5 is generated in library TEST
  It requires:
    Send length: 2249 bytes
    Receive length: 2221 bytes
```

If the Send or Receive length exceeds the Entire Net-Work limit of 32000 bytes, a window appears with a corresponding warning. Enter **Y** (Yes) to continue, or **N** (No) to cancel the generation.

If you choose **Y**, this setting is kept for the entire SYSRPC session, that is, you can continue generating stubs without receiving further warnings.

If the total data (without internal RPC information) sent or received exceeds the limit of 1073733630 bytes (which is 1 GB minus 8 KB of internal RPC information), SYSRPC stops processing and issues a corresponding error message. This error message displays the subtotal of the data in bytes that could be transferred at the field up to which the subtotal was calculated. The corresponding field is then marked. In this case, reduce the amount of data and then continue generating the stub.

If the stub was generated in the library SYSRPC, you must move the stub to the application library or steplib using the appropriate Natural transfer utility (SYSMAIN, SYSTRANS or Natural Object Handler). Note that you may have to recatalog the stub sources in the target environment.

## Parameter Specification

In the input fields provided on the Stub Generation screen, you can specify the parameters that are used in the stub subprogram:

Field	Description
Attr	The attribute which specifies the parameter as: M (modifiable field), O (output field) or I (input field).
Type	The Natural data type, such as N (numeric). Data Types C and Handle are not allowed.
Len	The length of the variable.  Natural Data Type A is restricted to 1073733630 bytes, Data Type B is restricted to 536866815 bytes. Dynamic variables are not allowed.
Prec	Only applies to data types N (numeric) and P (packed). Optional.  The precision of the variable, that is, the number of digits after the decimal point.
Dim1/2/3	Only applies to arrays. Optional.  The first, second and third dimension of the variable.  The maximum of occurrences is 1073733630 bytes for each dimension.

## Example of Stub Generation

The following example shows four modifiable parameters that correspond to the variable definitions in the Natural subprogram TEST5 supplied in the Natural system library SYSRPC:

```

DEFINE DATA
PARAMETER
  2 #IDENTIFIER (A10)
  2 #N-OF-ID (I4)
  2 #FREQ (P5.2)
  2 #A100 (A100/5,4)

```

Stub Generation							
	Attr	Type	Len	Prec	Dim1	Dim2	Dim3
1	M	A	10				
2	M	I	4				
3	M	P	5	2			
4	M	A	100		5	4	

# SYSRPC - Parameter Maintenance

**Applies to client sessions only.**

The Parameter Maintenance function is used to dynamically (within a session) modify some of the RPC profile parameters set in the NATPARM parameter module.

**Attention:**

The parameter modifications are retained as long as the user session is active; they are lost when the session is terminated. Static settings are made using the Natural profile parameters.

This section covers the following topics:

- Invoking Parameter Maintenance
  - Specifying Profile Parameters
- 

## Invoking Parameter Maintenance

 **To invoke and use the Parameter Maintenance function**

1. In the Code field of the Client Maintenance screen, enter **PM**.  
The Client Parameter Maintenance screen appears.
2. Modify the values of the input fields: see Specifying Profile Parameters below.
3. Choose PF3/Exit to save modifications and exit the Client Parameter Maintenance screen.  
Or choose PF12/Cancel to exit without saving any parameter modifications.  
The SYSRPC Client Maintenance screen appears.

## Specifying Profile Parameters

In the input fields provided on the Client Parameter Maintenance screen, you can specify the following profile parameters:

For further information on profile parameter settings, see the section Profile Parameters in the Natural Parameter Reference documentation.

<b>Field</b>	<b>Explanation</b>
Timeout	<p>Specifies the number of seconds the client is to wait for an RPC server response.</p> <p>See also the profile parameter TIMEOUT as described in the Natural Parameter Reference documentation.</p>
Try Alternative Servers	<p>Specifies whether an RPC client is to try to execute a service on an alternative server (ON) or not (OFF). See also Using an Alternative Server in the Natural RPC documentation.</p> <p>See also the profile parameter TRYALT as described in the Natural Parameter Reference documentation.</p>
Compression for AUTORPC = ON	<p>Specifies the compression type for an automatically generated RPC call; see Using Compression as described in the Natural RPC documentation.</p> <p>See also the profile parameter COMPR as described in the Natural Parameter Reference documentation.</p> <p>For more information on automatic RPC execution, see Working with Automatic Natural RPC Execution (Natural RPC documentation).</p>

# SYSRPC - Server Command Execution

The SYSRPC utility provides the server execution commands Ping and Terminate. They are used to control active servers that have been defined in the Service Directory. The Ping command sends an internal message to verify a server connection. Terminate sends an internal message to terminate a server.

This section covers the following topics:

- Invoking Server Command Execution
- Pinging a Server
- Terminating a Server

## Invoking Server Command Execution

### ▶ To invoke Server Command Execution

- In the Code field of the Client Maintenance screen, enter **XC**.  
The default view of the Server Command Execution screen appears and displays the current definitions as shown in Example 1:

#### Example 1 - Default View

14:09:38	*** NATURAL Remote Procedure Call ***	2002-06-12
	Server Command Execution	SYSRPC
	Node	Server
	Message	
1	ETB045	
2		NRPC001
3		NRPC002
4	*LOCTRAN	
5		NRPC001-LOGICAL
6	LOGBROKER=NODE	
7		NRPC002
<hr/> Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12--- Help ERR Exit < -H +H -P +P TOP BOT> Canc		

The Server Command Execution screen provides two views:

- The default view which displays the columns Node, Server and Message. The fields under the column Message are truncated and display a maximum of 8 characters.
  - The extended view of the fields under the column Message which provide a maximum of 50 characters to display more of the message text. The extended view does not display the column Node and the fields under the column Server are truncated and display a maximum of 16 characters (default view shows 30 characters).
- To activate the extended view of the Server Command Execution screen:
    - Choose PF4.
    - Or, in the command line, enter the less than (<) sign.

A screen is displayed similar to Example 2:

**Example 2 - Extended Message View**

```

14:14:08          *** NATURAL Remote Procedure Call ***          2002-06-12
                   Server Command Execution                      SYSRPC

      Server      Message
1
2      NRPC001      Natural RPC Server 5.1.1.0 on WNT-x86
3      NRPC002
4
5      NRPC001-LOGICAL
6
7      NRPC002

-----
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11--PF12---
      Help  ERR  Exit  <    -H   +H   -P   +P   TOP  BOT>   Canc
    
```

- To switch back to the default view of the Server Command Execution screen:
  - Choose PF11.
  - Or, in the command line, enter the greater than (>) sign.

## Pinging a Server

You can ping a server from the default or extended view of the Server Command Execution screen. The example instruction below shows how to ping a server from the default view.

 **To ping a server**

- In the Code field of the Client Maintenance screen, enter **XC**.  
The default view of the Server Command Execution screen is displayed.
- In the empty column between the sequence number and the column Node, in the line(s) which belong to the server(s) to be pinged, enter the command **PI** as shown in Example 3:

**Example 3 - Pinging a Server**

```

09:49:41          *** NATURAL Remote Procedure Call ***          2002-06-13
                   Server Command Execution                      SYSRPC

      Node      Server      Message
1      ETB045
2      pi      NRPC001
3      NRPC002
4      *LOCTRAN
5      pi      NRPC001-LOGICAL
6      LOGBROKER=NODE
7      NRPC002
    
```

- Choose ENTER.  
The server(s) returns the message: "*Server V.R.S.PL on operating system*", where

*Server* denotes the type of server;  
*V.R.S.PL* is the 1-digit version, *R* the 1-digit release, *S* the 1-digit system maintenance level and *PL* the 1- or 2-digit patch level of the server;  
*operating system* denotes on which operating system the server runs.

Example message: Natural RPC Server 5.1.1.0 on WNT-x86.

- To display more of the message text which appears truncated on the default view of the Server Command Execution screen:
  - Choose PF4.
  - Or, in the command line, enter the less than (<) sign.

## Terminating a Server

You can terminate a server from the default or extended view of the Server Command Execution screen. Below is an example of how to terminate a server from the default view.

### To terminate a server

1. In the Code field of the Client Maintenance screen, enter **XC**.  
The default view of the Server Command Execution screen is displayed.
2. In the empty column between sequence number and column Node, in the line(s) which belongs to the server(s) to be terminated, enter the command **TE**; this is similar to entering the command **PI** as show in Example 3 above.
3. Choose ENTER.

The server returns the message: "Terminating *Server V.R.S.PL* on *operating system*", where

*Server* denotes the type of server;  
*V.R.S.PL* is the 1-digit version, *R* the 1-digit release, *S* the 1-digit system maintenance level and *PL* the 1- or 2-digit patch level of the server;  
*operating system* denotes on which operating system the server runs.

Example message: Terminating Natural RPC Server 5.1.1.0 on WNT-x86.

4. To display more of the message text which appears truncated on the default view of the Server Command Execution screen:
  - Choose PF4.
  - Or, in the command line, enter the less than (<) sign.

If the LOGON option has been set for a server or a node, logon data (user ID, password and library name) are sent to the server with the TE terminate command, as is usual for the CALLNAT. The Security Token Data window pops up to request user ID and password if no Natural Security is installed on the client side and no logon data are set with the USR1071P user exit for the current Natural session.

If LOGONRQ=ON (see also Using Natural RPC with Natural Security in the Natural RPC documentation) has been set on the server side, logon data must be sent from the client with the TE terminate command.

If Natural Security is installed on the server, the logon data transferred must enable a logon to the library SYSRPC.

## Terminating a Server with Replicates

When you are running a server with replicates you must terminate each replicate separately using the TE or Terminate command as described above.

Alternatively, the server can be terminated with the EntireX Broker Control Center or the EntireX System Management Hub.

# SYSRPC - Remote Directory Maintenance

The Remote Directory Maintenance function is used to maintain a remote directory in order to connect the client's calling program to a subprogram on a server.

For further information on how to apply the Remote Directory Maintenance function, refer to Specifying RPC Server Addresses (Operating a Natural RPC Environment), Using a Remote Directory Server (RDS), and Definition of Terms as described in the Natural RPC documentation.

This section covers the following topics:

- Invoking Remote Directory Maintenance
- Fields
- Commands and PF Keys

## Invoking Remote Directory Maintenance



**Warning:**  
**If you create a new RDS directory by entering Code C (see below), the existing file will be overwritten.**

**To invoke and use the Remote Directory Maintenance function**

1. In the Code field of the Client Maintenance screen, enter **RD**.  
 A window appears.
2. Enter **C** to create a directory.  
 Or enter **M** to modify a directory.  
 A window appears.
3. Enter an expiration time in seconds (see also Expiration Time below).  
 An editor screen appears similar to the one below:

```

-----S 01-----Columns 001 072
====>
***** NODE      L T      SERVER      L T      LIBRARY      L T      PROGRAM      L T
***** ***** top of data *****
000001 ETB01      Y B      NRPC2301      SYSTEM      SUB1
000002                                     SUB2
000003                                     NRCP2301      SYSTEM      SUB3
000004                                     NRPC2302      SYSTEM      SUB4
000005                                     SUB5
000006                                     SUB6
000007 ETB01      NRPC2301  Y      SYSTEM      SUB7
000008                                     SUB1          Y
000009                                     SUB2
000010                                     NRCP2301      SYSTEM      SUB3
000011                                     NRPC2302      SYSTEM      SUB4
000012                                     SUB5
000013 ETB01      Y      NRPC2301      SYSTEM      SUB6
000014                                     SUB1          Y
000015                                     SUB2

000016 ETB01      Y      NRPC2301      SYSTEM      SUB3
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      Quit  Save  Find  Chng  Up    Down      Canc
    
```

### Expiration Time

The remote directory data are loaded at runtime. The expiration time determines the period of validity of this data. If directory data are requested after the expiration time set, they will automatically be reloaded. If expiration time is set to 0 seconds, the remote directory data will not be reloaded.

**Note:**

If you create a new directory (Code C), to invoke an input line, enter the line command **I** at the beginning of the line "top of data". See also Line Commands below.

## Fields

The fields contained in the editor screen of the Remote Directory Service maintenance are identical to the fields described in Fields in the section Service Directory Maintenance. The field **L** is the equivalent to the field LOGON.

In addition to the fields provided in the Service Directory Maintenance, the field **T** (Transport) appears: enter **B** for EntireX Broker.

## Commands and PF Keys

Below is information on:

- Line Commands
- Direct Commands and PF Keys

### Line Commands

The line commands available in the editor screen of the Remote Directory Service maintenance are listed below. For further information see Editor Line Commands in the Software AG Editor documentation.

Enter a line command at the beginning of a line, that is, overwrite the sequential number and choose ENTER.

See also To copy or move a block of lines below and the direct command RESET.

Line Command	Function
A	Copies/moves a block of lines marked with C, CC, M or MM (see below) below the line in which the command was entered.
B	Copies/moves a block of lines marked with C, CC, M or MM above the line in which the command was entered.
C	Marks a single line to be copied.
CC	Delimits a block of lines to be copied.
D( <i>n</i> )	Deletes one or <i>n</i> lines beginning with the line in which the command was entered. <i>n</i> can be in the range from 1 to 9.
DD	Marks and deletes a block.  Delimit the block of lines by entering the command in the first and last line of the block, and choose ENTER to execute the command.
I( <i>n</i> )	Inserts one or <i>n</i> empty lines below the line in which the command was entered. <i>n</i> can be in the range from 1 to 9.
M	Moves a single line below the line in which the command was entered.
MM	Delimits a block of lines to be moved.

#### To copy or move a block of lines

1. At the beginning of the line where the block starts, enter CC or MM.
2. At the beginning of the line where the block ends, enter CC or MM.
3. Choose ENTER.  
The lines are marked, the message "Block is pending" occurs.
4. Go to the beginning of the line where you want to place the block and enter **A** (after) to copy or move the block **below** this line,  
Or enter **B** (before) to copy or move the block **above** this line.
5. Choose ENTER.

## Direct Commands and PF Keys

The direct commands and PF Keys available in the editor screen of the Remote Directory Service maintenance are listed below. For further information on direct commands, see Editor Line Commands in the Software AG Editor documentation.

Direct Command	PF Key	Function
RESET		Resets the marks set with the line commands CC, MM, DD (see Line Commands above) or with the direct command CHANGE (see below).
TOP		Scrolls to the beginning of the list.
BOT		Scrolls to the end of the list.
FIND <i>string</i>		Scans the editor for a <i>string</i> of characters, for example: FIND ETB1.  Choose PF5 to scan for the next occurrence.  See also FIND in the section Main Commands in the Software AG documentation.
CHANGE <i>string1 string2</i>		Replaces character <i>string1</i> by <i>string2</i> , for example: CHANGE ETB1 ETB2.  Choose PF6 to replace the next occurrence.  See also CHANGE in the section Main Commands in the Software AG documentation.
	PF1	Help. Invokes the editor online help for the Service Directory
	PF3	Quit. Exits the remote directory service screen screen.
	PF4	Saves modifications.
	PF5	Find. Scans for the next occurrence of the character string defined with the direct command FIND (see above).
	PF6	Change. Replaces the next occurrence of the character string defined with the direct command CHANGE (see above).
	PF7	Up. Scrolls one page backward.
	PF8	Down. Scrolls one page forward.
	PF12	Cancel. Exits the Service Directory screen without saving modifications.

# SYSTP Utility - Overview

The SYSTP utility is used to monitor and control characteristics of Natural that are specific to TP monitors.

This SYSTP utility documentation covers the following topics:

- General Information      Invoking the SYSTP utility and using the SYSTP statistics screens.
- General SYSTP Functions      General SYSTP functions for obtaining statistics on monitoring, print and work files, the swap pool, the usage of Natural buffers during a session, Natural subsystems and roll servers and Natural threads.
- SYSTP Functions under CICS      SYSTP functions specific to a CICS environment.
- SYSTP Functions under IMS/TM      SYSTP functions specific to an IMS/TM environment.
- SYSTP Functions under TIAM and UTM      SYSTP functions specific to TIAM and UTM environments.
- SYSTP in Batch for CICS Sessions      Obtaining statistical data on Natural/CICS sessions in batch mode.

# SYSTP - General Information

This section provides general information on SYSTP utility functions, describes how to invoke SYSTP and lists the commands available on the SYSTP statistics screen.

This section covers the following topics:

- General and Environment-Dependent Functions
  - Invoking SYSTP
  - Using SYSTP Statistics Screens
- 

## General and Environment-Dependent Functions

The SYSTP utility provides functions that are available in most environments, under most TP monitors. They are listed in Invoking SYSTP below and described in General SYSTP Functions.

Additionally, the SYSTP utility provides functions for particular TP monitors. These environment-dependent functions are different for each TP monitor. They are listed in Invoking SYSTP below and described in the corresponding sections of the SYSTP documentation for particular TP monitors. There are no environment-dependent functions under Com-plete and TSO.

## Invoking SYSTP

### To invoke the SYSTP utility

- In the direct command line, enter the system command SYSTP.

The Natural SYSTP Utility Main Menu is then displayed, from which you can select the following functions:

- General SYSTP Functions:
  - Natural Monitoring - SYSMON
  - Natural Print/Work Files - SYSFILE
  - Natural Swap Information
  - Buffer Usage Statistics - BUS
  - Natural Subsystems and Roll Server Information
  - Natural Thread Usage Statistics
- Environment-Dependent Functions (not available under Com-plete and TSO)
  - under CICS
  - under IMS/TM
  - unter TIAM and UTM



## Using SYSTP Statistics Screens

The headers of all SYSTP statistics screens contain the following information:

- The field User with the ID of the current user as assigned by the Natural system variable \*USER.
- The field TID with the terminal ID assigned to the current user by the Natural system variable \*INIT-ID.

On many of the screens accessible from the Natural SYSTP Utility Main Menu, in the leftmost screen column **M**, next to the list item desired, you can enter any of the following line commands:

<b>Command</b>	<b>Function</b>
.	Exit screen and return to previous level.
/	Position line to top of screen.
P	
S	Display detailed information on selected line.

Help information can be invoked for each function by pressing PF1.

If you enter a question mark (?) in the command line, all direct commands available within the SYSTP utility are displayed.

# General SYSTP Functions

This section describes the SYSTP functions which are available under most TP monitors:

- Natural Monitoring - SYSMON
  - Natural Print/Work Files - SYSFILE
  - Natural Swap Information
  - Buffer Usage Statistics - BUS
  - Natural Subsystems and Roll Server Information
  - Natural Thread Usage Statistics
- 

## Natural Monitoring - SYSMON

This function provides statistics related to Natural programs and screen transactions of Natural sessions.

When you invoke this function and the monitoring function is **not active**, a menu is displayed from which you can select the following function:

- Activate Monitor

When you invoke this function and the monitoring function is **active**, a menu is displayed from which you can select the following functions:

- Deactivate Monitor
- Display Monitor Terminal Statistics
- Display Monitor Program Statistics

### Activate/Deactivate Monitor

With these functions you can activate or deactivate the monitor function.

When the monitor function is activated, it begins collecting statistical information of current sessions. Once the monitor function is deactivated, a statistical summary is collected and written to the system log file.

**Note:**

When active, the monitoring function requires additional memory pool space and therefore may affect overall system performance. Set the RDCSIZE parameter to a minimum value of 2 KB. If no monitor buffer pool has been created, additionally, you must set the MONSIZE parameter. For a description of these parameters, refer to the relevant sections in Profile Parameters in the Natural Parameter Reference documentation

## Display Monitor Terminal Statistics

Terminal statistics can be displayed for all active terminals or for a single terminal.

- For all active terminals: enter **T** in the Code field.
- For a single terminal: enter **T** in the Code field and the terminal ID in the field "Name of LTERM or Program".

The following screen is used to display statistics for all active terminals:

```

17:03:13          ***** NATURAL SYSTP UTILITY *****          2000-11-27
User VR000001    - NATURAL Monitor Terminal Statistics -          TID 0756

Cm Name      Current      NAT- ADA- Ext- Mean-  Screen I/O  User   Sys  Fetch
              time time time  time   No  KB   Acc  Acc
-----
_ 0756      S2SCENT1          0   0   0   0.0    1   0    0    0    5
              0   0   0   .0    0   0    0    0    0
              0   0   0   .0    0   0    0    0    0
              0   0   0   .0    0   0    0    0    0
              0   0   0   .0    0   0    0    0    0
              0   0   0   .0    0   0    0    0    0
              0   0   0   .0    0   0    0    0    0
              0   0   0   .0    0   0    0    0    0
              0   0   0   .0    0   0    0    0    0
              0   0   0   .0    0   0    0    0    0
-----
Function : _          ( + next page / . Exit / ? Help )
-----
Select, mark with function or mark for additional information
Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Menu Exit          +          Canc
    
```

**Note:**

If the overview of active terminals is displayed repeatedly, an asterisk (\*) is set to the terminal most active since the last repetition.

To display statistics for a single terminal, you mark the desired terminal on the screen above.

The following statistics are provided for terminals and programs:

Column	Statistics	Explanation
NAT-time	Time in Natural	Time in Natural nucleus and in the interface.
ADA-time	Time in Adabas	Time waiting for response from Adabas.
Ext-time	Time in external program	Time needed by a user-written module.
Mean-time	Mean evaluation time	Elapsed time of one Natural screen transaction.
Screen I/O No	Number of Screen I/Os	Number of screen I/Os.
Screen I/O KB	Amount of data transmitted	Amount of data transferred to or from the screen.
	Evaluation time > 3 sec	Only applies to statistics for a single terminal. Percentage of evaluation times longer than 3 seconds.
	Evaluation time > 6 sec	Only applies to statistics for a single terminal. Percentage of evaluation times longer than 6 seconds.
User Acc	Number of user file accesses	Counter for accesses to Adabas user files.
Sys Acc	Number of system file accesses	Counter for accesses to Natural system file, including fetches.
Fetch	Number of fetches	Counter for total number of fetches.

## Display Monitor Program Statistics

Program statistics can be displayed for all active programs or for a single program:

- For all active programs: enter **P** in the Code field.
- For a single program: enter **P** in the Code field and the program name in the field "Name of LTERM or Program", and the library name.

The following screen is used to display statistics for all programs:

```

08:56:53          ***** NATURAL SYSTP UTILITY *****          2000-11-28
User VR000001    - NATURAL Monitor Program Statistics -          TID 0807

Cm Name      Current      NAT-  ADA-  Ext-  Mean-  Screen I/O  User  Sys  Fetch
              time time time  time   No  KB   Acc  Acc
-----
_ SMMMEN01 SYSTP          0   0   0  0.0    1   0    0    0    2
_ S2MRAHM1 SYSTP          0   0   0  0.0    0   0    0    0    0
_ S2SCOM01 SYSTP          0   0   0  0.0    0   0    0    0    0
* SMPMEN01 SYSTP          0   0   0  0.0    0   0    3    0    1
_ SMPSTA01 SYSTP          0   0   0  0.0    0   0    0    0    1
_ S2SCENT1 SYSTP          0   0   0  0.0    0   0    0    0    0
              0   0   0  .0     0   0    0    0    0
              0   0   0  .0     0   0    0    0    0
              0   0   0  .0     0   0    0    0    0
              0   0   0  .0     0   0    0    0    0
-----
Function : _          ( + next page / . Exit / ? Help )
-----
Select, mark with function or mark for additional information
Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Menu Exit          +          Canc
    
```

For an explanation of the screen output, see the table above.

**Note:**

If the overview of active programs is displayed repeatedly, an asterisk (\*) appears in the **Cm** column next to the program most active since the last repetition.

To display statistics for a single program, you mark the desired program on the screen above.

## Natural Print/Work Files - SYSFILE

This function provides information on the available work files and print files.

You can also invoke this function with the system command SYSFILE (Natural System Command Reference documentation).

This function can also be used in batch mode for CICS sessions.

When you invoke this function, the Work File Information screen appears with a list of all work and print files defined. The following information is provided for each file:

Column	Explanation
No.	The number of the work/print file.
Type	The type of assignment; that is, the operating system, TP monitor or Natural product file to which the work/print file is assigned.
Name	The name of the work/print file.
Recfm, Lrecl, Blksz	The record format, logical record length and block size of the work/print file (if applicable).
Status	The status can be:  available for input and/or output,  open for input and/or output.

Under VSE/ESA, the logical-unit assignments are also displayed.

With the following keys and commands (which you enter in the first column of the list), you can scroll the list or display additional information:

Key	Command	Function
PF4	S	Displays various additional items of information on the file marked with the cursor/command.
PF5	P	Scrolls the file marked with the cursor/command to the top of the page.
	/	
PF6		Scrolls to the beginning of the list.
PF9		Scrolls to the end of the list.
PF7		Scrolls one page backward.
PF8		Scrolls one page forward.
PF10		Scrolls to the list of print files.
PF11		Scrolls to the list of work files.
	D	Displays the corresponding Natural control block (work file area) in dump format.

## Natural Swap Information

This function is only available under CICS and UTM.

The swap pool manager enables online monitoring and control of the Natural swap pool. This section describes how the swap pool manager is used rather than how the swap pool operates. For detailed information on the operation of the Natural swap pool, see Natural Swap Pool in the Natural Operations for Mainframes documentation.

From the swap pool manager main menu, you invoke the following functions and sub-functions for controlling and monitoring the swap pool:

- Administration
- Debugging Facilities
- Information
- Parameter Service
- Status Information

Each of these functions can be invoked by either function code or PF key.

## Administration

- Slot Size Calculation
- Change Swap Pool Status
- Update Reorg Control Data

### Slot Size Calculation

This function displays the optimum values for the layout of the swap pool based on the current usage.

You can store these values to be used for a later initialization/reorganization (once they have been stored, they can also be maintained with the Parameter Service function).

You can also initiate a swap pool reorganization using these values.

For further details, see the online help of this function.

### Change Swap Pool Status

This function is used to activate or deactivate the Natural swap pool. In addition, you can modify the wait time and the number of waits for swap pool synchronization.

For further details, see the online help of this function.

### Update Reorganization Control Data

With this function, you can modify the most important parameters in swap pool management. To modify the values you must enter a valid password.

For further details, see the online help of this function.

## Debugging Facilities

### Note:

Do not use this function without prior consultation of Software AG's Natural support.

This function is only available under UTM

With this function it is possible to activate or deactivate an internal screen debugging buffer. Activation of the screen debugging buffer is used to locate terminal I/O inconsistencies if they occur. The function records information on the last three terminal I/O sequences. The buffer has a size of 3 KB and is used in a wrap-around procedure.

In addition, you can activate/deactivate a trace function for asynchronous write operations to the Natural roll file.

For further details, see the online help of this function.

## Information

- Show Addresses
- Show Summary of Buffer Usage
- Show Swap Pool Information
- Show Logical Swap Pools
- Show Reorg Control Data
- Show Swap Pool Usage
- Create Statistics List

### Show Addresses

This function displays the addresses of various pools.

### Show Summary of Buffer Usage

This function is used to optimize the sizes of the various Natural buffers and the Natural user threads (MAXSIZE). It activates, deactivates and displays a summary of Natural buffer usage.

The activation and deactivation of buffer statistics can only be performed with a valid password. For the display of buffer statistics, no password is necessary.

The buffers displayed are the same as those displayed by the function Buffer Usage Statistics.

### Show Swap Pool Information

This function displays information on the swap pool currently in use, including control/statistics data, and memory sizes.

The individual items of information shown are explained in the online help of this function.

### Show Logical Swap Pools

This function displays the current table of logical swap pools.

On the table, you can mark a specific logical swap pool with any character to get additional information on it.

The individual items of information shown are explained in the online help of this function.

### Show Reorganization Control Data

This function displays all information related to the swap pool reorganization.

Displayed in the left half of the screen is the swap pool reorganization table. The table contains cumulative statistics on the comparative sizes between compressed Natural user threads and standard slot size. The table is cleared with each reorganization of the swap pool. The left half of the table shows how often and to what extent the user threads are larger than the standard slot size. The right half of the table shows how often and to what extent the user threads are smaller than the standard slot size. Sizes in this half of the table are expressed in units that are dependent on the factor specified by the swap pool manager.

In the row labeled **n**, count is taken of user threads which exceed/fall short of the standard slot size by over 9 pages/units. The average length of these user threads is displayed in the row labeled **Av.+n**.

The individual items of information shown are explained in the online help of this function.

### Show Swap Pool Usage

This function displays information on the usage of the swap pool since its initialization or the last reorganization.

The individual items of information shown are explained in the online help of this function.

### Create Statistics List

This function is used to create a list of the current swap pool usage statistics.

## Parameter Service

- Parameter Maintenance
- Password Maintenance

### Parameter Maintenance

This function is used to change online the parameters for the initialization or reorganization of the swap pool.

The subfunctions as well as the individual items that can be modified are explained in the online help of this function.

The use of this function is password-protected (see below).

### Password Maintenance

This function is used to change or recover the password used for the Parameter Service function.

The initial password is SYSTP.

### Status Information

With this function, you can display the current status of the Natural swap pool, of the summary of buffer usage and of the UTM screen debugging.

## Buffer Usage Statistics - BUS

This function provides statistical information on the usage of Natural buffers: which buffers are allocated for the current session, and how much buffer space is being used.

The Total figures at the end of the statistics list allow you to draw conclusions about the efficiency of buffer compression.

You can invoke this function either from the SYSTP menu or with the system command BUS.

When you invoke the function, a list is displayed showing all buffers which are actually being used in the current Natural session.

For each of these buffers, the following information is displayed on the Buffer Usage Statistics screen:

Column	Explanation
M	In this column, you can mark a buffer with a command (see below).
No.	The buffers are numbered sequentially in order of allocation.
Name	The name of the buffer. Only those buffers which have actually been requested in the current Natural session are listed.
Type	<b>V</b> indicates a variable buffer. The size of a variable buffer is increased automatically when necessary (even if it is allocated outside the Natural thread). If it is allocated outside the thread, it is copied into the thread at a terminal I/O; if it does not fit into the thread, it is truncated to its actually used length.
Size	The size of the buffer (in bytes).
Used	The number of bytes currently being used. This value is used for buffer compression in environments using threads (for example, CICS or UTM).
Perc. (Used)	The percentage currently being used; that is, the value of the Used column in relation to the value of the Size column.
MaxUsed	The maximum number of bytes which have been used in the course of the current session so far ( <b>not</b> the size being used at present).
Perc. (MaxUsed)	The percentage of current session usage; that is, the value of the Max. Used column in relation to the value of the Size column.
MaxSize	The maximum size (in bytes) that has been allocated to the buffer in the course of the current session so far (applies to variable buffers only).
Perc. (MaxSize)	The maximum size allocated so far (value of the Max. Size column) in relation to the current size (value of the Size column) (applies to variable buffers only). A percentage of 1000 or more is indicated by <b>999.9</b> displayed intensified.
<b>At the end of the list, the following information is displayed:</b>	
ThrdSize	The current size (in KB) of the Natural thread.
Total	The sums of all buffer sizes (in both bytes and KB) and percentages used/allocated. These totals can also be displayed via PF10 (see below).  For MaxSize, the total shows the maximum additional amount of thread size that would have been needed in the course of the session so far.

With the following keys and commands (which you enter in the first column of the list), you can scroll the list or display additional information:

Key	Command	Function
PF4	D	Displays the contents of the buffer marked with the cursor/command in dump format (for internal use by Software AG support personnel).
PF5	P	Scrolls the buffer marked with the cursor/command to the top of the page.
	/	
PF6	--	Scrolls to the beginning of the list.
PF7	-	Scrolls one page backward.
PF8	+	Scrolls one page forward.
PF9	++	Scrolls to the end of the list.
PF10		Displays the Total buffer usage figures.
PF11		Displays the relative addresses of the buffers, that is, relative to the input/output control buffer (IOCB).

## Natural Subsystems and Roll Server Information

This function is only available under OS/390.

You can use it to determine an optimum thread size or roll file size for a Natural application. It displays a list of the Natural subsystems together with the current status of the related authorized service manager and roll server.

The following commands are available for each listed subsystem:

Command	Function
B	Displays buffer pool information (name, size, type).
R	Displays roll server statistics.
S	Displays Zaps applied to the authorized service manager.
Z	Displays Zaps applied to the roll server.
L	Displays and resets entries in the roll file directory.

This information is useful for tuning the roll server, as described under Roll Server in the Natural Operations for Mainframes documentation.

## Natural Thread Usage Statistics

This function is only available under CICS, Com-plete, IMS/TM and UTM. It is not available in a Sysplex environment.

This function allows you to determine an optimum thread size or roll file size for a Natural application.

You should activate this function only when needed, and deactivate it after you have determined your optimum thread size, because this function occupies space in the Natural buffer pool. When you deactivate it, the space in the buffer pool becomes available again.

 **Proceed as follows:**

1. Define an oversized thread in the range of 512 to 1024 KB for your Natural application. Take into account the number of Software AG subproducts used.
2. Start your Natural application, either in production or in test mode.
3. Activate the Natural Thread Usage Statistics function: Invoke the SYSTP utility. On the SYSTP main menu, choose Function T (Natural Thread Usage Statistics). On the menu that appears then, choose Function A (Activate Statistics).
4. Use your Natural application under typical production conditions. The Thread Usage Statistics function runs in the background and logs the buffer sizes used.
5. Then invoke the SYSTP Thread Usage Statistics function again. On the menu that appears, choose Function S (Show Statistics), P (Print Statistics) or D (Deactivate and Print Statistics). It is recommended that you use Function D to free buffer pool space.

The following information is displayed on the Natural Thread Usage Statistics screen:

<b>Column</b>	<b>Explanation</b>
No.	The buffers are numbered sequentially in order of allocation.
Ext. Buffer	The sizes of these buffers are defined externally (in the Natural parameter module).
Defined Size	The buffer size as defined in the Natural parameter module.
Max. Allocated Size	The maximum buffer size allocated. Note that for the internal BB area, 14368 bytes are added to the ESIZE profile parameter value.
Max. Used Size	The maximum buffer size used.
Sum of external buffer sizes	The total of all buffer sizes defined in the Natural parameter module.
Sum of internal buffer sizes	The total of all buffer sizes requested by Natural internally.
Max. used thread length	The maximum thread length used by Natural. Define this length as your minimum (optimum) Natural thread length.  Round it up to the next KB number that can be divided by 2.
Max. compressed thread length	The maximum length of a compressed Natural thread that was written to the Natural roll file.  Define this length as your minimum (optimum) Natural roll file length.

## Show Physical GETMAIN Statistics

The physical GETMAIN statistics provide information on all physical GETMAINs relevant for the Natural work pools and the variable Natural buffers outside the Natural user threads.

They indicate the original buffer sizes (during the startup of a Natural session), the number of physical GETMAINs, the buffer length for the physical GETMAIN and the buffer position (above or below the 16-MB line).

The statistics data always refers to the buffers with the greatest lengths requested within a terminal I/O, for all users of the Natural application.

The statistics provides a maximum of six entries for each buffer. These entries may be overwritten through the wrap around procedure. The highest number equals the maximum number of the physical GETMAINs within a terminal I/O, for each buffer concerned.

The first two entries in the statistics refer to the Natural work pools (if available) above (WRKPOOLA), respectively, below (WRKPOOLB) the 16-MB line.

Here, the highest physical GETMAIN number refers to the amount of work pools simultaneously available during the terminal I/O. The sum of all work pool lengths amounts to the total storage requirement of the work pools within a terminal I/O.

All subsequent statistics entries refer to the physical GETMAINs for the variable Natural buffers, which either could not be defined in the Natural user thread due to insufficient space, or were increased outside the Natural user threads. For these buffers, the highest physical GETMAIN number indicates the greatest space requirement for each buffer within a terminal I/O. The total storage space requested earlier was freed before each of the following physical GETMAINs.

That is, the sum of all physical GETMAINs with the highest number shows the maximum storage requirement for the variable buffers outside the Natural user threads during a terminal I/O, for all users of the Natural application.

# SYSTP Functions under CICS

This section provides information on the SYSTP utility functions available under CICS.

## To invoke the SYSTP functions under CICS

- On the Natural SYSTP Utility Main Menu, in the Code field, enter **E** for Environment-Dependent Functions.

On the menu displayed then, you can select the following functions which are explained in this section.

- Natural User Sessions
- Natural Roll Facilities
- Natural Thread Groups
- Natural Storage Threads
- NCI Global System Information
- NCI Generation Options
- Natural Thread Group Definitions
- Own Natural User Session
- CICS Task Information
- System Administration Facilities
- Applied NCI Zaps

### Note:

In the remainder of this section, the Natural CICS Interface is also referred to as NCI.

---

## Natural User Sessions

This function is used to display a list of active user sessions in a Natural environment.

When you invoke this function, the Natural User Sessions screen appears displaying the following information:

Column	Explanation
Term ID	Unique terminal ID within CICS associated with the Natural session.
User ID	Natural user ID of the Natural session.
Tran	CICS transaction ID under which Natural session is currently running. For pseudo-conversational sessions, this is the pseudo-conversational restart transaction ID.
Start Date/Time	Starting date and time of the Natural session.
Last Act	Time of last screen output.
Stat	Session status: see Operational Status below.
Program	Natural program currently active.
Library	Natural library in which the user is currently working.

If you press PF10, the display of the session date and time is replaced by the following session resource data:

Column	Explanation
Thrd Grp	Thread group to which user is assigned.
Thread	Name of thread last used.
Roll Fac	Assigned roll facility.

The sections below cover the following information:

- Commands for Natural User Sessions
- Natural User Session Statistics

## Commands for Natural User Sessions

For each item displayed on the Natural User Sessions screen, you can execute any of the commands described below.

### To execute commands for list items

- In the **M** column, position the cursor at the input field(s) next to the item(s) desired, and enter any of the line commands described below.

Command	Function
C	<p>Cancel session.</p> <p>Invokes a confirmation window where you can enter Yes to mark a session for termination. The session selected is then flagged with number (#) signs that appear in the column User ID. Additionally, for the session concerned, the operational status "Purged by Admin" appears on the Natural User Session Statistics screen (see below) of the user (administrator) who executed the cancel command.</p> <p>The session actually terminates when the owner of the session marked for termination performs the next terminal I/O, Adabas call or external program call. The session owner then receives receives a corresponding termination notification.</p>
F	<p>Flush session.</p> <p>Invokes a confirmation window where you can enter Yes to terminate a session immediately. The session terminated is then flagged with number (#) signs that appear in the column User ID. The user (administrator) who terminated the session receives a termination message when trying to invoke the Natural User Session Statistics screen (see below) for the session terminated but still listed on the Natural User Session screen. The session owner receives a corresponding termination notification.</p>
R	<p>Reactivate session.</p> <p>Reverses a <b>C</b> (Cancel) command as described earlier. The <b>R</b> command removes the termination flags set for a session and resets the session status to active.</p> <p>Note that you cannot reactivate a session that has been terminated with the <b>F</b> (Flush) command described above.</p>
S	<p>Invoke the Natural User Session Statistics screen with additional information on the individual user session(s) selected as described below.</p>
U	
W	<p>Wake up session.</p> <p>Reactivates immediately a session that has been suspended by a CMROLL call with a non-zero wait interval specified. See also the example program SUSPEND supplied in the Natural system library SYSEXTP.</p>

► **To limit the number of list items by specifying selection criteria**

- On the Natural User Sessions screen, press PF4.

The Selection for User Sessions window appears where you can select user sessions by inactivity date and time, inactivity time interval, Natural server ID, CICS system ID, terminal ID, user ID or transaction ID.

Inactivity date and time and inactivity interval list all sessions that were not active before the date and time specified or before the time interval specified. A date must be specified in the format *YYYY-MM-DD* (*YYYY* = year, *MM* = month, *DD* = day). The time must be specified in the format *HH:II:SS* (*HH* = hours, *II* = minutes, *SS* = seconds).

To specify a range of IDs:

Use the asterisk (\*) as a leading or trailing character or use the question mark (?) as a wildcard character, for example:

AB\* , \*BC, A?B

## Natural User Session Statistics

For each session displayed in the Natural User Sessions screen shown above, additional information can be displayed by invoking the Natural User Session Statistics screen. The following section describes the information provided on this screen.

**Note:**

All sizes on the Natural User Session Statistics screen are in KB unless otherwise indicated in the field descriptions below.

Field	Explanation
Started	Day, date and time when the session was started.
Last Actions	Date and time the user was active last.
User	Natural user ID as assigned by the Natural system variable *USER.
at Terminal	ID of the terminal associated with the Natural session as assigned by the Natural system variable *INIT-ID.
Transid	(Pseudo-conversational) transaction ID under which Natural is running.
Task # in	Task number assigned by CICS followed by the ID of the CICS region.
Cur Strg Used	Current amount of storage used by this session.
Max Strg Used	Maximum amount of storage ever used by this session.
Thread Size	Size of this thread.
Thread Name	Name of the thread used last. For threads allocated by using GETMAIN, the thread name is composed of the prefix NSCP followed by the terminal ID.
Common Thread Size	This group's common thread size.
Thread Group	Name of the associated thread group (triggered by starting the transaction ID).

Field	Explanation
of Type	Type of thread used for thread group:  SHR Permanent storage threads.  GETM Storage threads allocated by using GETMAIN.  NONE No threads used; all Natural storage requests are passed to CICS.
Natural Library	Natural library ID as assigned by the Natural system variable *LIBRARY-ID.
Natural Program	Name of the Natural program currently used by the session as assigned by as assigned by the Natural system variable *PROGRAM.
Line No.	Line number in the Natural program currently used by the session.
Operational Status	See Operational Status below.
Roll Facility	Name of associated roll facility.
Roll Recs (Last)	Number of records written to roll facility for last roll-out.
Roll Recs (Max)	Maximum number of records ever written during roll-out.
Roll Record Size	Record size of this roll facility.
Slot Size	Number of records required to roll-out a thread completely.
Restart Rec. No.	Number of the record that contains roll-out control information; this record must be rolled in first.  <b>VSAM Roll Files</b> The following information applies to VSAM roll files only: The relationship between restart record number (RecNum), slot number (SN) and slot size (SZ) is:  RecNum = (SN-1) * SZ + 2 or SN = (RecNum-2) / SZ + 1
Slot Number	Number of slot in VSAM roll file belonging to this session (for VSAM only). See also VSAM Roll Files in Restart Rec. No. above.
Compressed Length	Amount of relevant storage currently swapped/rolled out.
Session Resumes	Total number of session resumes.
Swap-Ins	Number of session resumes with swapping in from swap pool.
Thread Switches	Number of session resumes with swapping/rolling into a thread which is different to the one the session had been in before.
Roll-Ins	Number of session resumes with rolling in from roll facility.
Region Switches	Number of CICS region switches.
OpSys Switches	Number of operating system image switches in a Sysplex environment.

**Operational Status**

This field indicates any of the following operational statuses:

Status	Abbreviation	Description
Active	Act	Currently active.
Inactive	Ina	Inactive, still in thread.
Swapped	Swp	Swapped, in swap pool.
Rolled out	Rld	Rolled out, in roll facility.
Wait (Init)	WtI	Waiting for thread on session initialization.
Wait (Resume)	WtR	Waiting for thread on session resume.
Initializing	Int	Initializing session.
Resuming	Res	Resuming session, in thread, not active yet.
Suspending	Sus	Suspending session.
Terminating	Trm	Terminating session.
Swapping out	Swo	Session swapping out.
Swapping in	Swi	Session swapping in.
Rolling out	Out	Rolling out from thread or swap pool.
Rolling in	In	Rolling in from roll facility.

The following additional information can appear in Operational Status:

Status	Description
Conversational	Dialog-oriented session (PSEUDO=OFF) as opposed to pseudo-conversational/transaction-oriented session.
Forced Conversational	Last screen I/O of a PSEUDO=ON session was conversational.
No-Roll	Session is not allowed to roll.
Compressed	Session is compressed (in swap pool or roll facility).
Thread Switched	The thread currently used is not the same as used before.
Thread Locked	Session kept from switching threads (for example, RELO=OFF); may also force No-Roll/Conversational status.
Purged by Admin	Session canceled by administrator (flag set).
Spool Task	The task is a spool/print task.
Asynchronous Task	The task is an asynchronous task, not bound to a terminal.

## Natural Roll Facilities

This function is used to display which swap files are available for rolling out user work areas to make room in the swap pool for active users. These swap files are known as roll facilities.

When you invoke this function, the Natural Roll Facilities screen appears for the current CICS region (as indicated by the CICS ID in the screen title). For each roll facility, the following information is displayed:

Column	Explanation
Facility Name	TEMPSTOR is used for auxiliary temporary storage, MAINSTOR for main temporary storage, and remaining file names are VSAM roll files as defined in the CICS file control table (FCT).
Record Size	Record Size of this roll facility.
Slot Size	Number of records required to roll out a thread completely (maximum thread size divided by record size, rounded up).
No. of Slots	Number of sessions which fit into this roll file (number of file records divided by slot size, rounded down); applies to VSAM roll files only.
Facility Users Cur/Max	Current (Cur) and maximum (Max) number of user sessions assigned to this roll facility.
Roll Counts Out/In	Number of session roll operations from or into this roll facility.
Status	Indicates Full if the facility users equal the number of available slots.

 **To display all Natural user sessions that use a roll facility**

- In the **M** column, position the cursor at the input field(s) next to the roll facility (facilities) desired, and enter the line command **U** or press PF4.

The Natural User Sessions screen appears as described in the relevant section.

## Natural Thread Groups

This function is used to display which thread groups are available to Natural.

When you invoke this function, the Natural Thread Groups screen appears for the current CICS region (as indicated by the CICS ID in the screen title). For each thread group, the following information is displayed on this screen:

Column	Explanation
Group Name	Thread group name.
Group Users Cur/Max	Current (Cur) and maximum (Max) number of users assigned to this thread group.
Thread Type	Type of thread used: see Natural User Session Statistics above.
TCBs	Maximum number of sessions concurrently active.
Thread Size	Thread group's common thread size.
Strg Used	Maximum storage allocated to any thread in this group.
Queue Sizes Cur/Max/atMax	Current (Cur) and maximum (Max) queue size for the thread group's central wait queue and the number of times the maximum was reached (atMax).  Only applies if the parameter THREADS has been defined as greater than zero for this thread group.
VSAM/Aux/Main	Roll facilities defined for group; CICS temporary storage (auxiliary or main) always backs up VSAM if VSAM roll files are not available or full.

 **To display additional information on a thread group**

- In the **M** column, position the cursor at the input field(s) next to the item(s) desired, and enter any of the following line commands or press the corresponding PF key:

Command	PF Key	Function
D	PF11	Display Natural thread group definitions (see the relevant section).
T	PF10	Display Natural storage threads (see below) in thread group.
U	PF4	Display all Natural user sessions (see the relevant section) that use the thread group selected.

## Natural Storage Threads

This function is used to display information on the storage threads in the Natural environment.

When you invoke this function, the Natural Storage Threads screen appears for the current CICS region (as indicated by the CICS ID in the screen title). The screen displays the following information:

Column	Explanation
Thread Name	Name of the thread.
Grp No.	Number of the group to which this thread belongs.
Thrd Size	Usable thread size.
Strg Used	Maximum amount of storage ever used in this thread.
Use Count	Number of times this thread has been selected for processing.
Roll-Ins Log./Phys.	Logical (Log.): Session resumes. Physical (Phys.): Roll-in from roll facility.
Queue Sizes Cur/Max/atMax	Current (Cur) number of users queuing on thread. If this number <i>n</i> is greater than 1, <i>n</i> minus 1 users are waiting; maximum (Max) queue count for this thread; number of times at maximum (atMax).
Term ID	Terminal ID belonging to the Natural session whose data are in thread.
Task No.	ID of CICS task currently active in this thread. If no ID is displayed, no session is active in this thread.

 **To display additional information on Natural storage threads**

- In the **M** column, position the cursor at the input field(s) next to the item(s) desired, and enter any of the following line commands or press the corresponding PF key (if available):

Command	PF Key	Function
C		See "Cancel session" in Natural User Sessions.
D	PF11	Display Natural thread group definitions (see the relevant section).
F		See "Flush session" in Natural User Sessions.
G	PF10	Display Natural thread group (see the relevant section).
R		See "Reactivate session" in Natural User Sessions.
U	PF4	Display statistics on the Natural user sessions currently active in the thread. See also Natural User Session Statistics.

## NCI Global System Information

This function is used to display data on the system directory.

When you invoke this function, the Global System Information screen appears for the current CICS region (as indicated by the CICS ID in the screen title). The screen provides the following information:

Field	Explanation
Natural User Sessions	Current and maximum number of Natural sessions in the system.
Concurrent SCP Active	Current and maximum number of concurrent system control program (SCP) requests (INIT/SUSP/RESM/TERM).
SIR Block Extensions	Current and maximum number of local SIR block extensions.
Slots in 1st SIR Block	Number of user sessions that fit into the primary user control block (first USERS subparameter in NCMDIR macro).
Slots in SIR Block Extns.	Number of user sessions that fit into a secondary user control block (second USERS subparameter in NCMDIR macro).
VSAM Roll File Slots	Number of VSAM roll files to check (ROLLFLS).
Possible Roll Facilities	Number of VSAM roll files plus two for CICS TEMPSTOR.
Thread Groups	Number of thread groups determined by evaluating all NCMTGD macro specifications at system startup.
System Recoveries	Number of corrections of statistics counts and/or control block chain.
Size of DIR Extension (B)	Number of bytes used at system startup for thread control blocks and VSAM roll file online directories.
Operating System Host ID	Name of the operating system image.
CICS System ID	The ID of the CICS region.

Field	Explanation
Available Resources:  Swap Pool Local Buff. Pool Sort Buffer Pool DL/I Buffer Pool Edit Buffer Pool Monitor Pool	Resources available in the current NCI system environment: swap pools, Natural buffer pools and monitor buffer pools.  Type, size (in KB) and location (below or above the 16 MB line) of all buffer pools supported.
Max Thread Size	Largest thread size across all valid thread groups.
VSAM Roll Files	Indicates whether VSAM roll files are available; that is, existing in VSAM, formatted and defined in the CICS FCT.
Main/Aux TempStor	Indicates whether CICS main or auxiliary temporary storage is available for the Natural/CICS roll facilities.
Session Logging	Indicates whether the Natural/CICS log destination is defined in the CICS DCT (destination control table) and whether the log destination is available. The log destination for sessions is defined with the (*) LOGDEST parameter of the NCMPRM macro.
Message Logging	Indicates whether the Natural/CICS error message log destination is defined in the CICS DCT and whether the log destination is available. The log destination for messages is defined with the (*) MSGDEST parameter of the NCMPRM macro.
Message Switching	Indicates whether the message switching transaction ID is defined in CICS and whether the transaction ID is available. The transaction for switching messages is defined with the (*) MSGTRAN parameter of the NCMPRM macro.  <b>Note:</b> If this transaction ID is not available, a SYSTP session flush (see "Flush session" in Natural User Sessions) is not possible.
Trace Active	Indicates whether the Natural/CICS trace function is currently active; see also System Administration Facilities.
Assembled Last	Date and time when the system directory source module was last assembled.

\* The NCMPRM macro parameters LOGDEST, MSGDEST and MSGTRAN are described in Natural CICS Generation Parameters, Natural under CICS, in the Natural TP Monitor Interfaces documentation.

## NCI Generation Options

This function is used to display generation parameter settings for Natural running under CICS. The values of these parameters are determined in the macro NCMPRM, which is part of the Natural/CICS parameter module created during installation.

When you invoke this function, the Generation Options screen appears for the current CICS region (as indicated by the CICS ID in the screen title). This screen displays an overview of the generation option settings for Natural.

Behind each parameter setting in the Generation Options screen is a parameter of the NCMPRM macro. These parameter names can be viewed by pressing PF10. Use PF10 to toggle between the screen containing the parameter names and explanations of the parameters.

#### Related Topics:

Installing the Natural CICS Interface - Natural Installation Guide for Mainframes

NCMPRM Macro Parameters - Natural CICS Generation Parameters, Natural under CICS, Natural TP Monitor Interfaces documentation.

## Natural Thread Group Definitions

This function is used to display Natural thread group definitions.

When you invoke this function, the Natural Thread Group Definitions screen appears for the current CICS region (as indicated by the CICS ID in the screen title). This screen displays the following information:

Column	Explanation
Grp No.	Thread group number.
Group Type	Type of group definition:  SHR     Permanent storage threads to be used for thread group.  GETM   Storage threads allocated by using GETMAIN.  NONE   No threads to be used; all Natural storage requests are passed to CICS.  ALIAS   Thread group redefinition to assign other primary roll facility triggered by transaction ID/task request key.
Roll Fac.	Primary roll facility assigned: VSAM, TEMPSTOR, MAINSTOR or "none".
Thread Size	Thread storage GETMAIN size (for thread group types GETM and SHR).
No. Thrds.	Maximum number of Natural sessions concurrently active in this thread group.
Transaction IDs	As defined in CICS Transaction Definition.
Task Request Keys	As defined in CICS Transaction Definition.

#### To display additional information on a thread group definitions

- In the **M** column, position the cursor at the input field(s) next to the item(s) desired, and enter any of the following line commands or press the corresponding PF key:

Command	PF Key	Function
G	PF10	Display Natural storage threads (see the relevant section) associated with the thread group.
S	PF4	Display thread group definitions for current thread group only.

## Own Natural User Session

This function invokes the Natural User Session Statistics screen described in Natural User Session Statistics.

## CICS Task Information

This function invokes the SYSTP Task Information window which displays status information on the Natural task in a CICS environment.

## System Administration Facilities

This function is used to access facilities for debugging and tracing.

When you invoke this function, a menu is displayed which offers you the following functions:

- Trace Facilities  
(reserved for internal use by Software AG only)
- Debugging Facilities  
(reserved for internal use by Software AG only)
- System Snapshot for Logging
- Reset System Highwater Marks

The section below contains information on:

- System Snapshot for Logging
- Reset System Highwater Marks

### System Snapshot for Logging

This function provides complete SYSTP batch reports (see also SYSTP in Batch for CICS Sessions) with information on all SCP facilities, regardless of whether they have been used or not. Such facilities are:

- thread groups,
- TYPE=SHR threads,
- roll facilities.

All this information is logged to the Natural/CICS log file (if available).

### Reset System Highwater Marks

This function comprises the system snapshot function previously described. In addition, all system highwater marks can be reset, for example:

- The number of user sessions,
- Every thread group and roll facility,
- The number of UCB block extensions,
- The amount of storage,
- All thread groups and TYPE=SHR threads,
- All wait queue values and counts,
- All roll facility roll counts.

## Applied NCI Zaps

This function invokes the Applied NCI Zaps screen for the current CICS region (as indicated by the CICS ID in the screen title). This screen displays the numbers of all Zaps that have been applied to the current Natural TP environment.

# SYSTP Functions under IMS/TM

This section provides information on the SYSTP utility functions available under IMS/TM.

## To invoke the SYSTP functions under IMS/TM

- On the Natural SYSTP Utility Main Menu, in the Code field, enter **E** for Environment-Dependent Functions.

On the NII Menu displayed then, you can select the following functions which are explained in this section:

- Broadcasting
- Display Environment Data
- Monitoring
- Multi Session
- Applied NII Zaps

### **Note:**

In the remainder of this section, the Natural IMS/TM Interface is also referred to as NII.

---

## Broadcasting

This function is used to broadcast messages to specific user groups in an IMS environment.

When you invoke this function, the Broadcasting Menu appears from which you can select the following functions:

- Create Broadcast Messages
- List all Broadcast Messages

For details on the broadcasting function, see the section Natural under IMS/TM - Special Functions in the Natural TP Monitor Interfaces documentation.

## Display Environment Data

This function is used to display environmental data on Natural IMS/TM.

When you invoke this function, the Environment Table screen appears for the environment table used by the current Natural session. The screen displays the current parameter settings of the Natural IMS/TM interface.

The parameters cannot be updated. For more information on IMS parameters, see the section Natural under IMS/TM in the Natural TP Monitor Interfaces documentation.

## Monitoring

This function is used to display monitoring data about Natural user sessions that run under the same Natural subsystem.

When you invoke this function, the Monitoring screen appears where you can select the following functions to display monitoring data about user sessions:

- **Active Sessions:**  
Displays all active Natural sessions that run under IMS/TM.
- **Suspended Sessions:**  
Displays all Natural sessions that are currently suspended under IMS/TM.
- **User Selection:**  
Invokes a window where you can specify selection criteria to display particular Natural user sessions only.

## Multi Session

This function is used to display or create multiple Natural sessions.

When you invoke this function, the IMS Multi Session screen appears where all active sessions are listed. Additionally, the settings of the Create and Resume keys are displayed.

## Applied NII Zaps

This function invokes the Applied NII Zaps screen, which displays the numbers of all Zaps that have been applied to the current Natural TP environment.

# SYSTP Functions under TIAM and UTM

This section provides information on the SYSTP utility functions available under TIAM and UTM.

## To invoke the SYSTP functions under TIAM and UTM

- On the Natural SYSTP Utility Main Menu, in the Code field, enter **E** for Environment-Dependent Functions.

On the menu displayed then, you can select the following functions which are explained in this section:

- P-Key Utility
- Show Common Memory Pools

## P-Key Utility

This function supports the loading of programmable P keys on Siemens 975X terminals under the TP monitors TIAM and UTM.

This function allows you to either load the standard Natural key settings (function-key mode KN, KO or KS) to the keys P1 to P20 or to load user-defined values to individual keys.

When you invoke the P-Key utility, the following menu appears:

```

15:54:05          ***** NATURAL SYSTP UTILITY *****          1998-03-25
User VR000001          - P-Key Utility -          TID 0709

                Code   Function                Parameter
                KU    Load User Values        A,H
                KS    Set KS Mode             L,N
                KN    Set KN Mode             L,N
                KO    Set KO Mode             L,N
                KF    Load F1 - F20
                ?     Help
                .     Exit

                Code .. ___                Parameter A

Select function.
Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                Help Menu Exit KU    KS    KSN  KN    KNN  KO    KON  KF    Canc

```

On this menu, you enter a function code and an optional parameter code. The valid parameter codes for a function are listed to the right of the function. The various codes have the following meaning.

Parameter	Meaning
A	Values are entered in alphanumeric format.
H	Values can be entered in alphanumeric or hexadecimal format.
L	Load option. Mode is set and P keys are loaded.
N	No-load option. Mode is set, but P keys are not loaded.

## Load User Values

This function allows you to load your own values to the keys P1 to P20 (for example, terminal functions, send codes). Values can be entered either in alphanumeric or hexadecimal format. You choose the desired format by entering **A** (alphanumeric format) or **H** (hexadecimal or alphanumeric format) in the Parameter field on the menu. When you invoke the Load User Values function (by pressing PF4 or by entering Code **KU** on the menu of the P-key utility), the following screen is displayed:

```

15:55:41          ***** NATURAL SYSTP UTILITY *****          1998-06-25
User VR000001          - Edit User Keys -          TID VR000001

P1:          _____
              40404040404040404040404040404040404040404040404040404040
P2:          _____
              40404040404040404040404040404040404040404040404040404040
P3:          _____
              40404040404040404040404040404040404040404040404040404040
P4:          _____
              40404040404040404040404040404040404040404040404040404040
P5:          _____
              40404040404040404040404040404040404040404040404040404040
P6:          _____
              40404040404040404040404040404040404040404040404040404040
P7:          _____
              40404040404040404040404040404040404040404040404040404040
P8:          _____
              40404040404040404040404040404040404040404040404040404040
Function ( + / - / L / . )          _          Mode:          ALPHA
Select function or fill up values for P-Keys.
Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
          Menu  Exit  Load          -          +          Canc
    
```

The Mode is set to HEX or ALPHA depending on the parameter you specify when invoking the function. You can switch modes by overwriting the existing mode with A(LPHA) or H(EX).

- In ALPHA mode, you can use the left half of the screen to enter an alphanumeric value next to the key you wish it to be loaded to.
- In HEX mode, you can also assign a value to a key in hexadecimal form on the right half of the screen.

For each P key, enter an alphanumeric value in the empty input field or a hexadecimal value in the line below it (for Parameter Value **A**, the hexadecimal field is input blocked).

If no value is specified for a key, the standard Natural key setting (function-key mode KN, KO or KS) applies for this key; thus, it is possible to have a mixed P-key usage; that is, some keys with user-defined functions, others with the standard Natural functions.

Load the values by pressing PF4 or by entering **L** in the Function field.

Page the screen to additional P keys by pressing PF8 or by entering a plus (+) sign in the Function field.

**Note:**

Natural automatically converts all binary values which are smaller than H'40' to H'6F' (= question mark). So, before any binary values smaller than H'40' can be loaded, the macro CMTAB (translation table) has to be changed accordingly so as to avoid this automatic conversion. This is particularly important for H'27' (= ESCAPE) and H'19' (= Endemarke). Whenever CMTAB is modified, the Natural print mode module NATPM has to be reassembled and relinked to the Natural nucleus.

**Set Key Assignment Mode**

The following functions are used to set key assignment modes on Siemens terminals:

<b>Mode</b>	<b>Function</b>
Set KS Mode	Executes the terminal command %KS and is invoked either by pressing PF5 or by entering Code <b>S</b> on the menu of the P-key utility.
Set KN Mode	Executes the terminal command %KN and is invoked either by pressing PF7 or by entering Code <b>N</b> on the menu of the P-key utility.
Set KO Mode	Executes the terminal command %KO and is invoked either by pressing PF9 or by entering Code <b>O</b> on the menu of the P-key utility.

For a full explanation of key assignment modes, see Natural under BS2000/OSD in the Natural Operations for Mainframes documentation.

## Loading Send-Key Codes to P Keys

This function is used to load specific send-key (F) codes F1 to F20 to the keys P1 to P20. The function is similar to the key assignment mode KN, except that F codes can be selected individually.

When this function is invoked, the following screen appears:

```

15:56:34          ***** NATURAL SYSTP UTILITY *****          1998-06-25
User VR000001          - Load F-Codes -          TID VR000001

P01  _           P02  _           P03  _           P04  _           P05  _
P06  _           P07  _           P08  _           P09  _           P10  _
P11  _           P12  _           P13  _           P14  _           P15  _
P16  _           P17  _           P18  _           P19  _           P20  _

Mark P-Key to be loaded with F-Code
Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Load      Menu  Exit                                     Canc
    
```

To load P keys with F codes, mark the appropriate keys and press ENTER. Only the keys which are marked are loaded with F codes. Other P keys retain their original values.

## Application Programming Interface LPFSUP01

The Load User Values function is also available to user applications as an application programming interface (API). The API consists of the Natural subprogram LPFSUP01, which performs the loading of the keys. LPFSUP01 can be copied into user libraries or steplib.

LPFSUP01 is called as follows:

```
CALLNAT 'LPFSUP01' P-VALUE(*)
```

P-VALUE must be defined as an array: (A24/20).

### Example:

```
DEFINE DATA LOCAL
  1 P-VALUE (A24/20)
END-DEFINE
* LOAD '/STA L EM DUE1' TO P1, '/STA P EM DUE1' TO P4
COMPRESS '/STA L' h'192786' INTO P-VALUE(1)
COMPRESS '/STA P' h'192786' INTO P-VALUE(4)
CALLNAT 'LPFSUP01' P-VALUE(*)
END
```

See also the example LPFEXAM1 in the library SYSTP.

## Show Common Memory Pools

This function displays a list of all common memory pools used in Natural.

The individual items of information shown for each common memory pools are explained in the online help of this function.

# SYSTP in Batch for CICS Sessions

SYSTP can also be used to obtain statistical data on Natural/CICS sessions in batch mode.

The Natural log file into which the statistical data on Natural/CICS sessions is written must be assigned to the Natural batch job as Work File 1 (that is, via CMWKF01). It must also be defined to the online system, which means in the CICS DCT (destination control table); see the LOGDEST parameter in macro NCIPARM of the Natural/CICS parameter module in the section Natural under CICS in the Natural TP Monitor Interfaces documentation.

- Invoking SYSTP in Batch Mode
  - Evaluating the Log File
- 

## Invoking SYSTP in Batch Mode

To invoke the SYSTP utility in batch mode, you specify either of the following commands in the batch job:

- SYSTP *xxx*
- LOGON SYSTP  
SYSBATCH *xxx*

where *xxx* indicates what kind of data is to be processed; *xxx=nci*, for example, would indicate that the data is collected by a Natural/CICS online system.

## Evaluating the Log File

Data is written into the Natural log file when Natural is initialized or reset, and when a Natural session is terminated.

The Natural CICS Interface writes the following records into the Natural log file:

- a start log record whenever the Natural environment is initialized or reset,
- a session log record whenever a Natural session is terminated.

When a Natural environment is initialized, a system ID is written into the system control block. This system ID also belongs to all log records. Therefore, a Natural log file can be shared by several Natural/CICS online environments.

The information logged serves to keep track of the usage of the Natural/CICS online environment. Therefore, most of the information refers to facilities of the Natural environment. The log file is not intended to be an accounting or monitoring tool that refers to facilities of CICS.

Based on the system ID, several reports are created with data related to a Natural session:

- log file data listed in chronological order, which means that session log records are sorted by session end date and time;
- statistical information on how the Natural environment was set up and used;
- statistics on thread groups (if used);
- statistics on program storage threads (if used);
- statistics on roll facilities (if used).

This set of reports is created for all Natural environments with records about Natural/CICS sessions in the Natural log file.

**Note:**

The session termination log records, of course, reflect only resources which have been used by the corresponding sessions. Therefore, these records may not reflect the full Natural environment. Reports of a full Natural environment can be obtained by making a snapshot of the whole Natural environment using the System Administration Facilities.

**Sample Batch Job - OS/390:**

```
//NATLOG JOB (user,,999),CLASS=K,MSGCLASS=Z
//NATBATCH EXEC PGM=natbatch,REGION=2000K,
// PARM=( 'DBID=nn,FNR=nn,IM=D,MT=0,MADIO=0,AUTO=OFF,MENU=OFF' )
//STEPLIB DD DISP=SHR,DSN=natural.loadlib
//DDCARD DD *
ADARUN DA=nn,SVC=nnn,DE=3380
//CMPRINT DD SYSOUT=A
//CMWKF01 DD DISP=SHR,DSN=nat.log.file
//SYSOUT DD SYSOUT=X
//SYSUDUMP DD SYSOUT=X
//CMSYNIN DD *
SYSTP NCI
/*
```

**Sample Batch Job - VSE/ESA:**

```
* $$ JOB JNM=NATLOG,CLASS=0,DISP=D,PRI=3
* $$ LST CLASS=A,DISP=D,COPY=1,RBS=100,DEST=*
// JOB NATLOG
// DLBL CMWKF01, 'nat.log.file'
// EXTENT SYS001,xxxxxx
// ASSGN SYS001,DISK,VOL=xxxxxx,SHR
// ASSGN SYSLST,cuu
// LIBDEF PHASE,SEARCH=...
// EXEC natbatch,SIZE=natbatch,PARM='SYSRDR'
DBID=nn,FNR=nn,IM=D,MT=0,OBJIN=R,MADIO=0
AUTO=OFF,MENU=OFF,BWORKD=(1,1,4628,VB)
/*
ADARUN DA=nn,SVC=nnn,DEVICE=3380,TNAE=999999,TT=999999,MODE=MULTI
/*
SYSTP NCI
/*
// EXEC LISTLOG
/&
* $$ EOJ
```

# SYSTRANS Utility - Overview

The information provided in this section deals with all environments supported by Natural.

The SYSTRANS utility documentation covers the following topics:

- Introduction
- Unload Function
- Load, Scan and Restart Load Functions
- Direct Transfer Functions
- Direct Commands and CALLNAT Interface
- SYSTRANS Profile

# SYSTRANS - Introduction

- General Information
  - Invoking SYSTRANS Online
  - Name and Range Specification
  - SYSTRANS and Entire Connection
  - SYSTRANS and Natural Security
- 

## General Information

The Natural utility SYSTRANS allows you to transfer all Natural objects, maps, DDMs, libraries, command processors and error messages as well as Adabas FDTs from one hardware platform to another.

To be able to do this, the SYSTRANS utility uses a general record layout. This general record layout is independent of all hardware platforms and is used only for transporting objects. Natural objects are read from one hardware platform and then restructured according to the general record layout using the Unload function of the utility.

The formatted records are written to a Natural work file which can be transported to another platform using standard file transfer services. On the target platform, the Natural objects can then be read from the work file, and loaded into the local file or database system using the Load function of SYSTRANS. Natural objects are read from the work file and then restructured according to the structure of the new hardware platform.



**Warning:**

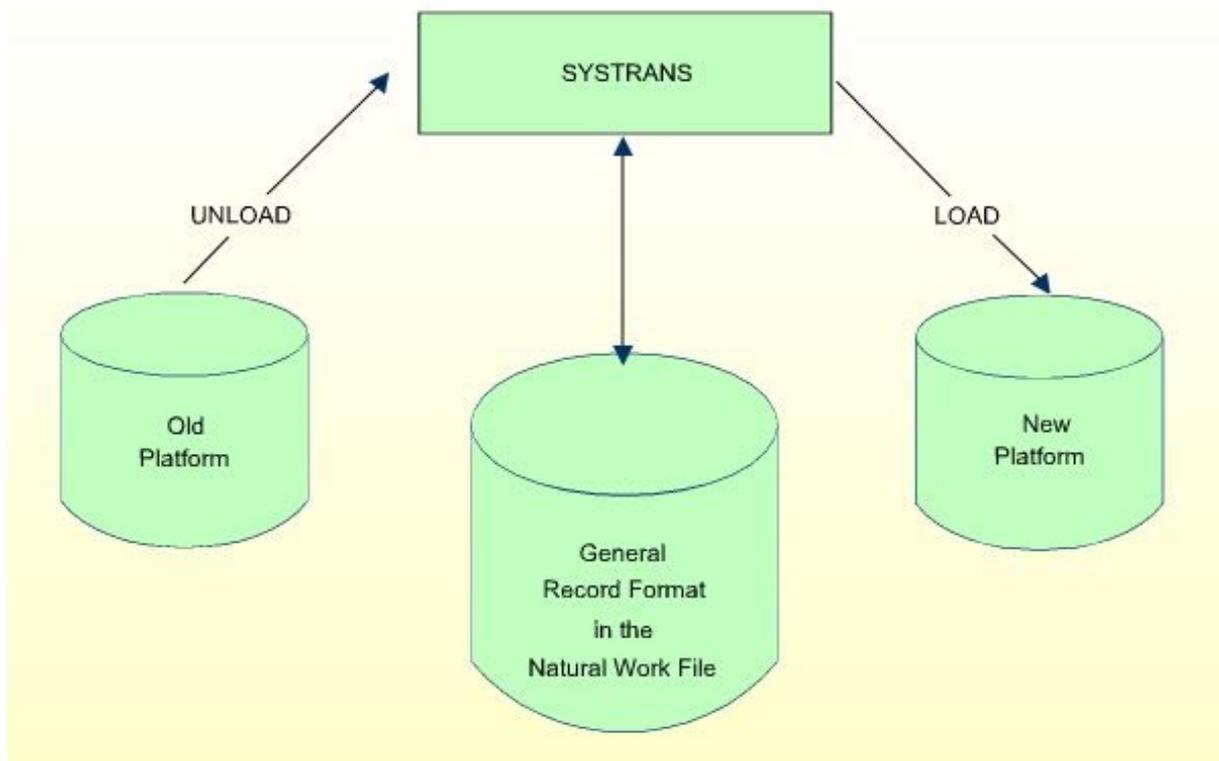
**When you are unloading objects, only one file per unload session is assigned to the work file.**

When you are unloading or loading command processors, LFILE 190 must point to an Adabas file for command processor sources.

Any functionality of the SYSTRANS utility can be restricted by using the user exit TRA-E1-S provided in the library SYSTRANS; for further details, refer to the information provided in the TRA-E1-S source.

You can define a profile for your SYSTRANS utility, both generally and user-specifically. For this purpose, Natural is delivered with the text object PROFILE. To activate this profile, you have to resave PROFILE under the name TRANPROF in the library SYSTRANS. See more information in the section SYSTRANS Profile.

The following graphic shows how the SYSTRANS utility works when transferring a Natural application from one platform to another:



The SYSTRANS functions Unload, Load, Scan and Restart can also be executed in direct-command mode or from within a Natural application; see also the section Direct Commands and CALLNAT Interface.

## Invoking SYSTRANS Online

There are two methods for invoking the SYSTRANS utility for interactive usage:

### ▶ To invoke SYSTRANS online from any Natural library

- Enter the command SYSTRANS.  
If you issue a Natural system command from the SYSTRANS command line, the command will apply to the library from which SYSTRANS has been invoked.

### ▶ To invoke SYSTRANS online from the Natural Main Menu

1. Select Maintenance and Transfer Utilities.  
The corresponding menu is displayed.
2. Select Transfer Objects to Other Platforms.  
The SYSTRANS Main Menu is displayed.

The SYSTRANS Main Menu offers the following options:

Unload TRANSFER Objects

Load TRANSFER Objects

Direct TRANSFER Functions

Scan TRANSFER Work File

Restart TRANSFER Load

See the corresponding sections for a description of these functions.

## Name and Range Specification

To select Natural objects, maps, DDMs or command processors, you can specify a name or a range of names.

In the list of options below, *value* is any combination of one or more characters:

	<b>Input</b>	<b>Selected Items</b>
Start Value	<i>value</i> >	All items whose names are greater than or equal to <i>value</i> .  Example: AB> Selected: AB, AB1, BBB, ZZZZZZZ Not selected: AA1, AAB
End Value	<i>value</i> <	All items whose names are less than or equal to <i>value</i> .  Example: AX< Selected: AB, AWW, AX Not selected: AXA, AY
Leading Characters	<i>value</i> *	All items whose names begin with <i>value</i> .  Example: AB* Selected: AB, AB1, ABC, ABEZ Not selected: AA1, ACB

## SYSTRANS and Entire Connection

You can use Entire Connection for Windows to transfer small applications between sessions in Natural mainframe or UNIX environments and sessions on the PC. You can also use Entire Connection to unload or load transfer files to or from a PC.

**Note:**

The size of a library to be transferred by Entire Connection is limited by the available disk space and time. The size of a library can be estimated by adding up the values for "Size in ESIZE" for saved Natural objects, which can be obtained by issuing the command "L DIR \*".

If you use Entire Connection, Work File 7 must be defined as Entire Connection work file.

For the definition of the work files, see Unload Work File Specifications (section: Unload Function) and Load Work File Specifications (section: Load, Scan and Restart Load Functions).

Entire Connection automatically converts the transferred data from EBCDIC to ASCII or vice versa; therefore, the SYSTRANS conversion option should not be set to Y (Yes).

**Note:**

Work File 3 does not have to be a PC work file.

Before you start the SYSTRANS utility, you have to enter the command "%+" to enable the Entire Connection communication with the PC.

You invoke and use the Unload Function and the Load, Scan and Restart Load Functions in exactly the same way as described in the corresponding sections.

Once you have specified the parameters for one of the options of either function, you are prompted to enter the name of the file to which the data are to be downloaded.

After completing an unload session, you have to return to the SYSTRANS Main Menu to close Work File 7.

## **SYSTRANS and Natural Security**

For the use of SYSTRANS in a Natural Security environment, see the section Protecting Utilities in the Natural Security documentation.

# SYSTRANS - Unload Function

You use the Unload function to unload source code objects. The Unload function performs the following:

- unloads data for any operating system;
- if requested, converts the data from ASCII to EBCDIC or from EBCDIC to ASCII;
- formats the objects to be unloaded according to the general record layout.

Below is information on:

- General Unload Options
  - Objects to be Unloaded
  - Unloading in Batch Mode under OS/390
  - User Exits for Unloading in Batch Mode
  - Unload Work File Specifications
  - Natural Profile Parameters
- 

## General Unload Options

When you invoke the Unload function, you first have to specify the General Unload Options before you can select the objects to be unloaded.

### To specify any of the General Unload Options

- Enter a Y (Yes) or an N (No).

You can specify the following:

Option	Explanation
Conversion EBCDIC ==> ASCII ASCII ==> EBCDIC	Conversion from EBCDIC to ASCII or from ASCII to EBCDIC is performed, depending on your hardware environment.
User-Defined Conversion Table	Applies if the above Conversion option is selected. If you have specified your own conversion table in the program SULCONV, all data are converted using this conversion table. If the User-Defined Conversion Table option is not selected, all data are converted using a conversion table that is internally defined. See also User-Defined Conversion Table below for more information on defining your own conversion table in the program SULCONV.
Substitute Line References	If line numbers are used as references in the source code, the line numbers of the referenced lines and the line number references are replaced with labels. The sources are not modified in the database.
Report	A report is displayed listing the objects that were unloaded.
Include Line Numbers	By default, line numbers in Natural objects are not transferred. Select this option if you wish to transfer line numbers, too.
Use Work File Input	Input for the Unload function is taken from a work file; see also the section Work File for Unload Command Input (Work File 2).
Use Selection List	If this option is set, a list is displayed where you can select objects, maps, libraries or DDMs to be unloaded; see further details under Applying the Use Selection List Option.
Use Entire Connection Work File	All unloaded data are directly written to Work File 7 when you exit the Unload function.  Work File 7 must have been defined as Entire Connection work file in your parameter file.

## User-Defined Conversion Table

In the library SYSTRANS, there is the program SULCONV with which you can define your own conversion tables. In this program, you can replace any character to match your hardware environment.

### Example:

The following steps are an example of how the ASCII character **A** can be converted to the EBCDIC character **a**:

1. Find out the decimal representation of the ASCII character **A**. In this case, the decimal representation of the ASCII character **A** is 65.
2. Find out the decimal representation of the EBCDIC character **a**. In this case, the decimal representation of the EBCDIC character **a** is 129.
3. Replace the value located at the 65th position of the table BASCE with 129.

## Applying the Use Selection List Option

If the Use Selection List unload option is set, a list is displayed which is used to select and unload objects, maps, libraries or DDMs from the specified library.

### To set the Use Selection List option

- Enter a Y (Yes).

The selection list is displayed after you have made all necessary specifications for the objects to be unloaded as described below.

### To start the unload procedure

1. Either select all items by pressing PF5 (All),

Or select a single item or multiple items by marking it (them) with any character entered in the Cmd column and then press ENTER.

2. Press PF4 (Unld).

To exit, press PF3 or enter a period (.) in the command line.

## Objects to be Unloaded

Once you have specified the General Unload Options, a menu is displayed on which you select one of the object types below to be unloaded.

- Natural Objects
- Maps
- DDMs
- Adabas FDTs
- Error Message Texts
- Command Processors
- Library

### Unloading Natural Objects

If you select Natural Objects as the type of objects to be unloaded, you can make the following specifications:

Field	Explanation
S/C Type	For future use; nothing can be specified at the moment.
From Library	The name of the library where the Natural objects are located. If the Use Selection List option is set, you can invoke a selection list of all available libraries and unload them: Press PF1 and select the library.
Object Name	The name of the object to be unloaded See also Name and Range Specification in the section Introduction.  The Object Name field does not appear if the Use Selection List option is set.  To display a selection list of all objects contained in the specified library: Press ENTER. See also Applying the Use Selection List Option.
Object Type	The type(s) of the object(s) to be unloaded.
To Library	The name of the library into which the objects are to be loaded. If no library is specified, the name specified as From Library is used.
Data Area Format	The format in which to unload data area sources. Valid values are:  <ul style="list-style-type: none"> <li>N Unloads data areas in new internal data area format.</li> <li>O Unloads data areas in old internal data area format.</li> </ul> <p>If data area sources cannot be converted into old internal data area format, SYSTRANS stops processing and issues a corresponding error message.</p> <ul style="list-style-type: none"> <li>* Unloads data areas in the format in which they are stored.</li> </ul> For details, see Data Area Editor in the Natural Editors documentation.

### Options for the Natural Objects Selection List

See also the section Applying the Use Selection List Option.

In this selection list, you can limit the number of Natural objects displayed by specifying the following:

Field	Explanation
S/C Type	For future use; nothing can be specified at the moment.
Member	A single name or a range of names of Natural objects to be listed. See Name and Range Specification in the section Introduction. Additionally, you can use a wildcard (?) within the object name or within the asterisk (*) range.
Type	The types of object to be listed.  To display a selection list of possible object types: Enter a question mark (?) in this field or press PF1 (Help).

Field	Explanation										
Mode	The programming mode with which an object has been created. Valid values are:  S structured mode; R reporting mode; * both structured and reporting mode or no mode, as with data areas on mainframe computers.										
Version	A mask for the version of objects to be listed. All objects for which the version matches this mask are listed. If an asterisk (*) is specified as version mask, no checks are performed.										
User ID	The ID of the user who saved the objects to be listed. If an asterisk (*) is specified as user ID, no checks are performed.										
Save Date	A date on which or a date range within which an object was saved. Options are: <ul style="list-style-type: none"> <li>● a date (2003-01-17)</li> <li>● a starting date range (2002-06&gt;). If you specify a starting date, all Natural objects saved on or after this date are listed;</li> <li>● an ending date range (2003-01&lt;). If you specify an ending date, all Natural objects saved up to this date are listed;</li> <li>● a date mask (2003-01 or 2003-01*). If you specify a date mask, all Natural objects saved during that particular time period are listed.</li> </ul> <p>As abbreviations for special dates or date ranges, the following strings can be entered:</p> <table border="0"> <thead> <tr> <th>String</th> <th>Explanation</th> </tr> </thead> <tbody> <tr> <td><u>T</u>ODAY</td> <td>The date of the current day. The day can be followed by <i>+nnnn</i> or <i>-nnnn</i> (where <i>nnnn</i> are numeric values) and/or by &gt; or &lt; . The resulting date is computed as the date of the current day plus or minus <i>nnnn</i> days.</td> </tr> <tr> <td><u>Y</u>ESTERDAY</td> <td>The date of the day before the current day.</td> </tr> <tr> <td><u>M</u>ONTH</td> <td>The date range of the current month.</td> </tr> <tr> <td><u>Y</u>EAR</td> <td>The date range of the current year.</td> </tr> </tbody> </table>	String	Explanation	<u>T</u> ODAY	The date of the current day. The day can be followed by <i>+nnnn</i> or <i>-nnnn</i> (where <i>nnnn</i> are numeric values) and/or by > or < . The resulting date is computed as the date of the current day plus or minus <i>nnnn</i> days.	<u>Y</u> ESTERDAY	The date of the day before the current day.	<u>M</u> ONTH	The date range of the current month.	<u>Y</u> EAR	The date range of the current year.
String	Explanation										
<u>T</u> ODAY	The date of the current day. The day can be followed by <i>+nnnn</i> or <i>-nnnn</i> (where <i>nnnn</i> are numeric values) and/or by > or < . The resulting date is computed as the date of the current day plus or minus <i>nnnn</i> days.										
<u>Y</u> ESTERDAY	The date of the day before the current day.										
<u>M</u> ONTH	The date range of the current month.										
<u>Y</u> EAR	The date range of the current year.										
Save Time	A time at which or a time range within which an object was saved. Options are: <ul style="list-style-type: none"> <li>● a time (10:11:12)</li> <li>● a starting time range (09&gt;). If you specify a starting time, all Natural objects saved at or after this time are listed;</li> <li>● an ending time range (17:00&lt;). If you specify an ending time, all Natural objects saved up to this time are listed;</li> <li>● a time mask (10: or 10:* ). If you specify a time mask, all Natural objects during that particular time period are listed.</li> </ul> <p>The time must be specified in the format <i>HH:II:SS</i> (<i>HH</i> = hours, <i>II</i> = minutes, <i>SS</i> = seconds).</p>										

## Unloading Maps

If you select Maps as the type of objects to be unloaded, you can make the following specifications:

Field	Explanation
S/C Type	For future use; nothing can be specified at the moment.
From Library	The name of the library where the maps are located. If the Use Selection List option is set, you can invoke a selection list of all available libraries and unload them: Press PF1 and select the library.
Map Name	The name of the map to be unloaded. See also Name and Range Specification in the section Introduction.  The Map Name field does not appear if the Use Selection List option is set.  To display a selection list of all maps contained in the specified library: Press ENTER.  See also Applying the Use Selection List Option.
Incorporate all Predict Rules	If you set this option and Predict is installed, all Predict rules associated with the map are incorporated into the map source.
Unload associated Free Rules	This option is currently not available.
To Library	The name of the library into which the maps are to be loaded. If no library is specified, the name specified as From Library is used.

**Options for the Map Selection List**

In this selection list, you can limit the number of maps displayed. Here, the same applies as for the Options for the Natural Objects Selection List with the exception of the Types field which cannot be changed.

**Unloading DDMs**

If you select DDMs as the type of objects to be unloaded, you can make the following specifications:

Field	Explanation
DDM Name	<p>The name of the DDM to be unloaded. See also Name and Range Specification in the section Introduction.</p> <p>If the Use Selection List option is set, you can invoke a selection list with all DDMs contained in the specified library or in the FDIC system file: Specify any range as DDM name and press ENTER.</p> <p>See also Applying the Use Selection List Option.</p>
From Library	<p>Only to be used in UNIX and Windows environments.</p> <p>The name of the library where the DDMs are located.</p> <p>If the Use Selection List option is set, you can invoke a selection list of all available libraries: Press PF1.</p>
To Library	<p>Only to be used if your target environment is UNIX or Windows.</p> <p>The name of the library into which the DDMs are to be loaded.</p> <p>If no library is specified, the following applies:</p> <ul style="list-style-type: none"> <li>● If the DDMs are unloaded from a mainframe environment, all DDMs will be automatically loaded into the library SYSTEM.</li> <li>● If the DDMs are unloaded from any other environment, the name of the library specified as From Library will be used.</li> <li>● If the target environment is a mainframe environment, any specification is ignored.</li> </ul>

### Options for the DDM Selection List

See also the section Applying the Use Selection List Option.

In this selection list, you can limit the DDMs displayed by specifying the following items:

Option	Explanation
DDM Name	<p>A single name or a range of names of DDMs to be listed. See Name and Range Specification in the section Introduction. Additionally, you can use a wildcard (?) within the object name or within the asterisk (*) range.</p>
DBID	<p>The database ID of the DDMs to be listed. If you wish to list only DDMs which have a specific database ID (1), enter that database ID in this field.</p>
FNR	<p>The file number of the DDMs to be listed. If you wish to list only DDMs which have a file number (1), enter that file number in this field.</p> <p>(1) These are the database ID and file number of the physical database file of which the DDM is the logical representation.</p>

### Unloading Adabas FDTs

If you select Adabas FDT as the type of objects to be unloaded, you can make the following specifications:

Field	Explanation
Source DBID	The database ID of the Adabas FDT you want to unload.
Source FNR	The file number of the Adabas FDT you want to unload.
Target DBID	The database ID to be used by the target system for the Adabas FDT. If you enter a 0, the Source DBID specification applies.
Target FNR	The file number to be used by the target system for the Adabas FDT. If you enter a 0, the Source FNR specification applies.
Adabas Password for Source FDT	The appropriate password if your Adabas FDT is password-protected.
Adabas Cipher Code for Source FDT	The appropriate cipher code if your Adabas FDT is protected by a cipher code.

## Unloading Error Message Texts

If you select Error Message Texts as the type of objects to be unloaded, you can make the following specifications:

Field	Explanation
Message Type	The type of error messages to be unloaded:  U User-defined error messages N Natural error messages
From Library	The name of the library where the messages are located (only to be used with user-defined error messages).  If the Use Selection List option is set, you can invoke a selection list of all available libraries: Press ENTER.
Message Number	The range of error message numbers to be unloaded.
Language Code	The language code of the error messages to be unloaded. If you specify an asterisk (*), all language codes are unloaded. For valid codes, see the description of the system variable *LANGUAGE in the Natural System Variables documentation
To Library	The name of the library into which the messages are to be loaded (only to be used with user messages). If no library is specified, the name specified as From Library is used.

**Note:**

For Natural error messages, you need not specify a library, because they are always unloaded from either the FNAT system file or the error messages subdirectory.

## Unloading Command Processors

If you select Command Processors as the type of objects to be unloaded, you can make the following specifications:

Field	Explanation
S/C Type	For future use; nothing can be specified at the moment.
From Library	The name of the library where the command processors are located. If the Use Selection List option is set, you can invoke a selection list of all available libraries: Press PF1 and select the library.
Command Processor Name	The name of the command processor to be unloaded. See also Name and Range Specification in the section Introduction.  The Command Processor Name field does not appear if the Use Selection List option is set.  To display a selection list of all command processors contained in the specified library: Press ENTER.  See also Applying the Use Selection List Option.
To Library	The name of the library into which the command processors are to be loaded. If no library is specified, the name specified as From Library is used.

### Options for the Command Processor Selection List

See also the section Applying the Use Selection List Option.

In this selection list, you can limit the command processors displayed by specifying the following items:

Option	Explanation
Member	A single name or a range of names of command processors to be listed. See Name and Range Specification in the section Introduction. Additionally, you can use a wildcard (?) within the object name or within the asterisk (*) range.

### Unloading Libraries

If you select Library as the type of objects of various types to be unloaded from a specific library, you can make the following specifications:

Field	Explanation
From Library	The name of the library to be unloaded. If the Use Selection List option is set, you can invoke a selection list of all available libraries: Press ENTER.
Unload Natural Objects	Indicates whether the Natural objects contained in the specified library are to be unloaded.  Enter a Y (Yes) to unload Natural objects.
Range of Natural Objects	The name of the object to be unloaded. See also Name and Range Specification in the section Introduction.
Types of Natural Objects	The type(s) of the object(s) to be unloaded. If you specify an asterisk (*), all types are unloaded.
Unload Maps	Enter a Y (Yes) to unload maps.
Range of Maps	The name of the map to be unloaded. See also Name and Range Specification in the section Introduction.
Unload DDMs	Indicates whether the DDMs contained in the FDIC system file are to be unloaded. Enter a Y (Yes) to unload DDMs.
Range of DDMs	The name of the DDM to be unloaded. See also Name and Range Specification in the section Introduction.
Unload Error Messages	Enter a Y (Yes) to unload error messages.
Message Number	The range of error message numbers to be unloaded.
Language Code	The language code of the error messages to be unloaded. If you specify an asterisk (*), all language codes are unloaded. For valid codes, see the system variable *LANGUAGE in the Natural System Variables documentation.
Unload Command Processors	Enter a Y (Yes) to unload command processors.
Range of Command Processors	The name of the command processor to be unloaded. See also Name and Range Specification in the section Introduction.
To Library	The name of the library into which the objects are to be loaded. If no library is specified, the name specified under From Library is used.
Data Area Format	See Data Area Format in Unloading Natural Objects above.

## Unloading in Batch Mode under OS/390

The following example shows a JCL procedure which you can use to unload your programs and maps:

```

//*****
//*
//*   UNLOAD Natural SOURCES
//*
//*****
//TRANSFER JOB CLASS=G,MSGCLASS=X
//NATBAT EXEC PGM=NATBAT,REGION=3000K,
//          PARM='IM=D,INTENS=1'
//STEPLIB DD DISP=SHR,DSN=NATURAL.LOAD
//          DD DISP=SHR,DSN=ADABAS.LOAD
//CMPRINT DD SYSOUT=X
    
```

```

//DDCARD DD *
ADARUN  PROG=USER, DB=10, MODE=MULTI, SVC=249
//CMWKF01 DD DISP=SHR, DSN=WORK.FILE1
//CMWKF03 DD DISP=SHR, DSN=WORK.FILE3
//CMSYNIN DD *
SYSTRANS
U
N,N,N,Y,N,N,N,N
N
LIB1,*,*,LIB2
M
SRCLIB,*,Y,N,TARLIB
FIN
/*

```

**Note:**

As shown in the above example, do not specify the S/C Type field when unloading Natural objects, because the S/C Type field is an output field only.

## User Exits for Unloading in Batch Mode

Two user exits with which you can handle errors when unloading in batch mode are provided in source form under the names SUL-S-X1 and SUL-S-X2. To be invoked, both must be available as cataloged objects under the names SULEXIT1 and SULEXIT2 in the library SYSTRANS.

SULEXIT1 is invoked if an error occurs that leads to an abnormal termination. It allows you to define a return code.

SULEXIT2 is invoked in the case of error messages or warnings. If it returns a non-zero return code, the unload operation is abnormally terminated; otherwise processing is continued.

## Unload Work File Specifications

The following work files are used for unloading:

Work File 1	The file into which the data are unloaded, if the option Use Entire Connection Work File (see General Unload Options) is <b>not</b> specified.
Work File 2	The file which contains the unload commands if you use the option Use Work File Input as described under General Unload Options.
Work File 3	The file used for temporary storage while unloading; this work file is only used if the option Use Selection List is set. See also Applying the Use Selection List Option.
Work File 4	The file into which the data are unloaded temporarily when using the Direct Transfer Functions (see the relevant section).
Work File 7	The file (in Entire Connection format) into which the data are downloaded when specifying the option Use Entire Connection Work File (see General Unload Options).

Use the following JCL parameter values to specify the work files:

Parameter	Work File 1/4	Work File 2	Work File 3
LRECL	96	80	43
RECFM	VB	FB	FB
BLKSIZE	6240	6240	6450

**Note:**

Since SYSTRANS writes records with a variable length, Work File 1 should be defined with the RECFM value VB to reduce the work file size.

**Work File for Unload Command Input - Work File 2**

Unload commands for command input from Work File 2 can be specified in either of the following ways:

- by specifying the individual parameters at fixed positions,
- by specifying the individual parameters separated by a comma (,).

In both ways, the sequence in which the parameters are specified is the same. The way to be used is determined by the second byte which can be specified as either a blank (for fixed positions) or a comma.

**Fixed Parameter Positions**

Bytes	Explanation
01 - 01	Object Type: N = Natural object M = Map D = DDM F = Adabas FDT E = Error Message  Asterisk (*) = comment line
02 - 02	Blank

Bytes	Explanation
03 - 34	Object identification  <b>Object = N/M:</b> 03 - 10 Object name or (*); this position is mandatory.  <b>Object = D:</b> 03 - 34 Object name or (*); this position is mandatory.  <b>Object = E:</b> 03 - 06 Start error message number; this position is mandatory. 07 - 07 Blank 08 - 11 End error message number or blank. 12 - 12 Blank 13 - 20 Language code, blank or (*); blank = default = (*).  <b>Object = F:</b> 03 - 07 Source DBID; this position is mandatory. 08 - 08 Blank 09 - 13 Source FNR; this position is mandatory. 14 - 14 Blank 15 - 19 Target DBID or blank; blank = Source DBID. 20 - 20 Blank 21 - 25 Target FNR or blank; blank = Source FNR.
35 - 35	Blank
36 - 43	From Library; this position is mandatory if the object = N/M/D/E (exceptions: D on mainframes and E only if error message type = U or blank).
44 - 44	Blank
45 - 52	To Library or blank if the object = N/M/D/E; blank = From Library.  E only if error message type = U or blank.
53 - 53	Blank

Bytes	Explanation
54 - 70	<p>Additional parameters</p> <p><b>Object = N:</b> 54 - 68 Object type list, blank or (*); blank = default = (*).</p> <p><b>Object = M:</b> 54 - 54 Y (Yes), N (No) or blank: Incorporate Predict Rules; blank = default = N. 55 - 55 Blank 56 - 56 Y (Yes), N (No) or blank: Unload associated Free Rules; blank = default = N.</p> <p><b>Object = F:</b> 54 - 61 Adabas password 62 - 62 Blank 63 - 70 Adabas cipher code</p> <p><b>Object = E:</b> 54 - 54 N (No), U (User-defined) or blank: error message type; blank = default = U.</p>

**Example:**

```

N PRGMNAME                FROMLIBR TOLIBRRY PNSGLAT
M MAPMNAME                FROMLIBR TOLIBRRY N N
D DDMNAME8901234567890123456789012 FROMLIBR TOLIBRRY
E 1234 5678 12345678     FROMLIBR TOLIBRRY U
F 12345 12345 12345 12345 ADAPASSW 12345678
    
```

**Parameters Separated by a Comma**

When specifying parameters separated by a comma, the following rules apply:

- The parameters must be separated by a comma (,).
- Blank characters between parameters are not required.
- The sequence of the parameters must be as with fixed positions.
- If a parameter is omitted, a comma (,) must be specified instead.

**Example:**

```

N,PRGMNAME, FROMLIBR, TOLIBRRY, PNSGLAT
M,MAPMNAME, FROMLIBR, TOLIBRRY, N, N
D,DDMNAME8901234567890123456789012, FROMLIBR, TOLIBRRY
E,1234,5678,12345678, FROMLIBR, TOLIBRRY, U
F,12345,12345,, ADABASPW,12345678
    
```

**Natural Profile Parameters**

To be able to use the Unload function, set the following Natural parameters:

<b>Parameter</b>	<b>Value</b>
MT	0
MADIO	0
MAXCL	0
ESIZE	64

# SYSTRANS - Load, Scan and Restart Load Functions

This section contains information on:

- Load and Scan
  - Restart Load
- 

## Load and Scan

### Load

You use the Load function to load source code objects. The Load function performs the following:

- loads data from the work file into any system environment;
- if requested, converts the data from ASCII to EBCDIC or from EBCDIC to ASCII;
- gives you the option to load free rules into Predict.

### Scan

You use the Scan function to scan for source code objects in Work File 1 without actually loading them.

When you invoke the Load or Scan function, you first have to specify General Options before you can select the objects to be loaded or scanned.

Below is information on:

- General Load/Scan Options
- Objects to be Loaded/Scanned
- Loading in Batch Mode under OS/390
- User Exits for Loading in Batch Mode
- Load Work File Specifications
- Natural Parameter Settings

## General Load/Scan Options

 To specify any of the following General Load or Scan Options

- Enter a Y (Yes) or an N (No).

The following options are provided:

Option	Explanation
User-Defined Conversion Table	<p>If data conversion is required by your target environment: If you set this option and have specified your own conversion table in the program SULCONV, all data are converted using this conversion table. If you do <b>not</b> set this option, all data are converted using a conversion table that is internally defined.</p> <p><b>Load only:</b> If no data conversion is required by your target environment: If you specify F (Force), a table is used that only replaces characters, which means that no actual data conversion takes place.</p> <p>See User-Defined Conversion Table below for more information on defining your own conversion table in the program SULCONV.</p>
Report	<p>Not applicable to the Scan. The Scan generates reports automatically.</p> <p>Displays a report listing the objects that were loaded.</p>
Translate Sources to Upper Case	<p>Not applicable to the Scan. Translates any source code to be loaded to upper case.</p>
Save Restart Information	<p>Not applicable to the Scan. Saves restart information while loading. This information can be used to resume load functions that terminated abnormally. See also Restart Load Function.</p>
Modify Library Names	<p>Not applicable to the Scan. Enables modification of library name(s) specified as To Library before the actual load operation begins.</p> <p>Enter a Y (Yes). After you have made all necessary specifications for the items to be loaded, a window is displayed in which you can specify a new library name.</p>
Use Entire Connection Work File	<p>Uploads or scans all data from Work File 7. When using this option, Entire Connection is required, and Work File 7 must have been defined as Entire Connection work file in your parameter file.</p>
Convert Data Area Sources	<p>Specifies whether to convert the data area sources read from the work file in old internal data area format to new internal data area format. For details, see Data Area Editor in the Natural Editors documentation.</p> <p>Valid values are:</p> <p>Y      Converts data areas from old internal data area format to new internal data area format.</p> <p>N      Does not convert data areas.</p>

### User-Defined Conversion Table

In the library SYSTRANS, there is the program SULCONV with which you can define your own conversion tables. In this program, you can replace any character to match your hardware environment.

#### Example:

The following steps are an example of how the EBCDIC character **a** can be converted to the ASCII character **A**:

1. Find out the decimal representation of the EBCDIC character **a**. In this case, the decimal representation of the EBCDIC character **a** is 129.
2. Find out the decimal representation of the ASCII character **A**. In this case, the decimal representation of the ASCII character **A** is 65.
3. Replace the value located at the 129th position of the table BEBCA with 65.

## Objects to be Loaded/Scanned

Once you have specified the General Load/Scan Options, a menu is displayed on which you select one of the below object types to be loaded or scanned.

- Natural Objects
- Maps
- DDMs
- Adabas FDTs
- Error Message Texts
- Command Processors
- Library
- All Objects

### Loading/Scanning Natural Objects

If you select Natural Objects as the type of objects to be loaded or scanned, you can make the following specifications:

Field	Explanation
S/C Type	For future use; nothing can be specified at the moment.
Library	The library to be loaded or scanned. If you specify no library, all libraries are loaded or scanned. You can specify a group of libraries by using asterisk notation (*). You can only specify a library that has been specified as target library with the Unload function. If no target library has been specified, the name of the library specified as From Library is used.  If the Modify Library Names option (see General Load/Scan Options) is set, you can enter a new library name before the actual load operation begins.
Object Name	The name of the object to be loaded or scanned. See also Name and Range Specification in the section Introduction.
Object Type	The type(s) of object(s) to be loaded or scanned.
Replace	Not applicable to the Scan. If you specify Y (Yes) and a source with the same name as the one you are loading already exists in the target environment, the target source is replaced.

### Loading/Scanning Maps

If you select Maps as the type of objects to be loaded or scanned, you can make the following specifications:

Field	Explanation
S/C Type	For future use; nothing can be specified at the moment.
Library	The library to be loaded or scanned. If you specify no library, all libraries are loaded or scanned. You can specify a group of libraries by using asterisk notation (*). You can only specify a library that has been specified as target library with the Unload function. If no target library has been specified, the name of the library specified as From Library is used.  If the Modify Library Names option (see General Load/Scan Options) is set, you can enter a new library name before the actual load operation begins.
Map Name	The name of the map to be loaded or scanned for. See also Name and Range Specification in the section Introduction.
Load Free Rules to Predict	Not applicable to the Scan.  If you specify Y (Yes), all free rules on the work file are loaded to Predict.
Replace	Not applicable to the Scan.  If you specify Y (Yes) and a map with the same name as the one you are loading already exists in the target environment, the target map is replaced.

### Loading/Scanning DDMs

If you select DDMs as the type of objects to be loaded or scanned, you can make the following specifications:

Field	Explanation
DDM Name	The name of the DDM to be loaded or scanned for. See also Name and Range Specification in the section Introduction.
Library	The name of the library to be loaded or scanned. Only to be used if your target environment is UNIX or Windows.  You can only specify a library that has been specified as target library with the Unload function. If no library is specified, the following applies to the load procedure: <ul style="list-style-type: none"> <li>● If the DDMs are unloaded from a mainframe environment, all DDMs will be automatically loaded into the library SYSTEM.</li> <li>● If the DDMs are unloaded from any other environment, the name of the library specified as From Library will be used.</li> </ul> Scan: If you specify no library, all DDMs are scanned.  If the Modify Library Names option (see General Load/Scan Options) is set, you can enter a new library name before the actual load operation begins.
Replace	Not applicable to the Scan. If you specify Y (Yes) and a DDM with the same name as the one you are loading already exists in the target environment, the target DDM is replaced.

### Loading/Scanning Adabas FDTs

If you select Adabas FDTs as the type of objects to be loaded or scanned, you can make the following specifications:

Field	Explanation
DBID	The database ID of the Adabas FDT you want to load or scan.
FNR	The file number of the Adabas FDT you want to load or scan.
Replace	Not applicable to the Scan. If you specify Y (Yes) and an FDT with the same DBID and FNR as the one you are loading already exists in the target environment, the target FDT is replaced.

If DBID and FNR are left blank or set to 0, all FDTs on the work file are loaded.

### Loading/Scanning Error Message Texts

If you select Error Message Texts as the type of objects to be loaded or scanned, you can make the following specifications:

Field	Explanation
Message Type	The type of error messages to be loaded or scanned for:  U User-defined error messages N Natural error messages * All error messages
Library	The library to be loaded or scanned. If you specify no library, all libraries are loaded or scanned. You can specify a group of libraries by using asterisk notation (*). You can only specify a library that has been specified as target library with the Unload function. If no target library has been specified, the name of the library specified as From Library is used.  If the Modify Library Names option (see General Load/Scan Options) is set, you can enter a new library name before the actual load operation begins.
Message Number	The range of error message numbers to be loaded or scanned.
Language Code	The language code of the error messages to be loaded or scanned; for valid codes, see the *LANGUAGE system variable in the Natural System Variables documentation.
Replace	Not applicable to the Scan. If you specify Y (Yes) and a message with the same number as the one you are loading already exists in the target environment, the target message is replaced.

**Note:**

For Natural error messages, you need not specify a library, because they are always loaded into the FNAT system file or the error messages subdirectory.

### Loading/Scanning Command Processors

If you select Command Processors as the type of objects to be loaded or scanned, you can make the following specifications:

Field	Explanation
Library	<p>The library to be loaded or scanned. If you specify no library, all libraries are loaded or scanned. You can specify a group of libraries by using asterisk notation (*). You can only specify a library that has been specified as target library with the Unload function. If no target library has been specified, the name of the library specified as From Library is used.</p> <p>If the Modify Library Names option (see General Load/Scan Options) is set, you can enter a new library name before the actual load operation begins.</p>
Object Name	<p>The name of the command processor to be loaded or scanned. See also Name and Range Specification in the section Introduction.</p>
Replace	<p>Not applicable to the Scan. If you specify Y (Yes) and a source with the same name as the one you are loading already exists in the target environment, the target source is replaced.</p>

### Loading/Scanning Libraries

If you select Library as the type of objects to be loaded or scanned, you can make the following specifications:

Field	Explanation
Library	<p>The library to be loaded or scanned. If you specify no library, all libraries are loaded or scanned. You can specify a group of libraries by using asterisk notation (*). You can only specify a library that has been specified as target library with the Unload function. If no target library has been specified, the name of the library specified as From Library is used.</p> <p>If the Modify Library Names option (see General Load/Scan Options) is set, you can enter a new library name before the actual load operation begins.</p>
Load/Scan Natural Objects	<p>Indicates whether the Natural objects unloaded for the specified library are to be loaded or scanned.</p> <p>Enter a Y (Yes) to load Natural objects.</p>
Range of Natural Objects	<p>The name of the object to be loaded or scanned. See also Name and Range Specification in the section Introduction.</p>
Types of Natural Objects	<p>The type(s) of the object(s) to be loaded or scanned.</p>
Load/Scan Maps	<p>Indicates whether the maps unloaded for the specified library are to be loaded or scanned.</p> <p>Enter a Y (Yes) to load maps.</p>
Range of Maps	<p>The name of the map to be loaded or scanned. See also Name and Range Specification in the section Introduction.</p>
Load/Scan DDMs	<p>Indicates whether DDMs are to be loaded or scanned.</p> <p>Enter a Y (Yes) to load DDMs.</p>
Range of DDMs	<p>The name of the DDM to be loaded or scanned. See also Name and Range Specification in the section Introduction.</p>
Load/Scan Error Messages	<p>Indicates whether the error message texts unloaded for the specified library are to be loaded or scanned.</p> <p>Enter a Y (Yes) to load or scan error messages.</p>
Message Number	<p>The range of error message numbers to be loaded or scanned.</p>
Language Code	<p>The language code of the error messages to be loaded or scanned. For valid codes, see the system variable *LANGUAGE in the Natural System Variables documentation.</p>
Load/Scan Command Processors	<p>Indicates whether the Command Processors unloaded for the specified library are to be loaded or scanned.</p> <p>Enter a Y (Yes) to load or scan for DDMs.</p>
Range of Command Processors	<p>The name of the command processor to be loaded or scanned. See also Name and Range Specification in the section Introduction.</p>
Replace	<p>Not applicable to the Scan. If you specify Y (Yes) and a source with the same name as the one you are loading already exists in the target environment, the target source is replaced.</p>

## Loading/Scanning All Objects

### Scan

If you select All Objects (that is, if you want to scan for all objects on the work file), no further parameters need to be specified.

### Load

If you select All Objects for loading (that is, if you want to load all objects contained on the work file), you can specify the following:

Field	Explanation
Replace	If you specify Y (Yes) and an object with the same name as the one you are loading already exists in the target environment, the target object is replaced.

All objects on the work file are loaded into the target libraries as specified with the Unload function, unless you have set the Modify Library Names (see General Load/Scan Options) option. If so, a second window is displayed, in which you can enter a new library name.

For more information on defining work files, see the section Load Work File Specifications.

## Loading in Batch Mode under OS/390

The following example shows a JCL procedure which you can use to load your programs and maps:

```

//*****
//*
//*   LOAD NATURAL SOURCES
//*
//*****
//TRANSFER  JOB CLASS=G,MSGCLASS=X
//NATBAT    EXEC PGM=NATBAT,REGION=3000K,
//          PARM='IM=D,MADIO=0,MT=0'
//STEPLIB   DD DISP=SHR,DSN=NATURAL.LOAD
//          DD DISP=SHR,DSN=ADABAS.LOAD
//CMPRINT   DD SYSOUT=X
//DDCARD    DD *
ADARUN     PROG=USER, DB=010,MODE=MULTI,SVC=249
//CMWKF01   DD DISP=SHR,DSN=WORK.FILE1
//CMSYNIN   DD *
SYSTRANS
L
N,Y,N,N,Y,N
*
N
NEWLIBS
FIN
/*

```

### Note:

As shown in the above example, do not specify the S/C Type field when loading Natural objects, because the S/C Type field is an output field only.

## User Exits for Loading in Batch Mode

Two user exits with which you can handle errors when loading in batch mode are provided in source form under the names SLD-S-X1 and SLD-S-X2. To be invoked, both must be available as cataloged objects under the names SLDEXIT1 and SLDEXIT2 in the library SYSTRANS.

SLDEXIT1 is invoked if an error occurs that leads to an abnormal termination. It allows you to define a return code.

SLDEXIT2 is invoked in the case of error messages or warnings. If it returns a non-zero return code, the load operation is abnormally terminated; otherwise processing is continued.

## Load Work File Specifications

The following work files are used for loading:

Work File 1	The file from which the data are loaded.
Work File 4	The file into which the load report is written when using the Direct Transfer Functions.

Use the following JCL parameter values to specify the work file:

Parameter	Value
LRECL	96
RECFM	VB
BLKSIZE	6240

## Natural Parameter Settings

To be able to use the Load function, set the following Natural parameters:

Parameter	Value
MT	0
MADIO	0
MAXCL	0

## Restart Load

You use the Restart Transfer Load function to resume a load operation that has been terminated abnormally.

To be able to use this function, the load option Save Restart Information must have been set (see General Load/Scan Options). If so, the necessary restart information (that is, the work file header and all relevant load parameters) is saved in a Natural text object called LOAD-LOG.

When you invoke the Restart Transfer Load function, you first have to specify the following general restart options:

Use Entire Connection Work File	Enter a Y (Yes) if the data to be loaded are to be uploaded from Work File 7. When using this option, Entire Connection is required, and Work File 7 must have been defined as Entire Connection work file in your parameter file.
---------------------------------	--

The Restart Load function checks whether the work file header stored as part of the restart information corresponds to that of the work file to be (re)loaded.

- If so, the Restart Load function resumes the previously terminated load operation.
- If not, the Restart Load function cannot be performed and a corresponding message is displayed. The same applies if you select the Restart Load function without the Save Restart Information option set to Y (Yes).

# SYSTRANS - Direct Transfer Functions

With the Direct Transfer Functions, you can unload and load source-code objects by using the Natural RPC (Remote Procedure Call) facility and the SYSRPC Utility as described in the relevant documentation.

When you invoke the Direct Transfer Functions, you can select the following:

- Direct Transfer using RPC
- Restart Direct Transfer
- Get Report from Direct Transfer Load
- Define Local TRANSFER System
- Preparing to Use Natural RPC for a Direct Transfer

## Direct Transfer using RPC

A direct transfer starts a remote server, which unloads local data to a remote work file (Work File 1) and then loads the data into the corresponding remote environment.

Before you select the objects to be transferred on the Unload Objects menu, you have to specify the following direct transfer options for defining your transfer session with Natural RPC:

Option	Explanation
Direction	An output field that specifies the transfer direction (for future use).
Remote TRANSFER System Number	A four-digit code ( <i>nnnn</i> ) identifying the location of the remote subprogram TRPC <i>nnnn</i> , which is to perform the transfer service.  <b>Note:</b> So that you can perform a remote transfer, a client stub of the same name as the subprogram TRPC <i>nnnn</i> must be available; see the section Preparing to Use Natural RPC for a Direct Transfer.
Unload Options	For the individual unload options, refer to the General Unload Options in the section Unload Function.
Load Options	For the individual load options, refer to the General Load/Scan Options in the section Load, Scan and Restart Load Functions.  <b>Note:</b> As Software AG'S EntireX Broker is involved when using Natural RPC, data conversion is automatically done by the EntireX Broker and not by SYSTRANS. Therefore, selecting the option "Load with User Conversion Table" means that in addition to the EntireX Broker data conversion, a table is used to replace certain characters by others. This table (SULAS-AS or SULEB-EB) is part of the program SULCONV located in your remote environment.
Replace	If you specify Y (Yes) and an object with the same name as the one you are loading already exists in the remote target environment, the target object is replaced.

Once you have specified the direct transfer options, the Unload Objects menu (see Objects to be Unloaded in the section Unload Function) is displayed, where you can select the objects to be unloaded.

As the load operation starts automatically after the unload operation is finished, the Load function is not explicitly invoked and you therefore have to specify the individual load parameters before you start unloading your objects. Therefore, when you return to the Unload Object menu, Load (instead of Exit) is assigned to PF3, which means that pressing PF3 both ends the unload session and starts the load operation.

## Restart Direct Transfer

You use the Restart Direct Transfer function to resume a direct transfer that has been terminated abnormally. When you select this function, you first have to specify your Remote TRANSFER System Number (see Direct Transfers using RPC).

To be able to restart a remote transfer:

- A client stub of the same name as the subprogram TRPC $n$  must be available; see Preparing to Use Natural RPC for a Direct Transfer.
- The direct transfer load option Save Restart Information (see General Load/Scan Options) must have been selected. Otherwise, the direct transfer cannot be resumed and a window is displayed showing you a corresponding message and asking you whether to perform the entire load operation again. If so, the Direct Transfer Reload window is displayed, in which you can again specify the relevant load options as well as the Replace option.

See the Restart Load function for further information on restarting a data transfer.

## Get Report from Direct Transfer Load

You use the function "Get Report from Direct Transfer Load" to obtain the load report, which is written to Work File 4 in the remote environment.

If you select this function, a window is displayed in which you enter your Remote TRANSFER System Number (see Direct Transfers (using RPC)). Then, press ENTER to have the contents of Work File 4 displayed on a screen. It displays a listing of the objects that have been loaded.

## Define Local TRANSFER System

This function is only available in environments which support the Natural RPC servers.

You use the Define Local TRANSFER System function to specify your own TRANSFER system number with which you can be addressed as a server by other clients. You can define several servers (that is, TRANSFER systems), but only one system number per server.

When you select this function, you have to specify a four-digit Local TRANSFER System Number ( $mmmm$ ) along with the DBID (database identification) and FNR (file number) of the target system file. The subprogram TRPC $m$  is then copied into the library SYSTEM on the specified system file.

## Preparing to Use Natural RPC for a Direct Transfer

### To use Natural RPC to perform a direct transfer

1. Define your TRPC $n$  subprogram in the service directory.
2. Generate the corresponding client stub and copy the generated stub in the library SYSTRANS.  
The client stub must have the same name as the TRPC $n$  subprogram; the parameters to be passed are provided in the parameter data area TRPCPDA, which is delivered in source form.  
For further information, see Creating Stub Subprograms (Natural RPC documentation) and the SYSRPC

Utility documentation.

3. Start the corresponding server.

# Direct Commands and CALLNAT Interface

- Executing SYSTRANS in Direct Command Mode
- Executing SYSTRANS from within an Application
- Available Keywords and Values

## Executing SYSTRANS in Direct Command Mode

In direct command mode, SYSTRANS functions are invoked by the command TRANSCMD (provided in the library SYSTEM) using the following syntax:

```
TRANSCMD function object parameters options
```

Multiple direct command strings can be specified with TRANSCMD. The individual parts of a SYSTRANS command string are defined by keywords. The corresponding values can be entered in upper or lower case. The sequence of parameters and options is variable, since individual values are identified by keywords, and not all keywords must be specified (see Available Keywords and Values).

### Example 1:

If TRANSCMD is issued from the Natural command line (or via the Natural stack), and all keywords and corresponding values directly follow the command name in the same input line, the keywords and values must be separated either by blanks or by the input delimiter (defined by the Natural profile parameter ID).

Example of issuing TRANSCMD from the Natural command line or via the Natural stack (assuming ID=','):

```
TRANSCMD LOAD NAT-OBJECT LIBRARY mylib NAME pgm*
```

or:

```
TRANSCMD LOAD,NAT-OBJECT,LIBRARY,mylib,NAME,pgm*
```

Example of issuing TRANSCMD via the Natural stack from within a Natural program (assuming ID=','):

```
STACK TOP COMMAND 'TRANSCMD LOAD,NAT-OBJECT,LIBRARY,mylib,NAME,pgm*'
END
```

### Example 2:

If TRANSCMD is issued in batch mode (or via the Natural stack), and keywords and corresponding values do not directly follow the command name in the same line, the keywords and values must be separated by the input delimiter character (defined by the Natural profile parameter ID).

Example of issuing TRANSCMD via the Natural stack from within a Natural program (assuming ID=','):

```
STACK TOP DATA 'LOAD,DDM,LIBRARY,mylib,NAME,ddm*'
STACK TOP DATA 'LOAD,NAT-OBJECT,LIBRARY,mylib,NAME,pgm*'
STACK TOP COMMAND 'TRANSCMD'
END
```

Example of issuing TRANSCMD in mainframe batch mode (assuming ID=','):

```

TRANSCMD
LOAD,NAT-OBJECT,LIBRARY,mylib,NAME,pgm*
LOAD,DDM,LIBRARY,mylib,NAME,dm*

```

## Executing SYSTRANS from within an Application

To invoke functions of SYSTRANS from within a Natural application, the Natural subprogram TRANSIF is provided in the library SYSTEM. Since only one SYSTRANS function can be executed at a time, subsequent calls to TRANSIF are necessary if several commands are to be processed, for example, when unloading.

The required SYSTRANS command syntax is as follows:

```
function object parameters options
```

The individual parts of a SYSTRANS command are defined by keywords. The keywords and corresponding values must be separated by blanks. Values can be entered in upper or lower case. The sequence of parameters (see Parameters below) and options (see Options below) is variable, since individual values are identified by keywords, and not all keywords must be specified.

The parameter data used in TRANSIF are described in the following table:

Parameter	Format/Length	Explanation
Command_Line_1	A253	First part of the command string
Command_Line_2	A253	Second part of the command string
Command_Line_3	A253	Third part of the command string
Command_Line_4	A253	Fourth part of the command string
Command_Line_5	A253	Fifth part of the command string
First-Call	L	Must be <b>true</b> for the first call of an UNLOAD command
P-NUMBERS		Returns the number of processed objects
Num_COPY	N5	Copycode
Num_CP	N5	Command processors
Num_DDM	N5	DDMs
Num_ERR	N5	Error messages
Num_FDT	N5	FDTs
Num_GLOB	N5	GDAs
Num_HELP	N5	Helproutines
Num_LOC	N5	LDAs
Num_MAP	N5	Maps
Num_NAT_OBJ	N5	Total number of Natural objects
Num_PAR	N5	PDA's
Num_PRED_RLS	N5	Rules
Num_PROG	N5	Programs
Num_SUBP	N5	Subprograms
Num_SUBR	N5	Subroutines

Parameter	Format/Length	Explanation
Num_TEXT	N5	Text
Num_TOT	N5	Total number of objects
Num_CLASS	N5	Classes
Num_Dialog	N5	Dialogs
Return_Message	A253	Returned message text; always filled.
Return_Code	I4	Return code. The following codes are possible:  0        No warning or error occurred. -4870    Error in command; see Cmd_Return_Code. any other negative values are Natural errors.  Any other values are error messages issued by SYSTRANS.
Cmd_Return_Code	I4	Return code from command analysis:  0    No warning or error occurred 1    No SYSTRANS command 2    Command too long 3    Invalid function 4    Invalid object 5    Invalid entry in command 6    Value too long 7    Invalid value (see message) 8    UNLOAD was not followed by END 9    Options in secondary UNLOAD 10   Function rejected by user exit
EXTENSIONS	(A1/1:V)	For future use.

An example of how to call TRANSIF can be found in the program DOCIF in library SYSTRANS.

## Available Keywords and Values

The following tables include all available keywords and values for the execution of SYSTRANS functions in direct command mode, that is, by using the direct command TRANSCMD or by using the Natural subprogram TRANSIF.

Depending on the specified function (see Functions below), keywords are either mandatory or optional. In the latter case, they are included in brackets [ ]. Keywords separated by a "|" character represent alternatives.

An underlined portion of a keyword represents an alternative for the complete keyword. For example, you can specify EX instead of EXECUTE, or U instead of UNLOAD.

All default values correspond to those described in the corresponding sections of the SYSTRANS interactive usage; see also the sections Unload Function and Load, Scan and Restart Load Functions. They determine the value used if no keyword and value are specified. Keywords that do not have a default value are mandatory.

## Functions

The following functions can be specified:

[EXECUTE]	keyword (can be omitted)
UNLOAD   LOAD   SCAN   RESTART	mandatory values
[END]	keyword

The function block must be specified as the first parameter block of the command string.

If the UNLOAD function is specified, also the keyword END must be specified as last function of the block in order to write the end record to the work file (except if only one UNLOAD command is specified and the ENDREC option is set to Y (Yes); see also Options below). With any other function, END can be omitted in online mode.

## Objects

The following objects can be specified:

[OBJECT]	keyword (can be omitted)
NAT-OBJECT   MAP   DDM   EDT   ERROR-MSG   ALL   COMMAND-PROCESSOR	values

The object block must be specified immediately after the function block, except for the function RESTART. Valid values depend on the specified function.

All values (except ALL) are valid for the functions: UNLOAD, LOAD and SCAN. The value ALL is valid for the functions LOAD and SCAN only.

## Parameters

The following parameters can be specified:

Parameter	Explanation
[WHERE]	Can be used to introduce the parameters block to make the command string more readable.
FROM   FROM-LIBRARY	Source library when unloading Natural objects, maps, DDMs or user messages.
[NAME]	Object name when unloading, loading or scanning Natural objects, maps or DDMs.
[TYPE]	Object type when unloading, loading or scanning Natural objects; message type ( <u>S</u> YSTEM   <u>U</u> SER) when unloading, loading or scanning error messages.
[TO-LIBRARY]	Target library when unloading Natural objects, maps, DDMs or user messages.
[INCORPORATE]	Specifies whether to incorporate rules when unloading maps.
[UNLOAD-RULES]	Specifies whether to unload rules when unloading maps.
[RULE-LOAD]	Specifies whether to load rules when unloading maps.
FDBID	Source database ID when unloading FDTs.
FFNR	Source file number when unloading FDTs.
[TDBID]	Target database ID when unloading, loading or scanning FDTs.
[TFNR]	Target file number when unloading, loading or scanning FDTs.
[PASSWORD]	Password to be specified when unloading FDTs.
[CIPHER]	Cipher code to be specified when unloading FDTs.
[FNUMBER]	Start value when unloading, loading or scanning error messages.
[TNUMBER]	End value when unloading, loading or scanning error messages.
[LANGUAGE]	Message language when unloading, loading or scanning error messages.
[LIBRARY   LIB]	Library when loading or scanning Natural objects, maps, DDMs or user messages.
[NEW-LIBRARY   NEW-LIB]	New library when loading Natural objects, maps, DDMs or user messages.

The parameters block is mandatory with the UNLOAD function (except if the option WORK-FILE-INPUT is set to Y (Yes) in the options block; see Options below). It can be omitted with the LOAD and SCAN functions, and it is not used with the RESTART and END functions. Valid keywords and possible values depend on the specified function.

## Options

The following options can be specified:

[WITH]
[CONVERSION]
[USER-TABLE]
[SUBSTITUTE-REFERENCE]
[REPORT]
[LINE-NUMBERS]
[WORK-FILE-INPUT]
[WF-NAME]
[SPECIAL-CONVERSION]
[TRANSLATE-TO-UPPER]
[SAVE-RESTART-INFO]
[USE-NTC-WORK-FILE]
[REPLACE]
[ENDREC]

SPECIAL-CONVERSION only applies to the Load function and is the equivalent of User-Defined Conversion Table = F (Force).

The options block is optional and can be introduced by the keyword WITH to make the command string more readable. Valid keywords and possible values depend on the specified function and correspond to the SYSTRANS General Options for the individual functions as described in the sections Unload Function and Load, Scan and Restart Load Functions.

When using the UNLOAD function, options are evaluated during the first call only and must not be changed during subsequent calls.

If WORK-FILE-INPUT is set to Y (Yes) when unloading, only one call to unload objects will be issued, and the "\*E" end record will be written to the work file after all UNLOAD commands are executed from Work File 2. Additional UNLOAD commands delete the contents of Work File 1 and overwrite the data.

If REPORT is set to Y (Yes), the report to be displayed will be written to Work File 4, which is overwritten with each call of TRANSIF or TRANSCMD. Work File 4 should therefore not be used for other purposes by your application.

If you want to create a report for subsequent calls, you have to call TRANSIF from within a Natural program that opens Work File 4, for example, with: IF FALSE WRITE WORK 4 RECORD.

# SYSTRANS Profile

You can define a profile for your SYSTRANS utility, both generally and user-specific. To this purpose, Natural is delivered with the text object PROFILE. To activate this profile, you have to resave PROFILE under the name TRANPROF in library SYSTRANS.

If you specify the UNLOAD or LOAD or SCAN functions, SYSTRANS searches for (and invokes) object TRANPROF in library SYSTRANS. In TRANPROF, you can enter general or user-specific profiles with corresponding function parameter defaults. These function parameter defaults are then displayed when you enter the function menu.

This section covers the following topics:

- Listing of Text Object PROFILE
- Parameters in the Text Object PROFILE

---

## Listing of Text Object PROFILE

```
*****
* Application: SYSTRANS (TRANSFER)
* Object:      PROFILE
*****
* Function:    Source of the SYSTRANS Profile TRANPROF
*
*   To activate save as 'TRANPROF' in Library SYSTRANS.
*
*   Used to set defaults for Unload or Load/Scan General Options.
*
*   For possible values see the '[General-Start]' to '[General-End]'
*   block, it defines the options for all users.
*
*   In the '[User-Start uid]' to '[User-End uid]' blocks (where 'uid'
*   is the user ID as contained in the Natural system variable '*USER')
*   it is possible to define options for single users.
*   See example in the '[User-Start UID-EXAM]' to '[User-End UID-EXAM]'
*   block.
*   Notes:
*     - Empty lines or lines starting with '*' or '/*' are ignored.
*     - any text after '/*' is ignored.
*     - The line length must not exceed 250 bytes,
*       in mainframe environment only 90 bytes are allowed
*****
[General-Start]
[Unload-Start]
  Conversion N
  User-Conversion N
  Substitute-Line-References N
  Report Y
  Include-Line-Numbers N
  Work-File-Input N
  Use-Selection-List N
  /* Work-File-Name TRANWRK1   /* remove '/*' to use this option
  Use-NTC-Work-File N
  /* Unload-Library SYSTEM     /* remove '/*' to use this option
    DATA-AREA-FORMAT *       /* *, I or D
[Unload-End]
```

```

[Load-Start]
  User-Defined-Conversion-Table N
  Special-Table N
  Report Y
  Translate-to-Upper-Case N
  Save-Restart-Information N
  Modify-Library-Names N
  /* Work-File-Name TRANWRK1 /* remove '/*' to use this option
  Use-NTC-Work-File N
  Load-Library *
  Replace N
    New-Data-Area-Format N /* Y or N
[Load-End]
[General-End]

[User-Start UID-EXAM]
  [Unload-Start]
    Report Y
  [Unload-End]
  [Load-Start]
    Translate-to-Upper-Case N
  [Load-End]
[User-End UID-EXAM]

```

## Parameters in the Text Object PROFILE

The general SYSTRANS profile can be set between [General-Start] and [General-End]. Settings entered here apply to all users. User-specific settings can be entered between [User-Start UID] and [User-End UID], where UID is the Natural user ID (\*USER).

Generally and for a user, you can set parameters for the UNLOAD function. You enter these parameters between [Unload-Start] and [Unload-End]. For the LOAD and SCAN functions, you enter the parameters between [Load-Start] and [Load-End]. The parameters for the UNLOAD and LOAD/SCAN functions are described in detail below.

### Unload Parameters

Parameters for the general UNLOAD options:

Parameter	Possible Value	
Conversion	Y (Yes)	N (No)
User-Conversion	Y	N
Substitute-Line-References	Y	N
Report	Y	N
Include-Line-Numbers	Y	N
Work-File-Input	Y	N
Use-Selection-List	Y	N
Work-File-Name	Not used on mainframes.	
Use-NTC-Work-File	Y	N

Other UNLOAD parameters:

Parameter	Possible Value
Unload-Library	Default library name.

## Load Parameters

Parameters for the general LOAD options:

Parameter	Possible Value	
User-Defined-Conversion-Table	Y (Yes)	N (No)
Special-Table (corresponds to the character version setting "User-Defined Conversion Table=F")	Y	N
Report	Y	N
Translate-to-Upper-Case	Y	N
Save-Restart-Information	Y	N
Modify-Library-Names	Y	N
Work-File-Name	Not used on mainframes.	
Use-NTC-Work-File	Y	N

Other LOAD parameters:

Parameter	Possible Value	
Load-Library	Default library name.	
Replace	Y (Yes)	N (No)