

Entire Operations

User's Guide

Version 5.4.3

December 2017

This document applies to Entire Operations Version 5.4.3 and all subsequent releases.

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

Copyright © 1988-2017 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

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

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

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

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

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

Document ID: NOP-ONOPUSERGUIDE-543-20191101

Table of Contents

Preface	xv
1 About this Documentation	1
Document Conventions	2
Online Information and Support	2
Data Protection	3
I System Overview	5
2 System Overview	7
Operating System Classes and Related Operating Systems	9
Entire Operations User IDs	9
Operating System User IDs	10
Owner	12
Job	13
Job Network	14
Subnetworks	15
Logical Conditions	15
Prerequisite Check	17
Events	23
End-of-Job Checking and Actions	23
Resources	24
Mailboxes	27
Operating System Server Nodes	29
Master Database and Active Database	30
Monitor (Server)	31
Monitor Start Network	33
Activation of Job Networks or Jobs	33
Run Number	38
Schedules	38
Calendars	39
Symbol Tables and Symbols	39
Job Control (JCL)	40
Dynamic JCL Generation (JCL Location MAC)	41
Accessing Entire Operations from other Applications	41
Job Execution as a Dummy Job	42
Logging Facility	42
Message Sending	43
System Messages	43
User Language	44
Reporting	45
Cross References	45
Editor	46
Cleanup of the Active Database	46
II Using Entire Operations	49
3 Online Help	51

Screen Level	52
Field Level	55
Error Messages	57
Online Technical Information	57
4 Starting and Ending an Entire Operations Session	59
5 Logging on and off an Operating System Server Node	61
Logon Function	62
Fields: Node Logon	64
Monitoring the Node Connection Status	65
Logoff Function	66
6 Entire Operations Main Menu	67
Items in the Main Menu	68
Special PF Keys: Main Menu	70
7 Main Screen Elements and Functions	71
Using PF Keys	73
Using Direct Commands	74
Selecting Items from a Menu	74
Filtering Items in a Result List	75
Selecting Items from a Selection Window	76
Confirmation Window when Deleting Items	76
Priority of Command Processing	77
Error Messages	77
8 Date and Time Formats	79
Date Display and Input Options	80
Time Display and Input Options	81
III Entire Operations Utilities	83
9 Entire Operations Utilities	85
Introduction	86
Extract Log Data to Log Selection File	87
Delete Old Data in Log Selection File	88
Output of Log Data to a File	89
Print Account Information from Entire Operations Log	89
Mass Change of Network / Symbol Table Version	90
Monitor or Task Wait Time Modification	91
Monitor Shutdown	91
Monitor Start	92
BS2000 Jobs	92
List or Delete TO-ACTIVATE Command Records	93
Check the Existence of Symbol Table Definitions	94
Mass Change of the Owner / Delete Owner with all Objects	94
Mass Change of Access Rights Granted to Networks	97
User ID Mass Update in Network and Job Definitions	99
Mass Change of Node Numbers	100
Mass Logon Processing in Batch Mode	100
Bulk Execution of MACRO Commands	102

Migration of Log Data from NOP Versions lower than 411 to NOP 412 and above	104
Data Migration to the current Entire Operations Version	106
IV Using Owners	109
10 Using Owners	111
Listing Owners	112
Owner at Logon	113
Linking Additional Owners	114
Owners Granted Access to Individual Networks	114
V Network Maintenance	115
11 Maintaining Job Network Definitions	117
Listing all Network Definitions	118
Selecting a Range of Networks to be Listed	119
Commands: Network Maintenance Screen	120
Displaying and Modifying a Network Definition	121
Adding a Network Definition	122
Fields and Commands: Network Definition	123
Operating System and Environment Defaults	126
Specifying Recipients for Network Messages	133
Granting Definition: Authorizing Other Users or Owners to Access a Network	134
Documenting Your Networks	137
Deleting a Job Network	138
12 Copying Job Network Definitions; Cloning Network Versions	139
Fields: Network Definition Copy	141
13 Displaying the Job Flow within a Network	143
14 Checking for a Loop in a Job Network	147
15 Maintaining the Usage of Network Versions	149
Network Version Usage List	150
Adding or Modifying a Version Usage Definition	151
16 Applying Network Defaults to Jobs (Mass Update)	153
17 Displaying Network Accounting Data	155
Displaying Accounting Data for Jobs in a Selected Network	156
Special PF Key: Job Accounting Data	157
18 Listing Jobs for a Network	159
19 Activating a Job Network Manually	161
Fields: Network Activation	164
VI Job Maintenance	167
20 Defining and Managing Jobs	169
Listing Jobs	170
Displaying and Modifying a Job Definition	174
Adding a Job Definition	181
Writing and Viewing Online Documentation for a Job	183
Deleting a Job Definition	185
Activating a Single Job Manually	185

Displaying Job Dependencies and Connecting or Disconnecting Jobs	188
Defining Job-Specific Log Information	194
21 Defining Job Types and Job Execution Features	197
Available Job Types	198
Defining Job Type Specific Execution Features	200
Defining Operating System Specific Execution Features	204
22 Using a Dummy Job	209
Dummy Job Execution by the Monitor	210
Permanent Dummy Jobs	210
Temporary Dummy Jobs	210
Excluding a Job from Actual Execution	211
End-of-Job Actions after Execution as a Temporary Dummy Job	212
Supervision of Running Jobs	213
23 Defining a Subnetwork	215
Link to the Main Network	217
Subnetwork Activation and Execution	218
Restrictions for Subnetworks	219
Adding or Modifying a Subnetwork Definition	220
Listing Jobs of a Subnetwork	221
24 Defining Parameters for an FTP Job	223
Adding an FTP Parameter Definition	224
JCL Generation for FTP Jobs	227
25 Defining and Managing JCL for a Job	229
Job Control for Jobs under BS2000	230
Job Control for Jobs under UNIX	230
Job Control for Jobs under Windows	231
Defining Master JCL for a Job	234
Displaying Master JCL	238
JCL Locations	239
Defining Operating System Dependent JCL Specifications	243
Handling JCL during Job Submission	246
Symbol Replacement in JCL	250
Importing JCL into a Natural Library	250
Pregenerating Active JCL	253
26 Editing Master JCL and Natural Sources	255
General Editing Considerations	256
UNIX and Windows: Handling of Tab Characters (H'09') within JCL Lines	257
Usage of Text Objects in JCL	257
Using the Editor	257
Editing Macro Sources for Dynamic JCL Generation	261
Locking of Natural Sources	270
27 Defining and Managing Job Input Conditions	271
Listing Input Conditions	272
Adding and Modifying a Master Input Condition	275

Input Condition References	281
Global Conditions	284
Input Condition: User Exit	284
Input Condition: Multiple Suffixes	288
Input Condition: File Existence	291
Input Condition: Mailbox	294
Input Condition: Symbol Value	296
Input Condition: User Switch - BS2000	299
Input Condition: Job Variable - BS2000	301
Listing Jobs Linked to an Input Condition	304
Deleting an Input Condition Definition	306
Displaying Currently Active Conditions	306
28 Handling Prerequisite Resources for a Job	307
Listing Prerequisite Resources Defined for a Job	308
Viewing the Usage of a Prerequisite Resource	309
Displaying, Modifying and Adding a Prerequisite Resource Definition	309
Deleting a Prerequisite Resource Definition	310
Columns and Fields: Prerequisite Resource Definitions	311
VII	315
29 End-of-Job Checking and Actions	317
30 End-of-Job Checking and Actions - Overview	319
List of End-of-Job Checking and Actions	320
Kinds of End-of-Job Actions	321
End-of-Job Checks for Various Operating Systems	322
31 End-of-Job Checking and Actions Screen	323
Invoking End-of-Job Checking and Actions	324
Columns: End-of-Job Checking and Actions	325
Line Commands	326
32 Adding Event Definition for z/OS or z/VSE Job	329
Add/Modify Event Definition	330
Field Descriptions: Add/Modify Event Definition	330
33 Modifying Event Definition for z/OS or z/VSE Job	335
Modifying an Existing Event Definition	336
Example of an A-type Event	337
Example of a C-type Event	338
Example of an R-type Event	339
Example of an S-type Event	340
34 Adding Event Definition for BS2000 Job	341
Adding an Event Definition	342
Field Descriptions	342
35 Modifying Event Definition for BS2000 Job	345
Modifying an Event Definition	346
Example of an S-type Event	347
Example of a U-type Event	348
Example of a V-type Event	349

Adding or Modifying a Job Variable	351
Example of a J-type Event	354
36 Adding Event Definition for UNIX or Windows Job	355
Adding an Event Definition	356
Field Descriptions	356
Exit Code Check Error Handling	358
37 Modifying Event Definition for UNIX or Windows Job	359
Modifying an Event Definition	360
Example of an S-type Event	360
38 Deleting an Event Definition	365
39 EOJ Checking Defaults for Various Operating Systems	367
Common Defaults for Event Checking	368
z/OS Defaults for Event Checking	368
z/VSE Defaults for Event Checking	369
BS2000 Defaults for Event Checking	369
UNIX and Windows Defaults for Event Checking	370
40 Creating Online Documentation for Events	371
41 Adding Output Condition Definitions	373
Adding Output Conditions for an Event	374
Adding Output Conditions for a Job Event	374
Fields and Columns: Output Conditions	375
42 Modifying Output Condition Definitions	379
43 Define Symbol Modification or Setting	381
Modifying a Symbol Modification Definition	382
Symbol Modification Definition Screen	383
Field Descriptions: Symbol Modification	383
Special PF Keys: Symbol Modification	384
44 Displaying Output Condition Use	385
45 Editing End-of-Job Checking or Action User Exits	387
Creating or Editing User Exits	388
Example	388
46 Defining Activation of Jobs or Job Networks	391
47 Defining Recovery Action	395
Recovery Action Definition	396
System Symbols for Recovery Actions	398
48 SYSOUT Actions	401
Defining Job SYSOUT Actions	402
Special PF Keys: SYSOUT Actions	403
49 Message Sending	405
Introduction	406
Defining a Message	406
Fields and Columns: Message and Message Recipients	407
Message Destination Types	408
Message Recipients	409
Special PF Keys: Message and Message Recipients	410

E-Mail on Mainframes	410
E-Mail on UNIX and Windows Systems	411
50 Defining End-of-Job Action User Exit	413
51 End-of-Job Action: Release Resource	417
52 Defining Other Actions	419
Invoking the Other Actions Function	420
Special PF Keys: Other Actions	421
VIII Active Job Networks	423
53 Maintaining Active Job Networks	425
Listing Active Job Networks	426
Deactivating Active Job Networks	428
Repeating Active Job Networks	431
Displaying Operating System Information on Active Tasks	433
Listing Next Network Activations	436
Viewing the Execution History of a Network Active	436
54 Maintaining Active Jobs	437
Listing Active Jobs	438
Listing Jobs of an Active Subnetwork	448
Adding a New Job to the Active Network	448
Viewing and Modifying the Active Jobs Schedule	450
Canceling, Holding and Releasing Active Jobs	453
Resubmitting an Active Job	456
Deactivating a Job in an Active Network	459
Reactivating an Active Job	461
Displaying and Modifying an Active Job Definition	462
Modifying EOJ Checking and Actions	464
Viewing Long Descriptions of Active Jobs	464
Displaying Prerequisites for Active Jobs	465
Viewing and Modifying Resources Used by Active Jobs	467
Modifying Scheduling Parameters	469
Browsing Active Job SYSOUT	469
55 Maintaining Active Job Conditions	475
Listing Active Conditions	476
Viewing an Active Condition and Changing Its Status	480
Adding an Active Condition	480
Deleting Single or Multiple Active Conditions	481
Viewing the Usage of Conditions Concatenated with an Active Job	483
Displaying Jobs Using an Active Condition	484
Viewing Calling Jobs of Subnetworks	485
56 Maintaining Active JCL (Job Control Language)	489
Defining JCL for an Active Job	490
Browsing Active JCL	490
Editing Active JCL	491
Exchanging Active JCL	492
Regenerating Active JCL	492

IX Schedule Maintenance	495
57 General Scheduling Considerations	497
Optional Schedule Definition	498
Schedule Extraction Times	498
Manual and Automatic Activations on the same Day	498
Multiple Network Activations	498
Influence of Deactivations on Schedules	498
Imported Schedules	499
Schedule Dependencies across the Turn of the Year	499
Using Calendars	499
58 Maintaining Schedules	501
Listing Schedules	502
Displaying, Adding or Modifying a Schedule	504
Deleting a Schedule Definition	511
59 Scheduling a Network	513
Defining a Network Schedule	514
Defining and Deleting Time Frames for a Network	517
Defining Dates for Explicit Network Execution	521
Defining Multiple Network Activations	523
Viewing a Network Schedule Definition as a Calendar	525
Producing a Network Start Summary	526
Schedule Modification and Planned Starts	528
Displaying Next Network Starts - Single Network	530
Displaying Next Network Starts - System-wide	535
Displaying the Network Execution History	540
60 Scheduling a Job	543
Defining Scheduling Parameters for a Job	544
Maintaining Schedule Dependencies for a Job	547
Defining Schedule Dependencies for an Input Condition	552
Sending a Late Message to One or More Users	552
Viewing Job Accounting Data	553
X Calendar Maintenance	557
61 Calendar Maintenance	559
General Rules and Restrictions	560
Listing Calendars	561
Displaying or Modifying a Calendar Definition	564
Adding a Calendar Definition	567
Defining Workdays and Holidays	568
Where Used - Listing Schedules Using a Calendar	571
Deleting a Calendar Definition	571
XI Working with Mailboxes	573
62 Working with Mailboxes	575
Viewing Mailbox Messages	576
Turning Automatic Notifications On and Off	578
Fields, Columns and Commands: Messages and Requests	578

Message Types	579
Origin of Mailbox Messages	582
Group Mailbox	582
SYSDBA Mailbox	582
User Mailbox	582
Clearing Mailboxes	583
XII Log Information	585
63 Log Information	587
Displaying Logged Information	588
Output of Log Information	594
Monitoring Entire Operations Activities	596
XIII Symbol Table and Symbol Maintenance	603
64 Purpose and Use of Symbol Tables and Symbols	605
Symbol Tables	606
Symbols	607
Subnetworks and Recovery Jobs	609
User Exits for User-specific Symbol Maintenance Tasks	609
65 Adding and Maintaining Symbol Tables	611
Listing Master Symbol Tables	612
Copying Master Symbol Tables - Cloning Symbol Table Versions	614
Adding a Symbol Table Master	618
Displaying and Modifying a Master or Active Symbol Table	624
Maintaining the Usage of Symbol Table Versions	627
Deleting a Master Symbol Table	631
66 Finding Symbol Tables Associated with Jobs and Networks	633
Listing Usable Symbol Tables	634
Listing Jobs and Networks that use a Symbol Table Master	636
67 Defining Symbols and Symbol Values	639
Listing Symbols of a Symbol Table	640
Displaying a Symbol Definition	640
Adding a Symbol Master	640
Copying a Symbol within a Symbol Table or to another Symbol Table	641
Modifying Symbols in a Symbol Table	642
Defining Multiple Symbol Values	642
Specifying a Range Check for Numeric Symbol Values	644
Predefined Symbols	645
Validating Symbol Values with a User Exit	653
Global Symbol Modification Exit	654
Symbol Setting triggered by the SYSOUT of a Job	654
Deleting a Symbol from a Symbol Table	655
68 Symbol Replacement	657
Symbol Escape Characters	658
Symbol Prompting during Network or Job Activation	660
Displaying Help Text for a Symbol	663
Canceling Symbol Prompting	664

Repeating Symbol Prompting	664
Entering Long Symbol Values and Modifying Active Symbols	664
Specifying User Exits for Symbol Modification	666
Rules and Restrictions for Symbol Replacement	669
69 Functions for Symbol Replacement	681
Results Returned by a Symbol Function	682
Function !D or ?D - Date from Today Calculation	683
Function !E or ?E - Date from Date Calculation	689
Functions !MV or ?MV and !MM or ?MM - Access to Multiple-Value Symbols	690
Function !TIMN or ?TIMN - Constant Time Values	692
Function !W or ?W - Date Calculation with Reference Calendar	693
XIV Reporting	697
70 Reporting	699
Report Types	700
Generating Online Reports	702
Fields and Columns: Reports	708
Special PF Keys: Reports	711
Determination Date for Report Data	712
Report Output Options	712
User Restrictions for Reports	712
Examples of Reports	713
Generating Batch Reports	734
XV Cross References	749
71 Cross References	751
Types of Cross Reference Reports	752
Generating Cross Reference Reports Online	753
Fields and Columns: Cross References	762
Examples of Cross Reference Reports	766
Generating Cross Reference Reports in Batch	771
XVI API Routines	777
72 Using an API Routine	779
API Usage Rules and Restrictions	780
Invoking the API Routines	781
Example of Use of API Routines	781
73 List of Available Entire Operations API Routines	783
EORUCB1N: Check Use of BS2000 User IDs	784
NOPFB2-N: Generate SYSOUT File Names for BS2000	784
NOPMLA1N: Start and Stop the Monitor Activity Log	787
NOPU--1N: Read Network Available to a Specific User	788
NOPU--2N: Return all Usable Symbol Tables for a Network	789
NOPU--3N: Get Correlation ID for an Activated Network	790
NOPU--4N: Store New Event in Entire Operations System File	790
NOPUAC5N: Activate Job Networks or Jobs	791

NOPUAS1N: Retrieve Numbers of Active Jobs in Defined Status	
Ranges	794
NOPUCN3N: Access Entire Operations Conditions	796
NOPUCS1N: Access Calendars and Schedules	797
NOPUJI4N: Import Operating System Jobs into the Active Queue	801
NOPUJS2N: Job Schedule Inquiry and Modification	802
NOPULW9N: Write Messages to System Automation Tools Log	804
NOPUMI1N: Set/Reset Text Milestones in Master and Active Jobs	805
NOPUMT3N: Expand Message Texts	806
NOPUNI1N: Invalidate Entire System Server Node Table Entries	808
NOPUNX1N: Entire System Server Calls to Access UNIX and Windows	
Files	808
NOPURE2N: Handle Resource Allocations	816
NOPURS1N: Access Entire Operations Master Resources	819
NOPUSN2N: Inquire Calling Job or Called Network for Subnetworks	821
NOPUSP3N: Display Long Texts for Symbol Prompting	822
NOPUST3N: Inquire Network and Job Status, Symbol Table	824
NOPUSY6N: Access Entire Operations Symbols	828
NOPUVI2N: Obtain Entire Operations Version Information	833
NOPUXD1N: Maintain End-of-Job User Exits for a Network	834
NOPUXI0N: Add Input Condition to an Activated Job	835
XVII User Exits	837
74 User Exits	839
What User Exits Can Do	840
Global User Exits	841
Front-end User Exits	841
Common User Exit Parameter Data Area NOPXPL-A	841
Generation of Dynamic JCL and SYSOUT File Names (BS2000 only)	847
User Exits for Resource Master Determination	848
User Exits for Setting Input Conditions	848
User Exits for End-of-Job Checking and Actions	848
User Exits for Symbol Functions	850
User Exits for Validation Checks of Symbol Values	853
XVIII Passing Files to Entire Output Management	859
75 Passing Files to Entire Output Management	861
Basic Requirements for File Transfers to Entire Output Management	862
Listing Files Defined for Entire Output Management	863
Adding and Modifying File Definitions	865
Deleting File Definitions	870
Handing Over SYSOUT and Files to Entire Output Management	870
Defining the Handover of SYSOUT Files from UNIX/Windows to	
BS2000	874
XIX Special Monitor Features and Batch Jobs	877
76 Special Monitor Features and Batch Jobs	879
Monitor Start Network	880

Day Start Network	880
Common Start Network Considerations	881
Cleanup in Batch Mode	881

Preface

This documentation contains all information relevant to the user of the Entire Operations system.

Before reading this document, you should be familiar with the *Concepts and Facilities* documentation, which defines and briefly describes the system entities and facilities referred to in this documentation. For the sake of coherence and clarity, some information contained in the *Concepts and Facilities* documentation is repeated in the section *System Overview* in this documentation. You should read this section before starting to work with Entire Operations.

Basic Functions	
System Overview	Provides an overview of Entire Operations: designations for operating systems, brief description of the components and functions available to use and control a job network.
Using Entire Operations	Describes the main functions of Entire Operations.
Entire Operations Utilities	Describes Entire Operations utilities.
Maintenance Functions	
Using Owners	Provides an overview of the main functions available to view and maintain owners.
Network Maintenance	Explains how to define and maintain job networks.
Job Maintenance	Explains the Job Maintenance facility and how to define different job types and job dependencies, create and use JCL, create online documentation for jobs, define input conditions and specify resources.
End-of-Job Checking and Actions	Describes how to instruct Entire Operations what actions to trigger after a job has terminated.
Active Job Networks	Provides information on the maintenance functions you can perform on active jobs and networks.
Schedule Maintenance	Describes how to define schedule objects, and how to schedule a job network and a job.
Calendar Maintenance	Explains the calendar maintenance facility that can be used to create calendars which are the basis of schedules for jobs and job networks.
Working with Mailboxes	Describes how to display mailbox messages.
Log Information	Describes how to display logged information and monitor Entire Operations activities.
Symbol Table and Symbol Maintenance	Provides information on symbol tables, symbol maintenance and symbol replacement functions.
Reporting	Explains the Entire Operations reporting facility, which makes a number of different reports available: information at the job and event level, information on job network definitions and schedules and overviews of schedules and planned activations.

Basic Functions	
Cross References	Describes functions that provide information about the use of objects in Entire Operations. In addition, the symbol search by value is explained.
API Routines	Provides information on the Application Programming Interface (API) which resides in the Entire Operations library and which is a set of routines that you can invoke from any other Natural application in order to access the internal data of Entire Operations.
User Exits	Explains how to work with user exits.
Passing Files to Entire Output Management	Explains how to pass files to Entire Output Management for output processing.
Special Monitor Features and Batch Jobs	Describes how to define a job network to be executed after each monitor start and before the activation of any other job and the cleanup in batch mode.

1 About this Documentation

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

Document Conventions

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

Online Information and Support

Software AG Documentation Website

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

Software AG Empower Product Support Website

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

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

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

To submit feature/enhancement requests, get information about product availability, and download products, go to [Products](#).

To get information about fixes and to read early warnings, technical papers, and knowledge base articles, go to the [Knowledge Center](#).

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

Software AG TECHcommunity

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

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

Data Protection

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

I System Overview

2 System Overview

- Operating System Classes and Related Operating Systems 9
- Entire Operations User IDs 9
- Operating System User IDs 10
- Owner 12
- Job 13
- Job Network 14
- Subnetworks 15
- Logical Conditions 15
- Prerequisite Check 17
- Events 23
- End-of-Job Checking and Actions 23
- Resources 24
- Mailboxes 27
- Operating System Server Nodes 29
- Master Database and Active Database 30
- Monitor (Server) 31
- Monitor Start Network 33
- Activation of Job Networks or Jobs 33
- Run Number 38
- Schedules 38
- Calendars 39
- Symbol Tables and Symbols 39
- Job Control (JCL) 40
- Dynamic JCL Generation (JCL Location MAC) 41
- Accessing Entire Operations from other Applications 41
- Job Execution as a Dummy Job 42
- Logging Facility 42
- Message Sending 43
- System Messages 43
- User Language 44
- Reporting 45
- Cross References 45

- Editor 46
- Cleanup of the Active Database 46

This section also introduces you to the Entire Operations components and facilities you can use to control and monitor the system.

In general, all components and facilities are available for all operating systems supported by Entire Operations. Exceptions and platform differences are described in this section, where relevant.

Before full control of batch processing can be passed to Entire Operations, certain objects must be defined to the system. This section gives a brief description of these objects and how Entire Operations uses them.

See also:

- *Performance Considerations* in the *Installation and Operations* documentation
- *Concepts and Facilities* documentation
- [User Exits](#)

Operating System Classes and Related Operating Systems

Within Entire Operations, the term “operating system class” means one or more operating systems, which are usually handled in the same way.

Operating System Class	Operating System
B	BS2000
M	z/OS
V	z/VSE
X	All supported UNIX operating systems including AIX, HP-UX, Linux and Sun Solaris
W	All supported Windows operating systems

Entire Operations User IDs

In Entire Operations, a user ID can be used to enter the system. Entire Operations user IDs should, but need not be defined to the host TP monitor.

Several users can log on to Entire Operations with the same user ID and password at the same time. For reasons of data security and in order to trace data modifications, however, each user usually has a personal user ID and password.

Entire Operations user IDs are relevant to the following:

■ Entire Operations User Profiles

Each Entire Operations user ID can have individual access rights to Entire Operations functionality and Entire Operations objects. For details, see *User Definitions and Profile Settings* in the *Administration* documentation.

■ Mailboxes

A user ID can be associated with up to ten mailboxes through which the user is notified of any pending logical conditions linked to those mailboxes (see the section [Mailboxes](#));

■ Logging

Entire Operations logs all activities and events occurring within the system, including user activities.

A user ID always has a link to at least one owner (see the section [Owner](#)).

Operating System User IDs

This document covers the following topics:

- [Working with Entire System Server Nodes](#)
- [Logon to an Operating System User ID](#)
- [Operating System User ID, Group, Domain](#)
- [Default User ID Determination](#)

Working with Entire System Server Nodes

If you want to work with operating system objects (e.g. editing JCL), you must perform Entire System server logons to the nodes you want to work with. After such a logon, you have access the access rights of the operating system user ID you specified. See the section [Logging on and off an Operating System Server Node](#).

For [Entire Operations networks](#) and [jobs](#), you must define operating system user IDs specifically as JCL user IDs and submit (execution) user IDs. See also [Defining Operating System Dependent JCL Specifications](#) in the section *Job Maintenance*.

Logon to an Operating System User ID

If you want to work with an operating system object and if you are not logged to the defined Entire System Server node, the node logon screen is presented automatically in many cases.

You may also perform an explicit node logon, by using the `LOGON NODE` direct command.

To view your current node logon status, use the direct command `STATUS NODES`.

Operating System User ID, Group, Domain

In network and job definitions, it is possible to specify

- a JCL user ID for the **JCL node**;
- a submit (execution) user ID for the **execution node**.

Additionally, it is possible to specify a group (UNIX), respective a domain (Windows).

If no group is defined for an UNIX node, the user ID's default group will be in effect.

If no domain is specified for a Windows node, the user ID is treated as a local user. If you enter the node's host name in the domain (group) field, the user ID is treated as local user too.

Default User ID Determination

- [Determination Rules](#)
- [Search Hierarchy for Submit User IDs](#)
- [Symbol Replacement](#)

Determination Rules

If no operating system user ID definition is made for **JCL node** or **execution node** locally, Entire Operations determines an operating system user ID, depending on

- The global submit security user type;
- The node-specific submit security user type.

For detailed information, see the relevant sections in the *Administration* documentation.

Search Hierarchy for Submit User IDs

If an operating system user ID other than the user ID of the Entire Operations Monitor (**Submit Security User Type** = M/User ID of the Entire Operations Monitor) is to be used, a search hierarchy for the operating system user ID is in effect. The fields **Monitor User ID** and **Submit Security User Type** are described in *Monitor Defaults - General* in the *Administration* documentation.

The search order is:

1. The job's (JCL or submit) user ID;
2. The network's (JCL or submit) user ID;
3. The node's default user ID (mainframe, UNIX and Windows);
4. The ID of the user who last modified the job.

Symbol Replacement

This applies to the network master definition, job master definition and job active definition.

Symbol replacement is possible in the fields:

- JCL User ID
- JCL Group
- Submit User ID
- Submit Group

If the activation escape character is used, the replacement is performed at activation time. This is required for JCL user ID and group. If the submission escape character is used, the replacement is performed before job submission. Symbol replacement errors in one of these fields are treated as permanent errors.

Owner

Entire Operations provides ease of use and enhanced security through the concept of owners. This involves dividing job networks into groups by assigning them to an owner name. The system administrator assigns an owner name to a user ID in the User Maintenance facility (see *User Maintenance* in the section *System Administrator Services* of the *Administration* documentation). This owner name is automatically passed to the job network defined by the user.

An owner can thus represent a department or project, or a group of related job networks. Users belonging to a specific owner can perform functions only on those job networks associated with the same owner.

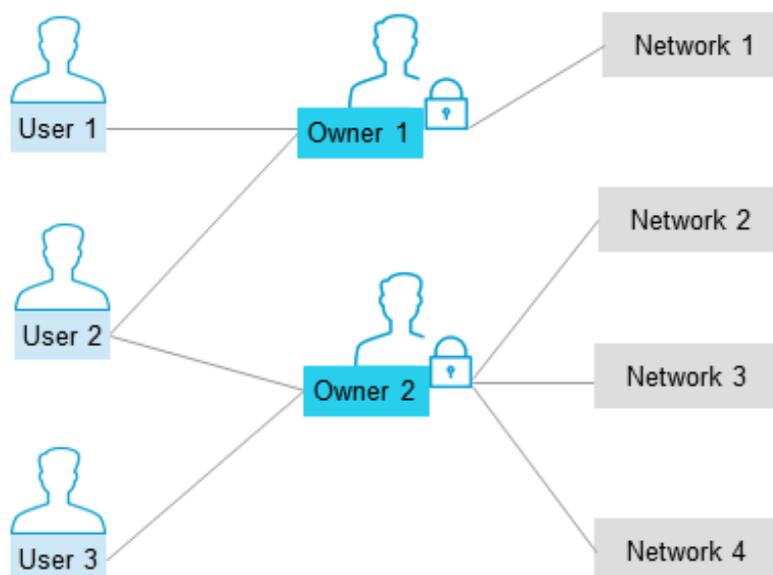


Note: In special cases, a user can be authorized to access networks belonging to other owners. The owner SYSDBA is authorized to access the networks of all owners.

Any number of job networks can be associated with any one owner. A job network name is unique within the System only in combination with its owner name.

Access to most other Entire Operations objects is also owner-dependent. The owner name appears in the top line of most system screens and windows.

The figure below illustrates the links between users, owners and job networks:



Job

In the **Job Maintenance** facility of Entire Operations, various **job types** can be defined.

All jobs are members of job networks and can be linked by logical conditions. Some differences arise in End-of-Job checking, depending on job type and operating system (see the section *End-of-Job Checking and Actions*). However, you can always define **Job OK** or **Job not OK** as a condition for subsequent system action.

For z/OS and z/VSE: An operating system job can consist of several steps. In these cases, Entire Operations can check the result of each job step as part of End-of-Job analyses and triggers system action accordingly.

A job is uniquely identified within a job network by its job name. The job name can, but need not be the same as the JOB or LOGON statement name (job name by which the operating system identifies the job). Before job submission, jobs can therefore only be identified by the name defined to Entire Operations. A job can only be accessed through Entire Operations by its Entire Operations name.

When defining a job, you must also specify:

- JCL location (depending on job type);
- JCL and **execution nodes** (if different from those, specified for the job network);
- JCL and submit user IDs;
- Scheduling parameters (optional, otherwise network default used);
- End-of-Job checking and End-of-Job action specifications (see the section *End-of-Job Checking and Actions* for details).

For further information, see also *Job Maintenance*.



Note: (z/OS only) We recommend that the JCL of one Entire Operations job contain only one job statement. Entire Operations retains only the first assigned job number of a submitted job.

Job Network

A job network is a group of jobs that stand in defined relation to each other. This relation is composed of dependencies, which are expressed as logical conditions. In the simplest case, two jobs in a job network can be linked by the condition: If Job 1 finishes OK, start Job 2 (see *Logical Conditions*).

A job network is uniquely defined by its owner and network name. Each network is given a start and deadline time which determine when the network is to be activated. If your installation includes multi-CPU support, you can also specify a default node name for the jobs in the network. This node name can be overridden at the job level (see *Operating System Server Nodes*).

A user can only access a defined job network if his or her user ID is associated with the same owner as the network, unless he has special authorization to access other networks.

A job network and a single job are the units of work that can be activated by Entire Operations. When a job network is activated, it is automatically given a run number that uniquely identifies this network activation. This feature allows several copies of the same job network to run simultaneously.

A job network can be a **subnetwork** of another job network.

Subnetworks

A job of the type [NET \(Subnetwork\)](#) enables you to define a subnetwork within a main network. This allows you to build nested networks. The subnetwork must already exist when the definition is created. The same subnetwork can be defined in different jobs of the main network. On activation, each active subnetwork is assigned a unique run number. Subnetworks can in turn be invoked within subnetworks. However, a subnetwork cannot invoke itself, because this could cause an infinite recursion.

For detailed information, see [Defining a Subnetwork](#) in the section *Job Maintenance*.

Logical Conditions

This section covers the following topics:

- [What Are Logical Conditions?](#)
- [Input Conditions](#)
- [Output Conditions](#)

What Are Logical Conditions?

Logical conditions are variables within Entire Operations and describe job dependencies. Condition names must be unique within a job network.

An active condition reflects the current value of the condition for a given job network activation. It can have the value true (the condition exists) or false (the condition does not exist). The **run number** assigned to the job network at activation is automatically passed to the conditions defined for the jobs in the network. An active condition is uniquely identified by owner, network, run number and condition name.

In Entire Operations, logical conditions are used in two roles:

- As input conditions;
- As output conditions.

These are described in more detail below.

Logical conditions can be global. You can have only one global condition per name and system. See also [Global Conditions](#) in the section *Job Maintenance*.

Input Conditions

Input conditions are prerequisites for job submission. Entire Operations does not submit a job until all input conditions and other prerequisites are set (fulfilled). An input condition can be set by the occurrence of an event detected by Entire Operations or manually by the user from the Active Conditions screen in the Job Maintenance facility. It can also be set by a reply to a mailbox request.

If no input condition is defined for a job, Entire Operations assumes a virtual true input condition. This means that this job can be submitted immediately at the (earliest) starting time defined for it, unless the job has other prerequisites such as resources.

Jobs are linked by defining the output conditions of one job as the input conditions of the subsequent job. For more information, see [Defining and Managing Job Input Conditions](#) in the section *Job Maintenance*. A quick way to link two jobs is to use the connect function, in which Entire Operations provides a default condition as an output condition for one job and as input condition for another (see [Connecting Jobs from the Same or Different Networks](#) in the section *Job Maintenance*).

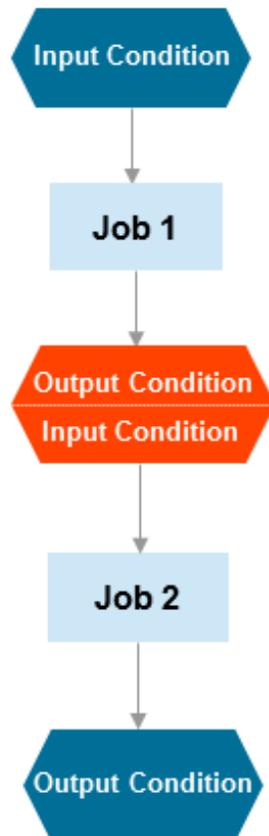
Input conditions can refer not only to the current run of a job network, but also to given time frames in the past or to previous runs.

You can also use an input condition to turn a job into a dummy job when it occurs. For further information on this topic, see [Input Conditions](#) and [Job Execution as a Dummy Job](#).

Output Conditions

Output conditions can be set or reset during End-of-Job checking of Entire Operations. For each job or job step (operating system job), you can specify any number of possible events. Each event can be associated with up to 20 output conditions. When any of these events occur, Entire Operations automatically sets the associated output conditions and starts those jobs which have these conditions as input conditions (see also [End-of-Job Checking and Actions](#)).

The figure below illustrates a simple example of two jobs linked by logical conditions:



To link the two jobs: an Output Condition of Job 1 is defined as an Input Condition for Job 2.

Prerequisite Check

Each active job is checked for its prerequisites, before it can be submitted. Only, if all defined prerequisites are available at the same time, the job can be started. The prerequisite checking of an active job is repeated until all defined prerequisites are available, but only before its latest start time is reached.

The following prerequisites must be met before a job can start running:

- The start and end times defined for a job or network must be reached,
- The input conditions defined for the job must be fulfilled,
- The resources defined for job usage must be available,

- Operating-system specific objects defined for a job (for example, a BS2000 user switch) must be available, and
- The execution node defined for the job or network must be available.

Entire Operations uses several procedures to reduce the effort involved for the prerequisite check. These procedures are transparent to the user. Nevertheless, they are explained in the following section.

- [Order of Prerequisite Checking](#)
- [Passive Wait](#)
- [Prerequisite Check according to the Round-Robin Procedure](#)

Order of Prerequisite Checking

The sort order of prerequisite checking is:

1. Earliest start time,
2. Owner, network, run, job.

The sort is only applied to jobs, which reside within the prerequisite check input queue at the same time.

Passive Wait

Active jobs waiting for one or several input conditions, resources or for the availability of an operating system server (node) are placed into a particular queue, which removes them temporarily from the active check carried out by the Monitor.

Active jobs are woken up (released) from the passive wait:

- During setup or deletion of active prerequisites at any location;
- During setup or deletion of resources at any location;
- After modification or deletion of definitions for input conditions and resources in active jobs;
- During Monitor start;
- During change of the day;
- By explicit request; see *Special Functions* in the *Administration* documentation.

After a wake up, an active check of the prerequisites, resources and operating system server is carried out again. If the prerequisites required for job start are not met, then another passive wait can result out of this.



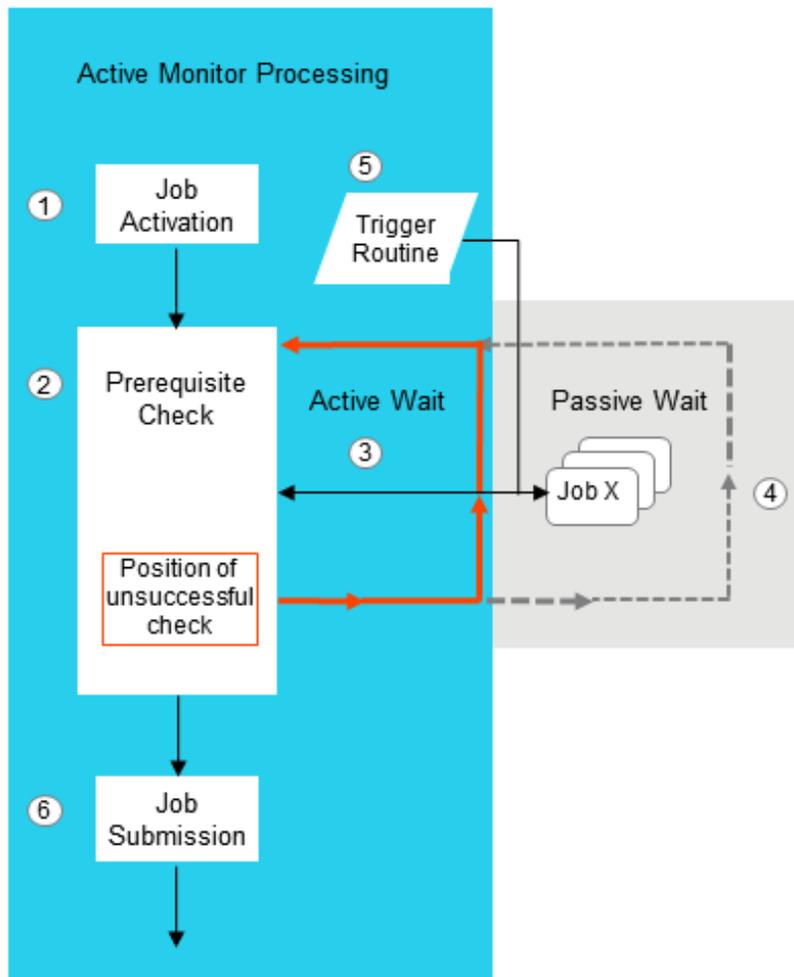
Note: The main passive wait release routine does not re-activate the waiting jobs at the same time. Instead, it performs the release in portions of 300 jobs. Between the portions,

there is a wait of 30 seconds. This spreads the Monitor and database activity for the prerequisite check of a large number of jobs over a longer period of time.

- [Course during Passive Wait for Prerequisites](#)
- [Exceptions from Passive Wait](#)

Course during Passive Wait for Prerequisites

The following graphic shows the course during passive wait for prerequisites:



Legend

- ① A network has been activated and job processing is controlled by the Monitor.
- ② The prerequisites of a job are checked after job activation.
If a prerequisite is not met (for example, the execution node defined for the job is not available), the prerequisite check stops at the position where it failed.
- ③ The job is placed into an active wait state waiting for the next check to meet the required prerequisite.
The next check continues at the position where the previous check failed.
- ④ The Monitor determines how long to wait for the missing prerequisites before it places the job into a passive wait state.
- ⑤ A trigger routine reactivates the job if the criteria defined to reactivate the job are met (for example, the missing execution node is available now), and forces the job back to active checking.
The check procedure (from active to passive wait and vice versa) can repeat several times.
- ⑥ If all prerequisites are met, the job is submitted for execution.



Note: Each time the Monitor is started, all jobs in the passive wait queue are reactivated for another prerequisite check.

Exceptions from Passive Wait

A job cannot be placed into a passive wait state in the following cases:

- Waiting for an input condition which depends upon the existence of a file;
- Waiting for an input condition which depends upon the result of a user exit.

In these cases, Entire Operations cannot acknowledge on its own when such a job is to be placed again into the active wait. Therefore, in such a case, an active job is not placed into the passive wait.

Nevertheless, at least for part of the wait, a passive wait can also be carried out for these jobs, if, in parallel to the above mentioned cases, they are waiting for a *normal* prerequisite, which is set up as shortly as possible before job submission.

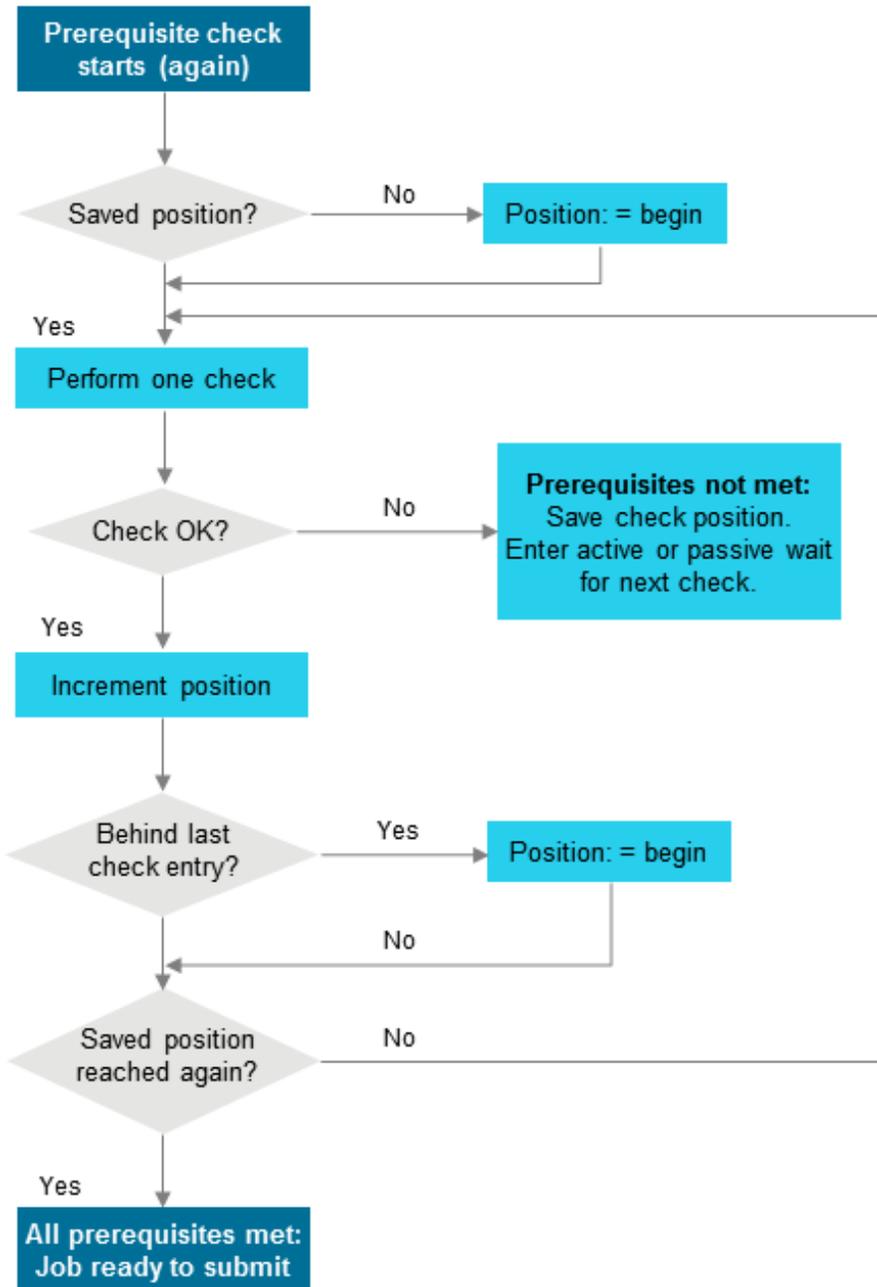
In other words: it is recommended to replace a wait for prerequisites with special dependencies by a wait for normal prerequisites.

Prerequisite Check according to the Round-Robin Procedure

If prerequisites and resources of an active job are actively checked, then the order of the job checks will be optimized dynamically.

For a follow-up check, the last unsuccessful check will be the starting point. This prevents successful checks from being redundantly repeated several times. It is guaranteed, however, that immediately before the job start release all input conditions and resources have been checked together at one point in time.

The following diagram shows the course of the Round-Robin Procedure for the check of prerequisites and resources:



Events

In the terminology of Entire Operations, an event is the occurrence of a defined situation which is recognized during End-of-Job analysis. Entire Operations automatically triggers system action, depending on the occurrence of events during job processing (see the section *End-of-Job Checking and Actions*). Any number of events can be defined for a job.

Some examples of possible defined events are:

- Exit code of a UNIX job equals 2;
- STEP2 of JOB1 ends with a condition code greater than 8;
- No job step ends with a condition code greater than 0;
- A defined message appears in the job SYSOUT;
- A database or file contains or does not contain certain expected data;
- The result of a user exit (expressed by its return code).
- A job variable contains certain expected data (BS2000).

End-of-Job Checking and Actions

End-of-Job actions refer to all actions performed after termination of a job. These actions can be performed automatically by Entire Operations or manually by the user.

End-of-Job checking and actions consists of two steps:

1. Analysis of job results (determination of End-of-Job status);
2. Triggering of appropriate system actions.

Entire Operations recognizes End-of-Job status by the occurrence of events predefined by the user. Such an event can be, for example, any of the events described in the previous section.

The internal result table used during the End-of-Job checking has space for 1200 entries per job.

If you do not specify any event, Entire Operations provides a default event expressed as Job OK or Job not OK, depending on whether a received condition code is greater or less than a default condition code, or, for BS2000, whether certain system messages are received.

For each of the user-specified or default events, you can define how Entire Operations is to act. Such an End-of-Job action can consist of any of the following:

- Set output conditions to continue with job flow;

- Send message to user or console with information about any abnormal event or pending condition;
- Print or cancel job SYSOUT data;
- Pass output files or SYSOUT to Entire Output Management;
- Execute user exit;
- Activate other job networks;
- Perform recovery;
- Set job variable (BS2000 only).

See also the section [End-of-Job Checking and Actions](#) for more information.

Job SYSOUT Check

- On z/OS: The job result check will be retried by the Monitor up to 10 times, when the message Job disappeared from Spool Queue appears.

The wait interval between SYSOUT read attempts is constantly at least 30 seconds (not to be confused with the Monitor wait time, because it may be very short).

- On BS2000: Entire Operations can only check job SYSOUT if it is assigned to a file. JCL of jobs that are to run under Entire Operations control must therefore not contain SYSOUT assignments to **dummy, primary* or to a temporary file, otherwise no End-of-Job checking is possible.

See also [Defining Job SYSOUT Actions](#).

Resources

This section covers the following topics:

- [What Are Resources?](#)
- [Ordering of Resource Allocation](#)
- [Periods of Resource Allocation \(Deallocation Modes\)](#)
- [Resource Release](#)
- [Inhibiting a Resource Deallocation if a Job Ended not ok](#)
- [Manual Emergency Actions](#)
- [Resource Amount Determination by User Exits](#)

See also:

- [Handling Prerequisite Resources for a Job](#) in the section *Job Maintenance*
- [Viewing and Modifying Resources Used by Active Jobs](#) in the section *Active Job Networks*
- *Resources* in the *Concepts and Facilities* documentation

- *Resources* in the *Administration* documentation

What Are Resources?

Resources can be prerequisites for a job. Entire Operations does not submit a job until the amount of resource defined is available. You can thus use resources to further control the job flow when all input conditions for jobs that can run parallel are set. You do this by defining the priority with which resources are allocated to a job.

Resources can be

- Quantitative or absolute;
- Reusable or not reusable.

Some examples of resources are listed below:

Resource	Type
Print forms	Quantitative, not reusable
Main storage	Quantitative, reusable
Line to a remote machine	Absolute
Availability of a device	Absolute

Each resource must be defined in the System Administrator Services as a master resource, before it can be defined as a prerequisite for any job.

The current amount of a master resource can be determined by an exit, which will be invoked periodically by the Entire Operations Monitor. For a detailed discussion, see *Resource Master Determination Exit*.

Resources can reflect real system resources or they can be virtual. Entire Operations monitors resources as defined in the Network and Job Maintenance facility (see [Handling Prerequisite Resources](#) in the section [Job Maintenance](#)).

Ordering of Resource Allocation

The following rules apply for the ordering of resource allocations:

1. If a resource is requested by the same owner, network, job, but different runs (at the same time), the active job with the lowest **run number** or earliest activation time will get the resource first.
2. If several different jobs of one active network or of several active networks wait for the same resource: The available quantity of the resource will be allocated to as many as possible jobs in parallel, but under the restriction of item 1.

If a resource waiter with a higher priority is found during a prerequisite resource check, the message `Res. resource available, other jobs have priority` will be written to the log. It is followed

by Res. *resource* - waiting with higher priority: and one or several lines with context information.

Periods of Resource Allocation (Deallocation Modes)

Usually a resource is allocated for the duration of a job execution. Before Entire Operations, resources could be allocated only in this way.

Starting with Entire Operations, it is possible to allocate resources for a longer period. This can be defined in the [prerequisite resource definition](#) for a job (deallocation mode).

The [resource deallocation modes](#) you can set are described in the section *Job Maintenance*.

Resource Release

Source deallocation will be forced if the resource allocation period is longer than retention period for active conditions.

The location and reason of resource releases is logged:

- During network deactivation;
- During job deactivation;
- During forced freeing of a resource allocation;
- During cleanup.

Inhibiting a Resource Deallocation if a Job Ended not ok

It is possible to inhibit a resource deallocation if a job ended `not ok`. This can e.g. be used to keep a resource during a recovery for an unsuccessful run. The additional rules apply here, too. This means that such a resource will also be released automatically if the allocating job is deactivated.

See the field **Deallocate if Job not ok** in the [Prerequisite Resource Definition \(Master\)](#) window described in [Columns and Fields: Prerequisite Resource Definitions](#) in the section *Job Maintenance*.

Manual Emergency Actions

All current resource allocations can be inquired in usage lists. From these active resource usage lists it is possible to force the deallocation of a given resource allocation.

For further information, see *Listing Jobs Currently Using a Resource* in the section *Resources* in the *Administration* documentation.

Please use this feature with care. Be aware that one or several active jobs may be started immediately, which were way for this resource.

Resource Amount Determination by User Exits

The available amount of a resource can be determined by the usage of an exit. A resource master determination exit (described in the *Administration* documentation) will be invoked by the Entire Operations Monitor before prerequisite resource checks.

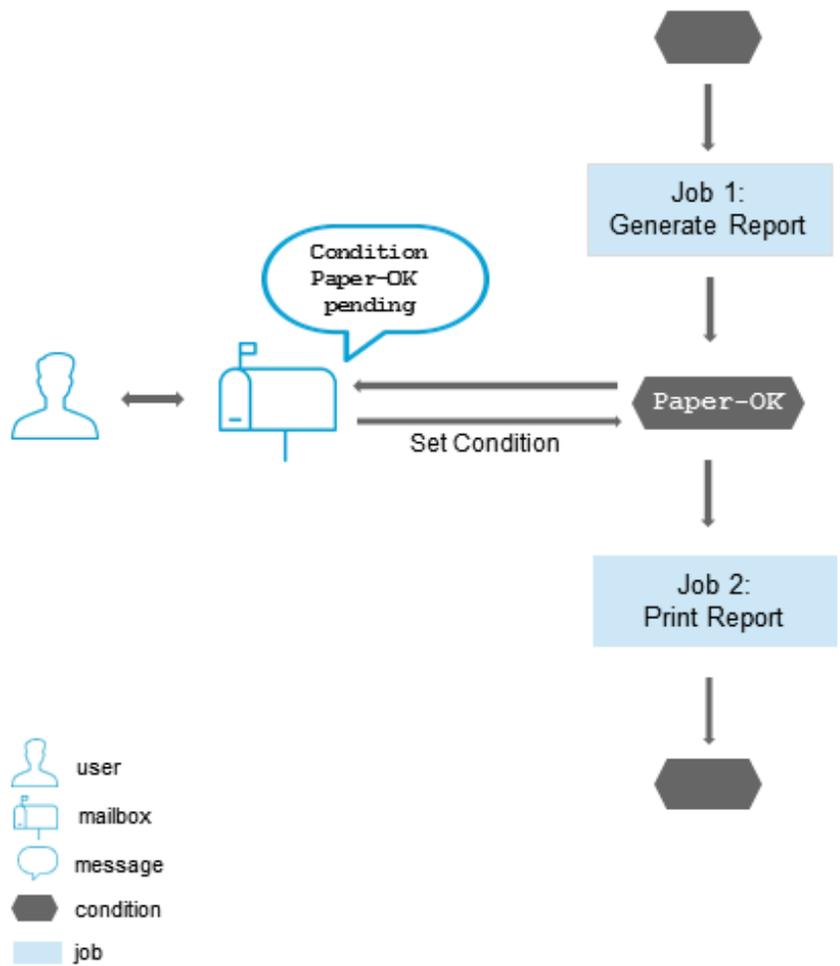
Mailboxes

Mailboxes are used for message sending to Entire Operations users. If a message requires a reply, it can be prompted in the mailbox window.

Any input condition can be assigned to a user interaction. You can send a mailbox request to notify a user to take appropriate steps and manually set the conditions necessary for a job to continue.

The concept of mailboxes thus allows you to integrate manual actions into the job network.

The following figure illustrates an example of a mailbox for paper supply:



The condition Paper-OK is defined as input condition for Job 2.

On receiving the message Condition Paper-OK pending, you can supply the required amount of paper and set the condition manually, directly in the mailbox window at your terminal. Entire Operations can then proceed with the next job (print report).

The mailbox SYSDBA, which is accessible for the owner SYSDBA, contains all messages for which no recipient was defined.

You can find a detailed description of all mailbox features in the section [Working with Mailboxes](#).

Operating System Server Nodes

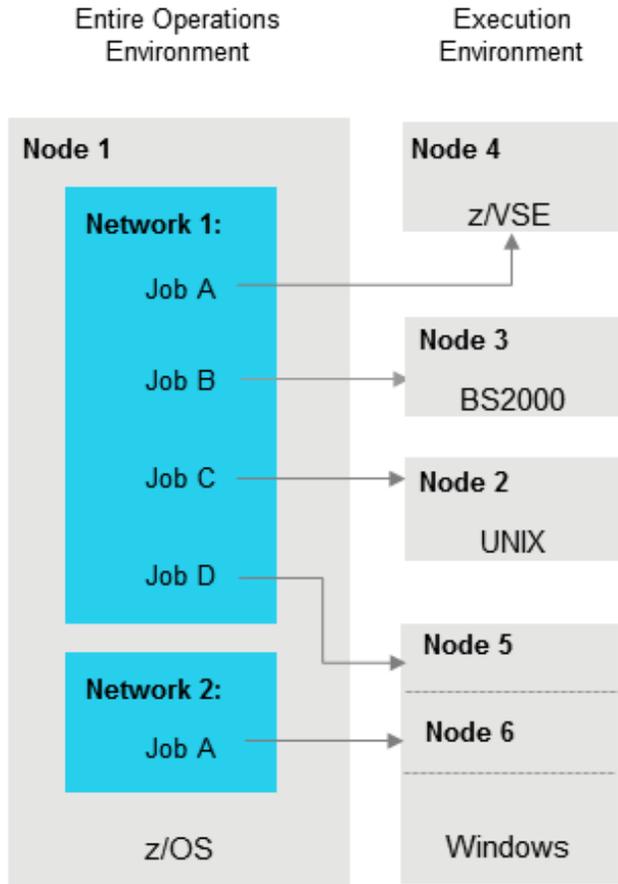
Nodes are Entire System Servers and refer to machines on which requests to the operating system are executed. They are distinguished by numerical identifiers in the same way as database IDs distinguish between different Adabas databases. Within Entire Operations, each machine is assigned a node name. More than one operating system server node can reside in one physical machine.

The machines identified by node names can run different target operating systems. Entire Operations recognizes the operating system.

Communication paths between otherwise isolated nodes are provided by the Software AG products Entire Net-work and EntireX Broker, which allow a transparent connection of nodes, irrespective of how they are physically linked.

When defining a job network in Entire Operations, you can specify default nodes for the JCL and execution of all jobs in the network. These default nodes can be overridden for any job, so that different jobs within the same network can run on different machines.

The following example illustration shows how Entire Operations can support servers and nodes on different machines and operating systems:



For more information, see *Definition of Nodes* in the *Administration* documentation.

Master Database and Active Database

This section covers the following topics:

- [Master Database](#)

- [Active Database](#)

Master Database

The master database stores all user, job network, job and scheduling definitions. It also contains all information pertaining to defined logical conditions, resources, calendars, and symbol tables. All information stored on the master database can be maintained online.

For further information, see *Master Database* in the *Concepts and Facilities* documentation.

Active Database

When a job network is activated, it is copied to the active database. The active database can contain several copies of the same job network, each distinguished by its unique **run number**. All current information pertaining to condition status, job status, active JCL and symbols is contained in and can be modified on the active database.

The master and active databases are normally located within the same physical DB file.

For further information, see:

- *Active Database* in the *Concepts and Facilities* documentation
- Direct command `SET ACTIVE` (sets the selection criteria for the active database) in the *Direct Commands* documentation
- *Cleanup of the Active Database* in the *Administration* documentation
- [Cleanup in Batch Mode](#) in the section *Special Monitor Features and Batch Jobs*

Monitor (Server)

This section covers the following topics:

- [Entire Operations Monitor Functions](#)

- [Distribute Monitor Functions to Subtasks](#)

Entire Operations Monitor Functions

The Entire Operations Monitor activates and processes job networks according to their scheduled dates and times. This includes the following functions:

- Activation of scheduled job networks;
- Check of prerequisites to job submission (input conditions and resources);
- Job submission;
- End-of-Job checking and actions;
- Logging of all events.

In technical terms, there are two ways of running the Monitor: as one or several subtask(s) or as a batch task:

Monitor Subtask(s)

The Monitor can be run as one or several subtask(s) of an Entire System Server task in z/OS or z/VSE operating systems.

The JCL of the Entire System Server task (XCOM node) must be extended to meet the needs of the Monitor. The XCOM parameters must also be extended. The REGION assignment for the Entire System Server task must be large enough to contain the Monitor. For more details, see the section *Installing Entire Operations on Mainframes* in the *Installation and Setup* documentation.

The advantages of this method are:

- all Entire System Server calls of the Monitor against its host node are handled locally, without any inter-PROCESS communication, and
- Entire System Server and the Entire Operations Monitor share the same address space.

Running the Monitor as a Batch Task

The Monitor can be run as its own batch task in z/OS or BS2000.

The Monitor can run as any normal batch job. The functions it provides in this mode are the same as when it runs as an Entire System Server subtask. However, as a batch task, the Monitor requires that the operating system server node must be active all the time it is active itself.

From an implementation point of view, the Entire Operations Monitor is a special user within Entire Operations. The difference is that the Monitor is not driven by any terminal input but by its own processing rules.

The system administrator can define a time interval between Monitor cycles. At the beginning of a cycle, the Monitor “wakes up” and checks the Entire Operations work queues, performing any necessary actions such as submitting jobs and End-of-Job analysis. The time between cycles depends on the number of jobs defined to the system and the average job run time. The shorter the wait time, the shorter is the interval between job termination and its End-of-Job analysis. The price for this is increased overhead due to Monitor reactivation.

Distribute Monitor Functions to Subtasks

The individual functions that the Entire Operations Monitor performs can be distributed to several subtasks. Subtasking allows processes to run in parallel and increases performance. Monitor functions can be distributed to subtasks under z/OS, z/VSE, BS2000 and UNIX. Under BS2000 and UNIX, Monitor subtasks are separate processes in the operating system.

For information on how the usual Monitor functions can be distributed, see *Using the Monitor Task Profile* in the section *System Administrator Services* of the *Administration* documentation.

Monitor Start Network

If a job network with the name MON-START is defined under the owner SYSDBA, this network is executed exclusively at each Monitor startup. This is called the Monitor start network.

No other job network is started until the start network is terminated correctly.

The last job of the start network must not set any condition (but can reset conditions). During execution of the start network, the absolute condition SYSDBA/MON-START-RUNNING is set.

If any job of the start network ends not OK, this condition remains true and blocks any other Monitor action. The condition can be reset manually to free continuation of other processing. While the absolute condition is active, the message `Start Network still running` appears in the log and on the system console during each Monitor pass.

See also [Monitor Start Network](#) in the section *Special Monitor Features and Batch Jobs*.

Activation of Job Networks or Jobs

This section covers the following topics:

- [Activating a Job Network or Job](#)
- [Determination and Activation of necessary Symbol Tables](#)
- [Terminology](#)
- [Automatic \(scheduled\) Activation](#)

- [Manual Activation](#)
- [Job Activation Notes](#)

Activating a Job Network or Job

Activating a job network or job means preparing it for execution. On activation, the following is performed:

- The definitions of jobs, networks, logical conditions, symbol tables etc. are copied to the Entire Operations active database and assigned a unique **run number**;
- If necessary, symbol prompting is requested (see also [Symbol Prompting during Network or Job Activation](#)). However, symbol prompting is not performed for any subnetworks.
- The global activation exit user exit is invoked, if defined in the Entire Operations defaults;
- The JCL defined for jobs within the network is copied to the Active JCL storage on the active database;
- Variables (symbols) used in dynamically generated JCL are substituted by their current values. This does not apply to variables defined to be substituted at job submission time;
- The JCL definitions of active job networks, respectively of active jobs, can differ from the JCL definitions in the master definition. To allow this, the corresponding symbol tables must contain certain reserved symbols on activation. See also [Predefined Symbols](#) in the section [Symbol Table and Symbol Maintenance](#).
- If you use pre-generated JCL, symbol replacement is performed at the time of JCL generation.
- The Entire Operations Monitor recognizes the job network as active and checks time frames, input conditions and resources defined for the jobs. If all prerequisites for any jobs are fulfilled, these jobs are submitted.

Determination and Activation of necessary Symbol Tables

During a network activation or single job activation, the list of the required (active) **symbol tables** (see the section [Symbol Table and Symbol Maintenance](#)) will be determined by Entire Operations. The result of the determination will be written to the Entire Operations log.

It may look like this:

```
List of active Symbol Tables created
Determined Symbol Table Versions for 17.01.14
... Ob  Job      St  SymTab      defined      determined
... NV          00  N1649T00    (current)    -> v002
... JM  J001     00  N1649T00    (unnamed)    -> (unnamed)
... JM  J003     ED  N1649T00    (current)    -> v002
... JM  J004     ED  N1649T00    (nv)         -> (unnamed)
... JM  J005     ED  N1649T00    (svn)        -> v002
```

The column **St** contains the status of the symbol table to be activated.

ED means “evaluation duplicate”. It will be set if a previous determination (evaluation) resulted in the same symbol table with the same version. In this case, the symbol table (version) will be activated only once.

The determined symbol table versions (see the *Concepts and Facilities* documentation) will be used for the following symbol table activation.

In case of any determination error, the network activation of job activation will be aborted.

Terminology

In this documentation and on the user interface, the terms *activation* and *network start/job start* are used.

■ **Activation**

denotes the process of creating an active copy of a network or job definition.

■ **Network start/job start**

denotes the actual execution start time of the activated/active job network or job.

Automatic (scheduled) Activation

Job networks are activated automatically in two steps:

- At the beginning of a new day or during Monitor startup, all schedules are checked for job networks to be executed during that day. This process is called schedule extraction and the data extracted are called the activation trigger records.
- The activation trigger records force job network activation a short time before the earliest start of the network. This time span can be defined in the Entire Operations defaults: see the **Extraction of Schedules** option described in *Default Setting (2)* in the *Administration* documentation.



Notes:

1. If no earliest start time is defined on the network level, the network is activated immediately after schedule extraction.
2. The modification of a calendar or schedule always triggers a schedule extraction for the dependent job networks. For this reason, a job network could be activated even for the current day after such a modification.

Automatic Activation - Symbol Prompting

After the creation of an activation trigger record, active symbol tables are created for the specific network run. If there is at least one symbol marked as to be prompted within these active tables, a **symbol prompting** request is sent to the mailboxes of all users defined as message recipients for that network.

The network activation is kept in hold, until any user sees the request and enters or confirms the symbols to be prompted. For this reason, schedule extraction can be performed several days in advance. (See *Global Schedule Extraction* in the section *Special Functions* in the *Administration* documentation.)

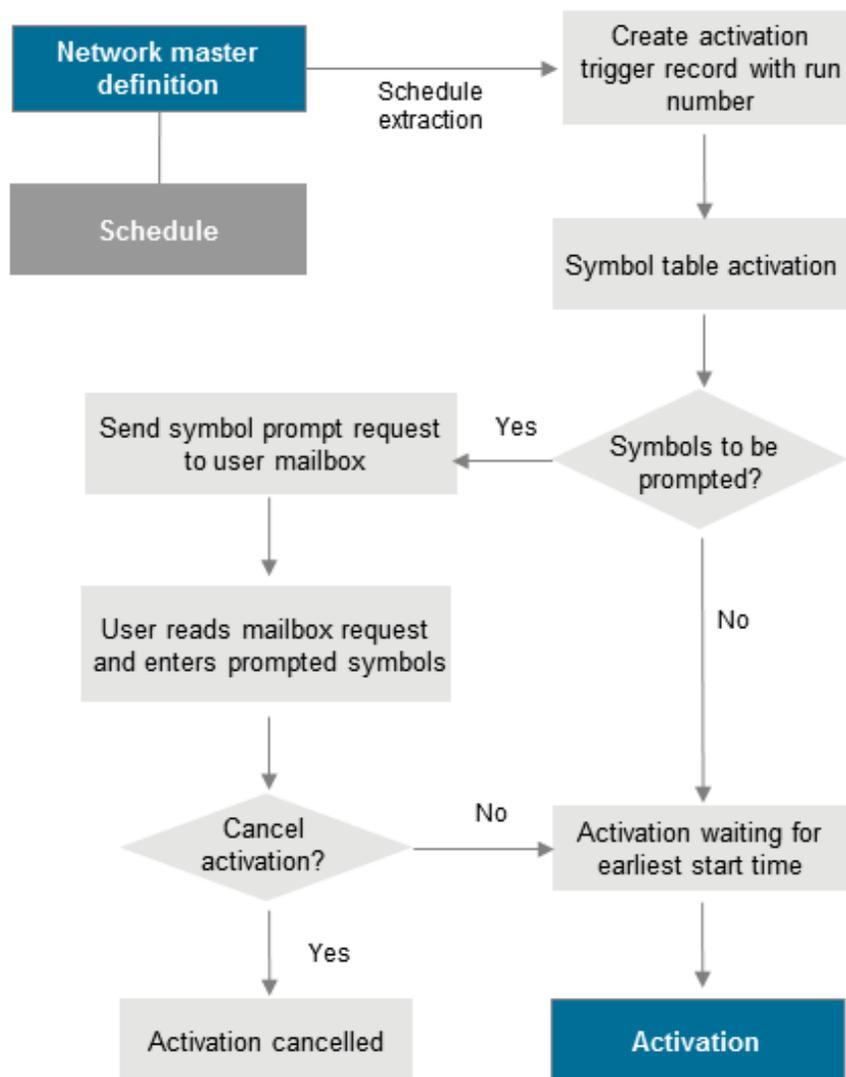
Manual Activation

It is also possible to activate a job network manually irrespective of any defined schedule. This may become necessary for a number of reasons, for example:

- No schedule has been defined for the job network;
- To override defined activation date and time;
- The job network is not scheduled for the required date.

Job network or job activation can also be triggered by any event within Entire Operations, for example by the termination of another job network or by the Entire Operations Application Programming Interface (API); see also [Accessing Entire Operations from other Applications](#). Like manual activation, this can be performed at any time.

Symbol prompting for active symbols is also performed, when a job or network is activated manually, if at least one symbol of a symbol table used by the job or network is appropriately marked.



Job Activation Notes

1. If the calculated latest start is after the calculated deadline, the last start will be set 1 minute before the deadline.
2. If the (new) latest start is before the earliest start, the job activation will be aborted with an error message.

Run Number

Entire Operations automatically assigns a run number to each active copy of a job network on the active database. This run number uniquely identifies the active copy of a job network and is automatically passed to its jobs, input conditions, etc.

The run number is assigned:

- During the creation of an activation trigger record;
- During a manual activation;
- If a network is activated by an API routine.

Run numbers are in the range 1 to 99999 by default and are unique on network level. When the maximum run number has been reached, assignment again starts from 1.

The upper limit for run numbers can be modified in the Entire Operations defaults as described in *Default Setting (2)* in the *Administration* documentation.

The assignment of a run number to each activation of a job network allows multiple activations of a job network on the same date, and allows you to distinguish between multiple active copies of the same job network.



Note: There is no guarantee that subsequent activations will have ascending run numbers. They are as unpredictable as operating system job numbers. Entire Operations retains the last run number, even for deleted job networks. If you define a new job network of the same name, the new run numbers start from the deleted network's last run number incremented by 1.

Schedules

A schedule is a predefined time table according to which a job network is activated. Entire Operations monitors schedules to determine which job networks are to be activated.

You can define activation dates in a schedule as explicit dates and/or periodic dates (days of the week, days of the month or a combination of days and months).

Entire Operations can optionally account for holidays in a schedule. For example, if you schedule a job network to run on every first day of a month and the schedule table is based on a calendar in which Saturdays and Sundays are defined as non-working days, then Entire Operations does not start the job network if the first day of the month is a Saturday or Sunday. Activation can be postponed until the following working day (Monday). In other words, Entire Operations can automatically interpret the first day of a month as the first working day of a month.

A schedule can be based on a predefined calendar which distinguishes between working days and non-working days (see the section [Calendars](#)).

You can inspect the defined schedule in calendar format, irrespective of whether activation dates are defined as explicit or relative dates: Entire Operations automatically translates relative dates into explicit dates.

You can make the execution of single jobs in a network dependent on their position in the schedule (for example, first schedule day of the week) or in the calendar (for example, last workday of the year).

For more information, refer to:

- [Schedule Maintenance and Scheduling](#)
- [Scheduling a Network](#)

Calendars

Calendars can form the basis for schedule tables defined for jobs and job networks. An Entire Operations calendar distinguishes between working days and non-working days as defined by the user (weekends, national holidays, personal vacations).

Calendars can be modified to change, include or delete non-working days. Modifications to calendars can affect the associated job network schedule(s).

Calendars are identified by owner, name and year, and can belong to an owner or be used system-wide. You can specify a system calendar or a calendar belonging to your owner for a schedule table, but you can only modify calendars belonging to your owner. System calendars can be modified by authorized users only.

Any number of calendars can be defined to Entire Operations. For further details, see the section [Calendar Maintenance](#).

Symbol Tables and Symbols

A symbol table contains a list of symbols (variables) that can be used for string substitution during JCL generation.

You can substitute symbols when activating a job network or job, that is, when the active JCL is loaded to the active database. You can also substitute symbols when submitting a job.

For detailed information on using symbol tables and symbols, see the section [Symbol Table and Symbol Maintenance](#).

Job Control (JCL)

This section covers the following topics:

- [Using Job Control in Entire Operations](#)
- [Editing JCL](#)

Related Topic:

- [Defining and Managing JCL for a Job](#) in the section *Job Maintenance*

Using Job Control in Entire Operations

Job control is used in Entire Operations as follows:

- **Master Job Control**
The JCL in its original format on the original storage medium. The usual JCL storage locations of the various operating systems are supported. The source texts for [dynamic JCL generation](#) are also considered master job control.
- **Active Job Control**
The actual JCL submitted to the operating system for execution. It is produced from the master JCL when the job or network is activated. The symbols are replaced with values from the active symbol table. If it is dynamic JCL, the generation is performed at this time. The active JCL is stored in the active Entire Operations database.
- **Pregenerated Active Job Control**
For reasons of performance, it might be necessary to generate active JCL in advance. See also [Pregenerating Active JCL](#) in the section *Job Maintenance*.

The JCL must be pregenerated again when:

- The definition of the master JCL storage has been modified;
- The master JCL has been edited;
- The corresponding symbol table has been modified.

Editing JCL

Job control can be edited with the Software AG Editor.

For detailed information on editing JCL, see [Editing JCL and Natural Sources](#) in the section *Job Maintenance*.

Dynamic JCL Generation (JCL Location MAC)

When Entire Operations activates a job network, the JCL of the jobs in the network is copied onto the active database. Entire Operations provides a facility which allows you to use variables in the original JCL and which can create parts of the JCL depending on program logic. Variables are substituted by their current values either at activation time or at job submission time (see [Symbol Replacement](#)). This is referred to as dynamic JCL generation and only applies to jobs with JCL location MAC (macro Natural source) in Entire Operations.

Dynamically generated JCL is useful if you wish the JCL to contain a step only under certain circumstances, for example, if the current date is YYYYMMDD, include job step X.

For information on defining dynamic JCL for jobs, see [Defining and Managing JCL for a Job](#) in the section *Job Maintenance*.

For information on editing macro JCL, see [Editing Master JCL and Natural Sources](#) and [Editing Macro Sources for Dynamic JCL Generation](#) in the section *Job Maintenance*.

Accessing Entire Operations from other Applications

The Entire Operations library contains some routines that can be invoked from any other Natural application to provide access to Entire Operations internal data. These routines are called the Application Programming Interfaces (APIs) and can be invoked simply with a Natural CALLNAT statement.

The Application Programming Interface provides the following features:

- Dynamic connection to the Entire Operations data file;
- Access to conditions;
- Access to symbols;
- Writing to the Entire Operations log.

The Application Programming Interface can be used for a number of purposes within and outside Entire Operations. Among them are:

- Dynamic modification of symbol tables during the execution of a job network;
- Modification of conditions from Natural programs;
- Exchanging information between Entire Operations and any other online or batch application;
- Setting input conditions for job networks from online applications;
- Inquiring the status of job networks from applications;
- Setting Entire Operations symbols from external tables;
- Inquiring Entire Operations symbols for use in external applications.

For more details, see the section [API Routines](#).

Job Execution as a Dummy Job

The execution of a dummy job means that the job is running without job control and without its own action within Entire Operations. Dummy jobs can have an expected run time, which they will be waiting in the system. Dummy jobs will always terminate with the state `o.k.`.

For detailed information on dummy jobs, see [Using a Dummy Job](#) in the section *Job Maintenance*.

Logging Facility

The Entire Operations Logging facility records every event and user action during job network processing. This information is available online. From the system log, you can select more detailed logs for individual jobs if logging at job level is specified at job definition time.

The default system log displays information about activities in the system as a whole such as user actions, date and time of events and messages about events. If more information is available for any item on the system log, it is preceded by an asterisk (*). Log information at the job level can be any of the following:

- **JCL log**
Displays the JCL of a specific job run;
- **SYSOUT log**
Displays the SYSOUT of a specific job run;
- **System message log**
Displays all operating system messages about jobs. The system log displays the first of these messages. You can select a job from the system log to display all system messages for that job.

A selection window in the Logging facility asks you to select the default log according to owner, network, job and run number.

For more information, see the section [Log Information](#).

Message Sending

Entire Operations can send messages to various locations. This is triggered by system-detected events and user-defined events.

Many [message destination types](#) can be defined. Among them is the sending of [e-mail on mainframes](#) and [e-mail on UNIX and Windows systems](#).

You can select the global events, which trigger message sending by Entire Operations.

Optionally, you can use a global exit for message sending. This exit detects all messages that must be sent for various reasons by the Entire Operations Monitor. The exit can store the message content in files and forward the message to other applications, etc.

See also [Message Sending](#) in the section *End-of-Job Checking and Actions*.

System Messages

Entire Operations displays status or error messages at the following locations:

Location	Description
On the current screen	If Entire Operations is used online. In many cases, additional information is written to the Entire Operations log. Subsequent to more complex errors it is recommended to have a look there. For more information, see Log Information .
Message column of the Active Jobs or All Active Jobs screen	Contains the last status message or error message for the active job. For more information, see the Message column described in Listing All Active Jobs for All Active Networks .
Entire Operations System Log screen	Contains all status messages and error messages. If database problems prevent you from writing to the log file, then the messages will be written to the SYSOUT of the Monitor tasks. For more information, see Log Information .
Monitor Task(s) SYSOUT	Contains mainly start and end messages of the Monitor tasks. In this case, some other important events are also logged in addition.

Location	Description
Console	<p>On mainframes, weighty messages of the Monitor tasks are written to the system console.</p> <p>In most cases, they are to be answered by the operator.</p> <p>An example of this is the non-availability of the database, while the Entire Operations Monitor is running.</p>

User Language

In Entire Operations, the languages English and German are available. The specified language controls the display in the following locations:

- All application screens and windows including status and error text messages;
- All online help texts including field-specific information;
- The Entire Operations log file.



Note: Entire Operations log messages are saved independent of the language. You can view them in English or German.

This section describes the locations where you can change the user language depending on your authorizations:

- [Direct Command](#)
- [System Default and User Profile](#)
- [Natural ULANG Parameter](#)

Direct Command

Enter either of the following Entire Operations direct commands and press ENTER:

```
SET LANGUAGE 1
```

(for English)

```
SET LANGUAGE 2
```

(for German)

See also the SET direct command in the *Direct Commands* documentation.

The language settings are kept for the duration of the Entire Operations session.

System Default and User Profile

As an administrator, you can specify the language in the following locations:

- On the **Default Setting (1)** screen of the **System Administrator Services** function.
- In a user profile on the **User Definition and Profile** screen.

See the relevant sections in the *Administration* documentation.

Natural ULANG Parameter

The Natural ULANG profile parameter controls the language used by the Entire Operations Monitor such as the SYSOUT of the Monitor tasks and the Console output.

You can specify ULANG dynamically at the start of a Natural session or, if authorized, statically in the Natural parameter module/file.

ULANG is described in the *Parameter Reference* of the Natural documentation.

Reporting

The **Reports** function helps you overview your job network environment to define objects, monitor the system and plan workloads.

All **report types** available are described in the section *Reporting*.

Cross References

The **Cross References** function is used to generate reports that contain cross references to individual objects.

All **types of cross references** available are described in the section *Cross References*.

Editor

Entire Operations provides a version of the Software AG Editor especially adapted to the Entire Operations environment.

You can use the editor to perform the following:

- Create or edit JCL for jobs. Existing JCL can be edited, even if it was written outside of Entire Operations using other editors;
- Create or edit JCL for jobs with JCL location MAC (macro);
- Write Natural programs to run as jobs in job networks or be executed as user exits;
- Write and browse text descriptions at the network, job, and event level (online documentation);
- Display JCL, job SYSOUT and listings in browse mode (no editing possible);
- Display the Entire Operations system log.

See also [Editing Master JCL and Natural Sources](#) in the section *Job Maintenance*.

Cleanup of the Active Database

This section covers the following topics:

- [Cleaning up the Active Database](#)
- [Deleting Work Files](#)

Cleaning up the Active Database

The operative data of Entire Operations must be removed again from the active database after a certain time. Part of this process is the removal of work files as well, which Entire Operations has created in the file system for job control purposes.

- The retention periods for active objects can be defined (see *Administration*) documentation.
- The cleanup may be defined to be carried out automatically every day. If no time is defined for the cleanup, then it will be started at 00:00. A time for the daily cleanup start can be defined. For a more detailed description, see *Administration* documentation.
- The cleanup of the active database can also be started manually any time (see *Administration* documentation).
- Furthermore, it is possible to run the cleanup of the active database in a Natural batch job (see [Cleanup in Batch Mode](#)) exterior to the Entire Operations Monitor. The cleanup in batch mode can be executed with the Monitor running or shut down.

Please note that the cleanup of the active database depending upon the data quantity to be processed affects the system. It is recommended to schedule the cleanup for silent times.

Cleanup runs can also be performed several times a day. This makes it possible to reduce the volumes to be processed per run.

Deleting Work Files

Entire Operations creates files in the operating system under BS2000, UNIX and Windows. Among other things, they contain the job SYSOUT or the JCL to be executed.

During the deactivation of active jobs, which have run in one of these operating systems, the assigned work files are deleted as well.

For BS2000, the names of these work files might also have been generated with a name exit, which is used for work file deletion, too.

All definitions are created in the Entire Operations Defaults. They are described in the *Administration* documentation.

II

Using Entire Operations

This section provides general information on the main features provided during an Entire Operations online session.

[Online Help](#)

[Starting and Ending an Entire Operations Session](#)

[Logging on and off an Operating System Server Node](#)

[Entire Operations Main Menu](#)

[Main Screen Elements and Functions](#)

[Date and Time Formats](#)

3 Online Help

- Screen Level 52
- Field Level 55
- Error Messages 57
- Online Technical Information 57

Entire Operations provides a comprehensive online help facility to help you select the appropriate function, command or item from any system screen.

Help texts are available on two levels:

Screen Level

The hierarchy of the help screens reflects the hierarchy of Entire Operations system screens. This means that you can invoke the online help facility from any system screen and you immediately see the help text defined for that screen. You can then return to the system screen or ask for more help.

The easiest way to invoke online help for any system screen is to simply press PF1 (Help). Alternatively, you can enter the direct command `HELP` in the **Command =>** line of any screen and press `ENTER`. From the **Main Menu**, there is also a third alternative: selecting the Help option.

- [Invoking Help from the Main Menu](#)
- [Invoking Help from a System Screen](#)

Invoking Help from the Main Menu

➤ To enter the online help facility at the top end of the hierarchy

- 1 Press PF1 (Help) in the **Main Menu**.

The main **Entire Operations Help** menu appears with a list of selectable items and a short explanation of their meaning:

```

05.12.14          Entire Operations Help          12:24:56
-----
----- Table of contents -----
Option ==>                                             H00000

          E n t i r e   O p e r a t i o n s   H e l p

Please select one of the following items to get detailed information:

    1  Help about Help   - The Entire Operations Help System
    2  The Maps          - Functionality of Entire Operations Maps
    3  Networks and Jobs - Maintenance of Networks and Jobs
    4  Schedules         - Definition and usage of Schedules
    5  Calendars         - Definition and usage of Calendars
    6  Active Queue     - Display and modification of Active Jobs
    7  Symbol Tables    - Symbol table editing
    8  Logging          - Selection and display of Log Information
    9  Dynamic JCL      - How to work with dynamic JCL
   10  System           - System Administrator Utilities
   11  Active Conditions - Maintenance of Active Conditions
   12  Editor           - The Entire Operations Editor
more ...

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
          Quit  End                               Forw

```

2 Press PF8 (Forw) to display the next help screen:

```

05.12.14          Entire Operations Help          12:51:24
-----
----- Table of contents (continued) -----
Option ==>                                             H0000020

          E N T I R E   O P E R A T I O N S   H E L P

Please select one of the following items to get detailed information:

    13  Reports          - Entire Operations Reporting Facility
    14  Cross References - Entire Operations Cross References
    15  Direct Commands - How to use the Command Line
    16  Import/Export   - Usage of the Import/Export Utility

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
          Quit  End                               Back

```

You can press PF7 (Back) to return to the previous help screen.

> **To select a Help item**

- 1 in the **Option ==>** field at the top of the menu, enter the number that corresponds to the required item, for example, 5 for **Calendars**.
- 2 Press ENTER.
- 3 The first help screen for the selected item (here: **Calendars**) appears:

```

05.12.14                Entire Operations Help                12:29:24
-----
----- Calendar Maintenance -----
Option ==>                H13000

Function
  This screen displays a list of defined calendars.
  The calendar owner may be preselected.
  Modification is allowed for own and linked owners only.

Line Commands ( * Enter letter for more help)
  C  Copy of a calendar.
  D  Deletion of a calendar.
* L  List (display) of a calendar (modification not possible).
* M  Modification of a calendar.
  S  Short description.
  W  Where used.
     Shows the job networks, for which the calendar is defined.

New calendars can be added with * PF2.

more ...

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
          Quit  End                                Forw
    
```

- 4 Help texts may consist of one or more screen pages. If a help text consists of more pages, you are notified by the prompt `more ...` at the bottom of the screen. Press PF8 (Forw) to display more help and press PF7 (Back) to return to the previous help screen.
- 5 Some help screens provide direct access to help texts on a related item or on another level in the help screen hierarchy. You can enter the number or letter for the required item in the **Option** field at the top of the help screen and press ENTER to display the associated help text. On help screens which list several items, an asterisk (*) marks the items on which more help is directly available.

Invoking Help from a System Screen

➤ To invoke a screen help

- 1 Press PF1 (Help) from any system screen.
- 2 A help text appears with general information about the screen from which you invoked the online help (example):

```

03.11.16                Entire Operations Help                18:01:05
----- Network Maintenance -----
Option ==>                H11000

Function
  A list of the defined or active networks.
  Various functions are available via line commands.
  New networks can be added with PF2 (appl. to master networks only)

The fields on the map

Cmd          For the line command.
Run          Number of currently active runs for this network.
Owner        The owner of the network.
Network      The network name.
Node         The default node for the jobs in the network.
Description  A short explanation.
              A bigger description is available with 'Description'.
Note: Character 'L' (Loop) is displayed between column Cmd and
       column Run if a definition loop was detected in the network.
more ...

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
          Quit  End                                Forw

```

- 3 Some Help screens provide direct access to help texts on a related item or on another level in the Help screen hierarchy. Proceed as described in [5] above.

Field Level

On the field level, you can invoke a window containing a help text for the field or a selection list:

- [Help Text](#)
- [Selection List](#)

- [PF Keys: Online Help](#)

Help Text

➤ To invoke a field help text from a system screen

- Enter a question mark (?) in the selected input field and press ENTER.

A window similar to the example below opens with an explanation of the field:

```

05.12.14          ***** Entire Operations *****          13:04:49
Job Maintenance   Owner NOPALL      Network EA0250S1   Version
-----
+-----+
|                                     | -J
|                                     | -J
|          Job Definition (Master)   | -J
| Job Name      ==> ST11-TEST_      Mod ==> SAG      05.11.14 14:04 | -J
| Description   ==> Where it all starts_____ | -J
| Job Type     ==> JOB              | -J
| Execution Node ==> N0082 MVS/ESA  | -J
|
| Special Type  ==> _____      Symbol Table ==> ?A0250S1__ | -J
| Restartable  ==> N                Symbol Table Version ==> _____ |
| Suffix Symbol ==> +-----+
|                                     |
|          Symbol Table               |
| End-of-Job Action Errors se       | *
| Enter-PF1---PF2--PF3--PF4---     | The Symbol Table contains the Variables
| Help Add End Edit                 | for the Dynamic JCL Generation.
|                                     | It can be maintained manually or be
|                                     | modified by programs.
|                                     |
| Enter * to select a Symbol Table. |
|                                     |
+-----+
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help Add End Save Up Down Menu
    
```

Selection List

If you do not know the appropriate values for an input field of an Entire Operations definition screen or window, you can usually open a selection window and select a value from a list: see the section [Selecting Items from a Menu](#) for further information.

PF Keys: Online Help

The following PF keys are available in the online help facility:

PF Key	Name	Function
PF2	Quit	Leave the online help facility.
PF3	End	Return to previous screen.
PF7	Back	Scroll up (backwards).
PF8	Forw	Scroll down (forward).

Error Messages

You can use the direct command `HELP` to display the long text of an Entire Operations system error message.

For the syntax that applies to the direct command `HELP` and all options provided with the command, see the *Direct Commands* documentation.

For further information on system messages and the locations where they appear, see the section [System Messages](#).

Online Technical Information

You can use the `TECH` direct command to displays technical information on your current Entire Operations session as shown in the following example:

```
+-----+
| Entire Operations      5.4.3.1      11.08.15
| System Automation Tools 3.4.4.1      11.08.15
| Entire System Server   3.5.4        Nd. 146
| Natural                8.2.5
| Editor                 8.2.5        25.07.11
|
| Program      MENUWOR
| Map
| Library      EOR543  Entire Operations 543
| Codepage     IBM01140
|
| User - Id    JMT      John M. Test
| Date / Time  04.09.15 11:44:16
| Terminal     1      48 VIDEO
| TP/OP System COMPLETE DAEF MVS/ESA SP7.2.1 JES2 z/OS 2.1
|
| PF3 End     PF4 NOP Fixes  PF5 SAT Fixes
+-----+
```

For further information, see *TECH* in the *Direct Commands* documentation.

4 Starting and Ending an Entire Operations Session

The startup procedure of Entire Operations is defined specifically for your installation: ask your system administrator.

Having started Entire Operations, the Entire Operations logon screen appears:

```
07/12/1618:14:50
=====
      EEEEE  NNN  NN  TTTTTTT  UU  RRRRRRRR  EEEEE
      EE      NNNN NN    TT      UU  RR   RR  EE
      EE      NN NN NN    TT      UU  RR   RR  EE
      EEEEE  NN  NNNN    TT      UU  RRRRRRRR  EEEEE
      EE      NN  NNN    TT      UU  RR  RR   EE
      EEEEE  NN  NN      TT      UU  RR   RR  EEEEE

      000000  PPPPPP  EEEEE  RRRRRR  AAAAAA  TTTTTTT  II  000000  NN  NN  SSSS
      00 00  PP  PP  EE      RR  RR  AA  AA    TT  II  00 00  NN  NN  SS
      00 00  PP  PP  EE      RR  RR  AA  AA    TT  II  00 00  NN  N  NN  SS
      00 00  PPPPPP  EEEEE  RRRRRR  AAAAAA  TT  II  00 00  NN  NNN  SSSSSS
      00 00  PP      EE      RR  RR  AA  AA    TT  II  00 00  NN  NN  SS
      000000  PP      EEEEE  RR  RR  AA  AA    TT  II  000000  NN  NN  SSSSSS
=====

User ID ==> SAG_____
Password ==>

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

 **Note:** If you are already logged on to Natural Security and your user ID is defined in Entire Operations, the above logon screen is bypassed and you are presented with the **Main Menu**. However, display of the logon screen can be forced by an appropriate setting in the Entire Operations defaults. If you wish to work in Entire Operations with a different user ID, you

must change the `STACK` statement in your startup procedure to reflect the required user ID (for example, `STACK=(LOGON SYSEOR;NATOP user-id)`). If in doubt, ask your system programmer.

The input fields on the logon screen have the following meaning:

Field	Description
User ID	<p>By default, your TP environment user ID appears in this field. You can enter Entire Operations under a different user ID by overtyping the default value. Any user ID entered in this field must be defined to Entire Operations. In addition, the user ID entered here must belong to the same owner as the user ID which appears here by default. For more information on the concept owner, see Owner and User Maintenance in the <i>Administration</i> documentation. If the user ID of your TP environment is not defined in Entire Operations, you cannot work with the product. The following message is displayed: EOR0009: INIT USER is unknown to Entire Operations.</p> <p>Note: The user ID entered here has a meaning for Entire Operations only. It is checked against internal Entire Operations <i>user definitions</i> and eventually against Natural Security. It has no meaning for any operating system object access. If you want to work with operating system objects (e.g. editing JCL), you must perform Entire System Server Logons to the nodes you want to work with. After such a logon you have access the access rights of the Operating System User ID you specified. For Entire Operations Networks and Jobs, you must define operating system user IDs specifically as JCL user IDs and submit (execution) user IDs.</p> <p>Refer also to Operating System User IDs, z/OS JCL User ID and z/OS Submit User ID.</p>
Password	Unused. Replaced by the passwords for single Entire System Server Logons .

If necessary:

1. Correct your user ID.
2. Press ENTER to display the Entire Operations [Main Menu](#).

➤ **To terminate an Entire Operations session**

- 1 In a **Command=>** input field, enter the direct command `FIN` or `QUIT` (see the *Direct Commands* documentation).
- 2 Press ENTER.

5 Logging on and off an Operating System Server Node

- Logon Function 62
- Fields: Node Logon 64
- Monitoring the Node Connection Status 65
- Logoff Function 66

This section provides instructions for logging on or off an Entire System Server node.

Logon Function

> To log on a node

- 1 Use the direct command LOGON or LOGIN as described in the *Direct Commands* documentation.

Depending on the operating system of the specified node, a **Node Logon** window similar to the examples below opens:

For a mainframe node:

```
+-----+
!                                     !
!           Entire Operations         !
!           Node Logon               !
!                                     !
! Node    ==> 148   (N0148)          !
!                                     !
!                                     !
!                                     !
! OpSys   ==> MVS/ESA                 !
! User ID ==> SAGTEST_____         !
! Password ==>                       !
!                                     !
!                                     !
!                                     !
! PF3 End                             !
+-----+
```

For a UNIX node:

```
+-----+
!                                     !
!           Entire Operations         !
!           Node Logon               !
!                                     !
! Node    ==> N0416 (416)            !
!                                     !
!                                     !
!                                     !
! OpSys   ==> Linux                   !
! User ID ==> SAGTEST_____         !
! Group   ==> _____             !
! Password ==>                       !
!                                     !
! PF3 End                             !
+-----+
```

For a Windows node:

```

Entire Operations
Node Logon

Node      ==> N0401 (401)
                                npr_pcsn01
OpSys    ==> Windows7
User ID  ==> SAGTEST_____
Domain   ==> SAG-HQ_____
Password ==>
PF3 End
  
```

In the input field(s), enter the required information. All fields are described in [Fields: Node Logon](#).

- 2 Press ENTER.

A **Node Status** screen similar to the example below appears:

```

07.12.16          ***** Entire Operations *****          17:40:34
                   Node Status at 17:40
-----
Node Server Name   Op.Sys.  Version   L LogonUID Status   Checked  Usage
N0123 NOP Test1    MVS/ESA  3.6.1     X SAG      active   16:32    1
N0456 NOP Prod1    MVS/ESA  3.6.1     X SAG      active   17:40    1

***** Bottom of Data *****

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      End      Check      Up      Down
  
```

You can press PF5 (Check) to refresh status display if last check was made some time ago.

The screen columns are explained in [Columns: Node Connection Status](#).

- 3 Press PF3 to return to the **Main Menu**.

Fields: Node Logon

The **Node Logon** window contains the following fields:

Field	Description
Node	Node number.
OpSys	Operating system on node.
User ID	<p>Valid user ID.</p> <p>If you logon to a mainframe node, input text is converted to upper case letters.</p> <p>If you logon to a UNIX or Windows node, input text is not converted to upper case letters. Therefore, make sure an exact spelling in mixed case.</p> <p>See also <i>Field Defaults</i>.</p>
Domain	<p>Input field for Windows or UNIX nodes:</p> <p>Enter a Windows domain or UNIX group, respectively.</p> <ul style="list-style-type: none"> ■ Windows: If this field is left blank, the logon will be carried out without domain. <p>Otherwise, this field must contain a valid domain. The default user ID and domain can be used (see the <i>Administration</i> documentation) for combinations of Entire Operations users and operating system servers (nodes).</p> <ul style="list-style-type: none"> ■ UNIX: If this field is left blank, the user's default UNIX group (from <code>/etc/passwd</code>) is used. <p>Otherwise, this field must contain one of the UNIX groups, which is visible in the output of the UNIX groups command.</p>
Group	<p>Input field for UNIX nodes:</p> <p>Enter a UNIX group.</p> <p>If this field is empty, the user's default UNIX group (from <code>/etc/passwd</code>) is used.</p> <p>Otherwise, this field must contain one of the UNIX groups, which is visible in the output of the UNIX groups command.</p> <p>See also <i>Field Defaults</i>.</p>
Password	<p>Input field for the password of the user specified in the User ID field.</p> <p>If you logon to a BS2000 node, input text is converted to upper case letters.</p> <p>If you logon to a z/OS node, input text is converted to upper case letters depending on the setting of the password mode for the node (see the <i>Administration</i> documentation).</p> <p>The text below the Password Field indicates the password mode set: upper case or case-sensitive.</p>

Field	Description
	You can set the password mode for a node in the Node Modification window described in <i>Adding and Modifying a Node Definition</i> in the <i>Administration</i> documentation.

Field Defaults

The default user ID - and perhaps also the group or domain - for logging on to a node can be defined for any combination of Entire Operations users and nodes (see *User Definitions and Profile Settings* in the *Administration* documentation). In this case, these values are displayed as defaults during the logon process.

They serve to simplify the logon process. If you want to logon with another user ID and/or group or domain, then these fields may simply be overwritten.

Monitoring the Node Connection Status

You can view a list with the current status of nodes defined in your environment.

➤ To monitor the node connection status

- In the **Command** line, enter the following direct command:

```
STATUS NODES
```

(See also the `STATUS` command described in the *Direct Commands* documentation.)

A **Node Status screen** appears.

You can press PF5 (Check) to refresh status display if last check was made some time ago.

The screen columns are explained in [Columns: Node Connection Status](#).

For further information, see also the API `NOPIST3N` described under [Inquire Network and Job Status, Symbol Table](#) in the section *API Routines*.

Columns: Node Connection Status

The columns contained on the **Node Status** screen are described in the following table.

Column	Description
Status	Node status of last Entire System Server call, for example, active or not active.
Node	Node number or short node name.
Server Name	Logical (long) node name of the server.
Op.Sys.	Operating system on node.
Vers	Entire System Server version.
L	If marked: logon was made to the node. If no user ID is shown in the LogonUID column, AUTOLOG=YES is active on the node.
LogonUID	User ID used to log on to the node.
Checked	Time of last status check.
Usage	Usage access count for the node for the current user only.

Logoff Function

> To log off a node

- Use the direct command LOGOFF as described in the *Direct Commands* documentation.

6 Entire Operations Main Menu

- Items in the Main Menu 68
- Special PF Keys: Main Menu 70

Once you have logged on to Entire Operations, the **Main Menu** appears. Your menu configuration may vary from the example below, depending on the authorizations in your user profile.

This example of the **Main Menu** displays all possible items:

```
16-12-01          ***** Entire Operations *****          16:15:13
Owner SAGTEST          Main Menu          User ID EXAMPLE
-----
      Main Menu          DC Solutions

1 Network and Job Maintenance          20 NATURAL ISPF
2 Active Job Networks
3 Schedule Maintenance
4 Calendar Maintenance
5 Log Information
6 Symbol Tables
7 System Administrator Services          Applications
8 Reports
9 Cross References          30 ISPF
10 Import/Export
11 Help

Command => _____

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help          End          Owner Mail
```

Related Topic:

- [Selecting Items from a Menu](#)

Items in the Main Menu

You can select the following items from the **Main Menu**:

- [Left Side - Main Menu](#)
- [Right Side - DC Solutions](#)

- [Right Side - Applications](#)

Left Side - Main Menu

1. [Network](#) and [Job](#) Maintenance

Define and maintain job network definitions on the master data-base, including job definitions, JCL, user exits, input conditions, End-of-Job handling, resources, and descriptions;

2. [Active Job Networks](#)

Maintain active database: this includes active networks and jobs, and all current definitions for a specific job run;

3. [Schedule Maintenance](#)

Maintain schedule definitions on the master database;

4. [Calendar Maintenance](#)

Maintain calendar definitions on the master database;

5. [Log Information](#)

Display different types of system information according to selection criteria;

6. [Symbol Tables](#)

Maintain symbol table definitions on the master database;

7. System Administrator Services

Maintain the Entire Operations environment, this includes user authorizations and maintaining the Entire Operations Monitor, available resources, nodes, system defaults, mailboxes, etc.;

8. [Reports](#)

Generate different types of reports, including workload forecasting on the basis of resources;

9. [Cross References](#)

Display cross references for user exits. Corresponds to the direct command `XREF`.

10. Import/Export

Log on to the Import/Export Utility. For further information, see the section *Import/Export Utility*.

11. [Help](#)

Display the online help facility.

An item is not displayed on your **Main Menu** (and is therefore not selectable) if your user profile does not contain the appropriate authorization.

Right Side - DC Solutions

If other Software AG products are installed, they can be called up directly from Entire Operations.

Right Side - Applications

If another application is defined in the menu, the name is shown here. For further information, see *User Application in Main Menu* in the *Administration* documentation.

Special PF Keys: Main Menu

You can perform the following functions from the **Main Menu** using these PF keys:

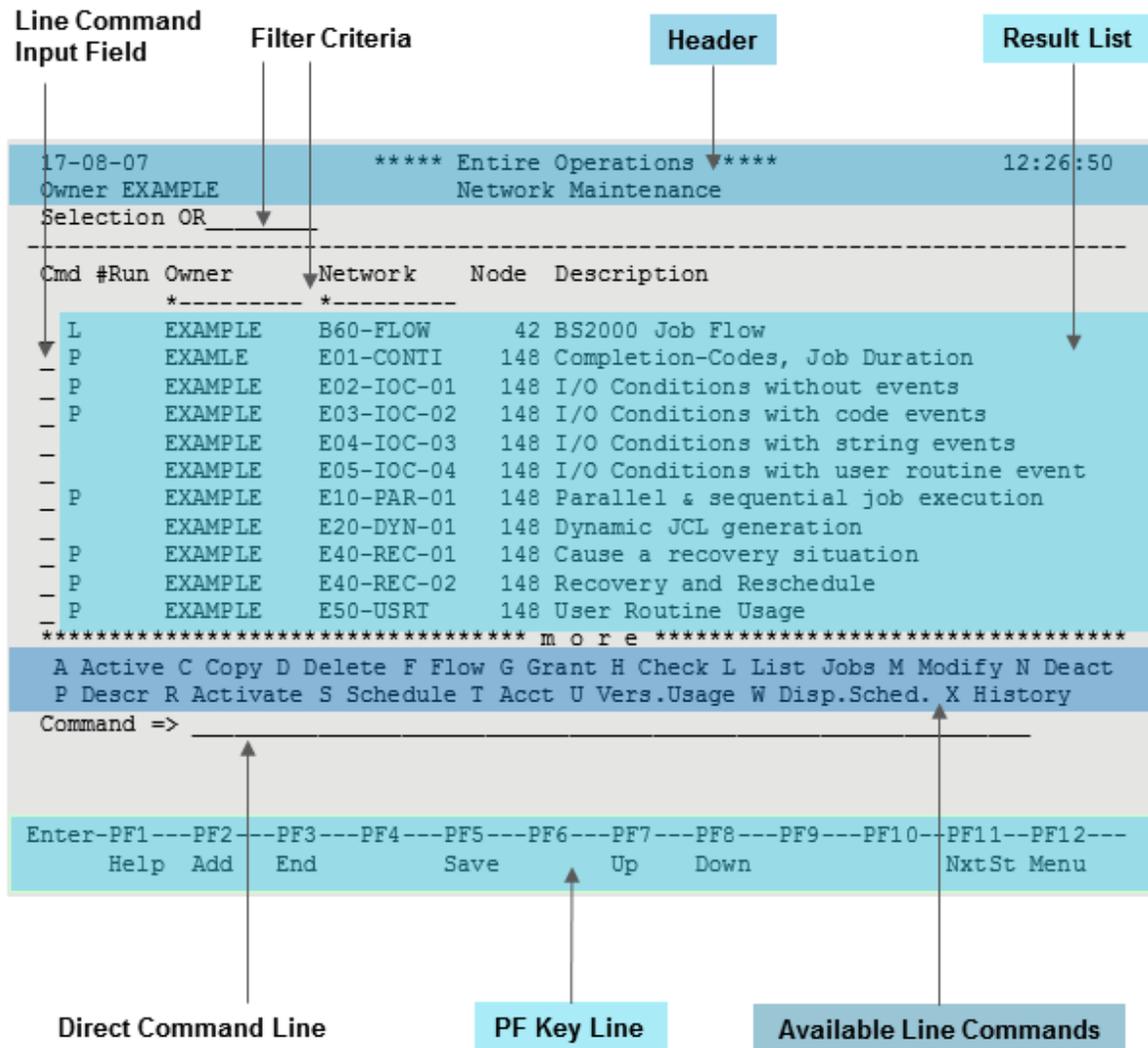
PF Key	Name	Function
PF11	Owner	Select a different owner from a selection list.
PF12	Mail	Display mailbox messages and requests. See also the section Working with Mailboxes .

7 Main Screen Elements and Functions

■ Using PF Keys	73
■ Using Direct Commands	74
■ Selecting Items from a Menu	74
■ Filtering Items in a Result List	75
■ Selecting Items from a Selection Window	76
■ Confirmation Window when Deleting Items	76
■ Priority of Command Processing	77
■ Error Messages	77

All Entire Operations screens have a similar layout. Some data appear on every screen, other data are screen-dependent.

The main elements that are typical for an Entire Operations list screen are indicated in the following example illustration of a **Network Maintenance** screen:



Legend

■ Header

The top line contains fields with the current date and time (see also *Date and Time Formats*), and a system identifier (Entire Operations). Subsequent lines in this section contain control fields with information such as current owner, network name and job name, as well as a screen name.

■ Filter Criteria

The values entered in the input fields above the result list indicate the selection criteria used for filtering the result list. See [Filtering Items in a Result List](#) and [Selecting a Range of Networks to be Listed](#).

■ Result List

The rows below the column headings contain the result list for an item selected from an Entire Operations menu or a subordinate selection screen.

If a result list is longer than the screen, a **** more *** line appears at the end of the list. You can press the appropriate PF keys (see [Using PF Keys](#)) to page up or down in the screen.

A **** Top of Data *** or **** Bottom of Data *** line tells you whether you are on the first or last page, respectively.

If the columns of a result list extend beyond the screen, you can press the appropriate PF keys (see [Using PF Keys](#)) to scroll data to the left or right, respectively.

■ Line Command Input Field

Input field (if applicable) for the one-character line commands listed below the result list with a short descriptive text.

The line commands available on a screen depend on the Entire Operations facility or function selected. They are described for each facility or function in the relevant documentation sections.

■ Direct Command Line

Input field for a direct command. See [Using Direct Commands](#).

■ PF Key Line

Lists commands that can be executed per PF key. See [Using PF Keys](#).

Using PF Keys

Some frequently used functions in Entire Operations are assigned to PF keys. A list of PF keys and associated functions is displayed in the last two lines of any system screen. Some PF keys may be assigned to different functions on different screens, but there are some functions which are always assigned to the same PF key:

PF Key	Name	Function
PF1	Help	Enter the Entire Operations online help system.
PF2	Add	Add a definition of the selected object type.
PF3	End	Return to previous screen.
PF5	Save	Save changes made to data.
PF7	Up	Scroll displayed data up (backwards).
PF8	Down	Scroll displayed data down (forward).

PF Key	Name	Function
PF10	Left	Scroll displayed data to the left.
PF11	Right	Scroll displayed data to the right.
PF12	Menu	Return to the Main Menu .

On some screens, object-specific functions may be assigned to PF keys. These are described in the appropriate section of this documentation. You can only use a PF key if it appears on the system screen together with the short descriptive text.

See also [Priority of Command Processing](#).

Using Direct Commands

You can use direct commands to directly call a function without having to move through a hierarchy of menus.

You can enter a direct command on any screen or window that contains a **Command =>** input field (also referred to as the command line).

All available direct commands are described in detail in the *Direct Commands* documentation.

See also [Priority of Command Processing](#).

Selecting Items from a Menu

> To select a menu item

- 1 In the **Command=>** input field of the **Main Menu**, enter the one or two digit number displayed next to the item you want to select. If you want to select the item of a subordinate menu (if available), add the required item number and separate the two numbers with a period (.).

For example:

7 selects the item **System Administrator Services**, a subordinate menu;

7.5 selects the **Monitor Defaults** item (5) of the subordinate **System Administrator Services** menu.

Or:

On the **Main Menu** or a subordinate menu, position the cursor on the line that contains the item you want to select.

- 2 Press ENTER.

Filtering Items in a Result List

You can reduce the number of items shown in the result list of an **Entire Operations screen** by specifying filter criteria in the input fields below the column headings on the screen.

Usually, you have the following input options to filter items, where *value* is any combination of one or more characters:

Input	Names Selected
*	All names.
<i>value</i> *	All names that start with <i>value</i> . Example: AB* Selected: AB, AB1, ABC, ABEZ Not selected: AA1, ACB
<i>value</i> >	All names greater than or equal to <i>value</i> . Example: AB> Selected: AB, AB1, BBB, ZZZZZZZ Not selected: AA1, AAB
<i>value</i> <	All names less than or equal to <i>value</i> . Example: AX< Selected: AB, AWW, AX Not selected: AXA, AY
<i>value</i> ?	All 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 names that match and <i>value</i> combined with a question mark (?) in any order. Example: A?C Selected: ABC, ACC Not selected: ABCA, AC

Selecting Items from a Selection Window

When you are required to specify an existing item in an input field of an Entire Operations definition screen or window, you can select an item from a list of available items provided in a selection window.

➤ To select an item from a selection window

- 1 Enter an asterisk (*) as a wildcard in the input field. To display a more specialized list, enter a prefix followed by an asterisk. For example, NET* displays a list of all items beginning with NET.
- 2 Press ENTER.

A window opens with a list of selectable items.

If the required item is not listed in this window, you can scroll toward the bottom of the list by pressing ENTER or PF8 (Down). Press PF7 (Up) to scroll back toward the top of the list.

- 3 Select an item by marking it with any character.
- 4 Press ENTER.

The window closes and the name of the selected item is written to the appropriate input field.

Confirmation Window when Deleting Items

➤ To delete an item from a list of items

- 1 On an Entire Operations list screen, enter the line command D next to the item you want to delete.

A confirmation window opens.

- 2 Confirm the deletion by entering the name of the item or the letter Y (Yes).
- 3 Press ENTER.

The item is deleted.

Priority of Command Processing

Commands and data are processing according to the following priority:

1. Command line
2. PF keys
3. Data entered in input fields

Error Messages

All error messages are displayed in the first screen or window line. You can correct the error on the system screen and continue, or leave the screen with PF3 (End).

For more details, see the *Messages and Codes* documentation.

8 Date and Time Formats

- Date Display and Input Options 80
- Time Display and Input Options 81

This section describes the date and time formats Entire Operation uses for processing system function and producing output reports. These formats are independent of the format options Windows provides in an Entire Operations GUI environment.

The current date and time appears in the top line of nearly every system screen.

The date displayed on the screen is represented in the Entire Operations default format specified in the **Date Format** field of the *Default Setting (1)* (see the *Administration* documentation).

If you enter a date or specify a data as a parameter, the syntax must conform to the format set in the Entire Operations defaults. Otherwise, you receive an appropriate error message indicating that the date or time does not match the required format.

Date Display and Input Options

Possible date representations and valid input values for date fields are described in the following section.

Input fields for dates have different lengths and you may have to (or can optionally) shorten the input string. Entire Operations always treats shortened input (or output fields) with the century.

A shortened input string is always extended to a complete date (even when not representable on the screen), according to the default format set for your environment. This is indicated in the following examples that assume the current date is December 30, 2016.

Date Pattern	Date Format Setting	Examples
DD.MM.YY	G German format.	30.12.16
		301216
		30.12
		3012
DD/MM/YY	E European format.	30/12/16
		301216
MM/DD/YY	A American format.	12/30/16
		123016
YY-MM-DD	I International format (default).	16-12-30
		161230
		1612
YYYYMMDD	8 Eight-digit format.	20161230
		2016

Explanations:

- DD is a two-digit day, MM a two-digit month, YY a two-digit year and YYYY a four-digit year.

A two-digit year is internally always processed with four digits.

- Input of a year value is optional. If you omit the year, the current year is used.

If Entire Operations interprets an input string as a past date (relative to the current date), an appropriate message occurs.

- The century that relates to a 2-digit year value is determined by using a sliding window that assumes the date to be within a range of +50/-50 years, relative to the current year. For example:

If the current year is 2016, the sliding window will cover the years 1966 to 2065. A 2-digit year value *nn* from 66 to 99 is interpreted accordingly as 19*nn*, while a 2-digit year value *nn* from 00 to 65 is interpreted as 20*nn*.

- Input of the separator characters slash (/), period (.) and dash (-) is optional. If you omit the separators, the input string is extended to a complete date.

Time Display and Input Options

The time is represented in one of the following formats:

Syntax	Example	Complete Time
HH:MM:SS	145231	14:52:31
	1452	14:52:00
	14:53	14:53:00
	14	14:00:00

Explanations:

- HH is a two-digit hour, MM a two-digit minute and SS a two-digit second.

The hour range is 0 to 23. The minute and second range is 0 to 59.

- Input of minutes and seconds is optional. If you omit the minutes and/or seconds, the input string is extended to a complete time.
- Input of the separator character colon (:) is optional. If you omit the separators, the input string is extended to a complete time.

III

Entire Operations Utilities

9 Entire Operations Utilities

▪ Introduction	86
▪ Extract Log Data to Log Selection File	87
▪ Delete Old Data in Log Selection File	88
▪ Output of Log Data to a File	89
▪ Print Account Information from Entire Operations Log	89
▪ Mass Change of Network / Symbol Table Version	90
▪ Monitor or Task Wait Time Modification	91
▪ Monitor Shutdown	91
▪ Monitor Start	92
▪ BS2000 Jobs	92
▪ List or Delete TO-ACTIVATE Command Records	93
▪ Check the Existence of Symbol Table Definitions	94
▪ Mass Change of the Owner / Delete Owner with all Objects	94
▪ Mass Change of Access Rights Granted to Networks	97
▪ User ID Mass Update in Network and Job Definitions	99
▪ Mass Change of Node Numbers	100
▪ Mass Logon Processing in Batch Mode	100
▪ Bulk Execution of MACRO Commands	102
▪ Migration of Log Data from NOP Versions lower than 411 to NOP 412 and above	104
▪ Data Migration to the current Entire Operations Version	106

Entire Operations utilities are used to perform selected Entire Operations functions.

A utility is usually a standalone Natural program that runs either online or in batch mode when invoked by a Natural batch job.

Introduction

- [Required LFILE Assignments](#)
- [Log Selection File - Format 1](#)

Required LFILE Assignments

LFILE	Explanation
216	Entire Operations System File 1.
131	System Automation Tools Log File.
215	Entire Operations System File 3 (Log Selection File).

Log Selection File - Format 1

An empty log selection file (LFILE=215) for accounting data is part of the Entire Operations installation files.

The DDM for the above is EOR-LOG-SELECTION-1.

The DDM is delivered with descriptors on almost every field. It is up to the user to release descriptors which are not necessary.

The file can be read and evaluated with Natural.

DB 255, File 215

This file applies to the default sequence of the EOR-LOG-SELECTION-1 file.

T	L	DB	Name	Format	S	D	Remarks
*	EOR Log Selection						
	1	AA	LGS-RECORD-TYPE	A6	N	D	
*							
G	1	AB	LGS-BS2-DATA				
	2	AD	LGS-BS2-USERID	A8	N	D	
	2	AE	LGS-BS2-ACCOUNT	A8	N	D	
*							

T	L	DB	Name	Format	S	D	Remarks
G	1	AG	LGS-TIME-DATA				
	2	AH	LGS-CPU-TIME	P7.2	N	D	/* in sec.
	2	AI	LGS-JOB-START	T12	N	D	/* date + time
	2	AK	LGS-JOB-STOP	T12	N	D	/* date + time
	2	AM	LGS-JOB-ELAPSED	P13.0	N	D	/* in 1/10 sec.
*							
G	1	AN	LGS-DATA-3				
	2	AO	LGS-LOG-TIME	T12	N	D	/* date + time
	2	AP	LGS-OWNER	A10	N	D	
	2	AQ	LGS-NETWORK	A10	N	D	
	2	AR	LGS-RUN	P13.0	N	D	
	2	AS	LGS-JOB	A10	N	D	
	2	AT	LGS-STATE	A3	N	D	
	2	AU	LGS-JOB-TYPE	A3	N	D	
*							
	1	AY	LGS-EXECUTION-NODE	N3.0	N	D	
	1	AW	LGS-JOB-ID	A5	N	D	
	1	AX	LGS-OS-JOB-NAME	A8	N	D	
*							
* Superdescriptors							
*							
	1	AV	LGS-KEY-1	A30	N	S	
*			LGS-OWNER	1-10			
*			LGS-NETWORK	1-10			
*			LGS-JOB	1-10			

Extract Log Data to Log Selection File

- [Natural Program Call](#)

- [Parameters](#)

Natural Program Call

LGAX1S-P P-DATE-FROM P-DATE-TO

Parameters

Name	Format	Description
P-DATE-FROM (optional)	A8	Start date for the extraction in format YYYYMMDD.
P-DATE-TO (optional)	A8	End date for the extraction in format YYYYMMDD.



Notes:

1. Start date and end date for the extraction are optional. If wanted, both parameters must be set. If the start date is smaller than the time stamp in the private control record (see note 2 and 3), the extraction is not performed.
2. Log data extraction always starts where the last extraction ended.
3. This time stamp is kept in a private control record. (When the program is started for the first time, the start date 01.11.2001 is used).
4. The extraction data consists of accounting data (see [Network and Job Accounting Data](#) in the section [Reporting](#)).
5. New extraction data do not overwrite existing extraction data.

Delete Old Data in Log Selection File

- [Natural Program Call](#)

- [Parameters](#)

Natural Program Call

LGAX1D-P P-RETENTION-DAYS

Parameters

Name	Format	Description
P-RETENTION-DAYS	I2	Entries older than this parameter are deleted.

Output of Log Data to a File

The output of log data for printing or further processing is described in the subsection [Output of Log Information](#) in the section [Log Information](#).

Print Account Information from Entire Operations Log

- [Natural Program Call](#)
- [Parameters](#)

Natural Program Call

LG-AP1-P P-TIME-FROM P-TIME-TO

Parameters

Name	Format	Description
P-TIME-FROM	A14	Format: YYYYMMDDHHIISS
P-TIME-TO	A14	Format: YYYYMMDDHHIISS

Mass Change of Network / Symbol Table Version

- Natural Program Call
- Parameters

Natural Program Call

```
OW-MB--P
P-OBJECT-TYPE P-OBJECT-OWNER P-OBJECT-NAME P-OLD-OBJECT-VERSION P-NEW-OBJECT-VERSION ←
P-ALLOW-UPD
```

Depending on the settings of the Natural parameter ID, parameters may need to be comma separated instead of using blanks.

Parameters

Name	Format	Description
P-OBJECT-TYPE	A2	Object Type: NV = Network Version. SV = Symbol Table Version.
P-OBJECT-OWNER	A10	Owner of Object.
P-OBJECT-NAME	A10	Object name.
P-OLD-OBJECT-VERSION	A10	Version of Object to be renamed
P-NEW-OBJECT-VERSION	A10	New Version of Object.
P-ALLOW-UPDATE	A1	N = Only perform a check. Y = Allow updates.



Notes:

1. The Natural session parameter IM must be set to IM=F.
2. The monitor should be stopped during the execution of this utility to avoid unintended results of network activations.
3. The program can be restarted in case of an abnormal termination to complete the rename action.
4. Every object (to be) changed is reported. The report is written to Natural main output.

Monitor or Task Wait Time Modification

- [Natural Program Call](#)
- [Parameters](#)

Natural Program Call

MO-WTM-P P-MONITOR-WAIT-TIME P-TASK-ID

Parameters

Name	Format	Description
P-MONITOR-WAIT-TIME	I4	In seconds.
P-TASK-ID	I2	Number of a monitor task. Only the wait time of this task will be modified.



Notes:

1. If a wait time less than 5 seconds is specified, the wait time is set to 5 seconds.
2. The modifications carried out using this program only apply until the end of the current monitor run.

Monitor Shutdown

- [Natural Program Call](#)
- [Parameters](#)

Natural Program Call

MO-SHD-P

Parameters

None.

Monitor Start

- [Natural Program Call](#)
- [Parameter](#)

Natural Program Call

MO-ST--P P-MONITOR-NODE

Parameter

Name	Format	Description
P-MONITOR-NODE	I2	Value = 1 to 99999

BS2000 Jobs

These jobs are used to update the submit user ID, submit password and account number.

- [Natural Program Call](#)
- [Parameters](#)

Natural Program Call

BS2U01-P P-OWNER P-USERID-OLD P-PW-OLD P-USERID-NEW P-PW-NEW P-ACCOUNT-NEW

Parameters

Name	Format	Description
P-OWNER	A10	Entire Operations owner of jobs to be modified.
P-USERID-OLD	A8	BS2000 user ID to be exchanged.
P-PW-OLD	A8	BS2000 submit password to be exchanged.
P-USERID-NEW	A8	BS2000 new user ID.
P-PW-NEW	A8	BS2000 new submit password.
P-ACCOUNT-NEW	A8	BS2000 new account number.

**Notes:**

1. An asterisk (*) for a parameter is converted to **blank**.
2. Job fields are only exchanged if the Entire Operations owner, the old BS2000 user ID, and the old submit password do match.
3. Fields are replaced, only if the new value is **not blank**.
4. A log is written to the screen. In batch mode, the log is printed to Natural printer 1. This printer must therefore be assigned in the JCL.

List or Delete TO-ACTIVATE Command Records

- [Natural Program Call](#)
- [Parameters](#)

Natural Program Call

TA-DEL-P P-FUNCTION

Parameters

Name	Format	Description	
P-FUNCTION	A1	D	Delete TO-ACTIVATE records.
		L	List TO-ACTIVATE records.

**Notes:**

1. With this batch utility you can list or delete the internal TO-ACTIVATE command records in the Entire Operations active database.
2. Use this utility only in co-operation with the Software AG support and if you have to perform a manual removal of the TO-ACTIVATE command records.
3. This utility may be used only if the Entire Operations Monitor is not running.

Check the Existence of Symbol Table Definitions

- [Natural Program Call](#)
- [Parameters](#)

Natural Program Call

CHNWST-P P-FUNCTION

Parameters

Name	Format	Description
P-FUNCTION	A1	A List all networks without a symbol table.
		S List subnetworks without a symbol table.

The report is written to print file 1 in batch and to screen in online.

Mass Change of the Owner / Delete Owner with all Objects

- [Natural Program Call](#)
- [Parameters](#)
- [Notes](#)

Natural Program Call

OW-MB--P
P-OWNER-OLD P-OWNER-NEW P-ALLOW-UPDATE

Depending on the settings of the Natural parameter ID, parameters may need to be comma separated instead of using blanks.

Parameters

Name	Format	Description	
P-OWNER-OLD	A10	Old owner	
P-OWNER-NEW	A10	New owner. If you specify ==DELETE== here, the owner will be deleted, with all depending objects.	
P-ALLOW-UPDATE	A1	N	Perform a check only.
		Y	Allow updates.

Notes

- The Natural session parameter `IM` must be set to `IM=F`.
- The Entire Operations Monitor must be stopped during the execution of this utility.
- The Entire Operations Monitor must be stopped during the execution of this utility.
- The old and new owner names may not be SYSDBA.
- The utility runs in two passes. During the first pass, the data to be changed are checked against potential problems. An error message (line with `E:`) will result in update mode (`P-ALLOW-UPD`) switched to `N`. A typical problem is "target exists", when owner rename would cause merging of Entire Operations objects within a network. A warning message (line with `W:`) indicates situation of interest which does not prevent from object renaming.
- The second pass is either the real update or diagnostic run depending on update mode value.
- Every object (to be) changed is reported. The report is written to Natural main output.
- The utility must run with Entire Operations system file 1 opened in Adabas exclusive (`EXU`) mode, which means: the Entire Operations Monitor and every other Natural session with Entire Operations system file 1 must be terminated before. This prerequisite is necessary to protect the integrity of Entire Operations objects and links between them. For this reason we suggest Natural profile parameter:

```
OPRB=(EXU=fnop1,UPD=fnat,flog,ACC=fsec)
```

where:

- `fnop1` = Entire Operations system file 1
- `fnat` = Natural system file FNAT
- `flog` = Entire Operations or System Automation Tools log file
- `fsec` = Natural Security system file

Other files required to run Natural should be added to `UPD=` or `ACC=` list.

- We suggest a four step job to run `OW-MB--P`:

- Step 0: Adabas ADAORD/ADASAV/ADAULD to save Entire Operations system file 1 when following OW-MB--P runs in update mode;
- Step 1: Adabas ADADBS utility with parameters:

```
ADADBS OPERCOM
STOPF=fnop1
ADADBS OPERCOM LOCKX=fnop1
```

- Step 2: Natural session with OW-MB--P;
- Adabas ADADBS:

```
ADADBS OPERCOM
UNLOCKX=fnop1
```

- If the utility runs in a “normal” Natural session, it will (most likely) run out of space in Adabas WORK area or max ISN hold limit or will be waiting for records being held by others (also with update mode N). Even if it runs to the end, all the changes on Entire Operations system file 1 will be backed out.
- There is another Natural profile parameter, DBUPD=OFF, which may be used to run this utility safely and in parallel to other sessions however the utility report may indicate false error messages due to other activities on Entire Operations system file 1.
- When OW-MB--P reports the E: ... targets exist (xx: *nnn*) error, *xx* is the two-character target code and *nnn* is the number of target records found. *xx* can be the following:

NM (Network Master)
 JM (Job Master)
 EM (Event Master)
 JA (Job Active)
 EA (Event Active)
 JC (JCI active)
 CO (Conditions active)
 SA (Symbols Active)
 RP (Resource Prerequisites)
 RU (Resource usage)

- Existing targets are reported in the sequence above, e.g. EM means that targets NM and JM were not found.

If the reported target is:

- JM: create manually missing network definition and jobs will appear in it. Deactivate/delete it again;
- CO: these are visible in Active Conditions.
- The rest of the targets can be cleaned up with a new special “orphan” cleanup batch utility:

```
CL-ORP-P owner update-mode
```

where all parameters are mandatory:

- *owner* (A10): P-OWNER-NEW from the erroneous OW-MB--P run (SYSDBA not allowed);
- *update-mode* (L): update mode N or Y.
- Everything about execution of OW-MB--P mentioned above applies the same way to CL-ORP-P, i.e. it must be executed with Natural profile parameter `OPRB` to open Entire Operations system file 1 in exclusive update mode or profile parameter `DBUPD=OFF` may be used for diagnostic purposes. The report about (possibly) deleted “orphan” objects is written to Natural main output.
- Be sure to keep all the reports of OW-MB--P and CL-ORP-P runs with update mode Y.

Mass Change of Access Rights Granted to Networks

- [Natural Program Call](#)
- [Parameters](#)

Natural Program Call

```
LOGON SYSEOR
NV-MB--P
P-FUNCTION,P-OWNER,P-NETWORK,P-VERSION,P-TYPE,P-NAME,P-OPTION
FIN
```

The Natural program NV-MB--P provides the option to add, modify or delete user access rights granted to networks. NV-MB--P can run online and in batch mode.

The above Natural program call assumes that the Natural profile parameter `ID` (input delimiter character) is set to a comma (,).

Example:

```
LOGON SYSEOR
NV-MB--P
U,EXAMPLE,*,*,0,NOPALL,DORA
```

In the example above, the access rights of owner `NOPALL` change to `DORA` for all networks and network versions of the owner `EXAMPLE`.

Parameters

Name	Format	Description	
P-FUNCTION	A1	D Delete all user access rights granted to all networks specified with P-OWNER, P-NETWORK and P-VERSION.	
		U Changes user access rights granted to all networks specified with P-OWNER, P-NETWORK and P-VERSION: <ul style="list-style-type: none"> ■ Access rights are modified to P-OPTION for the user specified with P-TYPE and P-NAME. ■ Access rights are added if the user specified with P-TYPE and P-NAME does not yet exist. 	
P-OWNER	A10	Name of the network owner or a range of names specified with a wildcard (* or ?). You can only use one parameter to specify a name range.	
P-NETWORK	A10	Name of the network or a range of names specified with a wildcard (* or ?). You can only use one parameter to specify a name range.	
P-VERSION	A10	Name of the network version or a range of names specified with a wildcard (* or ?). You can only use one parameter to specify a name range.	
P-TYPE	A1	Only required for P-FUNCTION=U User type specification:	
		0	Owner
		U	User
P-NAME	A8	Only required for P-FUNCTION=U Name of the owner or user whose access rights you want to change.	
P-OPTION	A5	Only required for P-FUNCTION=U Options for access restrictions (multiple values are possible):	
		R	Read access
		W	Read and write access
		D	Read, write and delete
		0	Owner rights with permission to grant access to other users
		A	Network activation rights

User ID Mass Update in Network and Job Definitions

- Natural Program Call
- Parameters

Natural Program Call

US-UPD-P P-FUNCTION P-OLD-VALUE P-NEW-VALUE P-NODE P-OWNER P-NETWORK

Parameters

Name	Format	Description	
P-FUNCTION	A3	NSU	Change network default submit user ID.
		NSG	Change network default submit group.
		NJU	Change network default JCL user ID.
		NJG	Change network default JCL group.
		JMU	Change job modification user ID.
		JSU	Change job submit user ID.
		JSG	Change job submit group.
		JJU	Change job JCL user ID.
		JJG	Change job JCL group.
		JSB	Change SYSOUT node; <i>catid</i> ; user ID on BS2000 node.
P-OLD-VALUE	A20	If not empty, change only if old user ID is equal to this. This field is case-sensitive.	
P-NEW-VALUE	A20	May contain user ID or group (see P-FUNCTION) or blank to reset corresponding user ID. The field is case-sensitive.	
P-NODE-A5	A5	Must be a defined node in Entire Operations. If you want to pass a node number directly, you have to assign it to a field in the format A5 first. Mnemonic (short) node names in the A5 format can also be passed. SYSOUT definitions can only be used by jobs that have this node defined as a submit node.	
P-OWNER	A10	These two fields can be used for wildcard range definition. Wildcard is allowed in one field only	
P-NETWORK	A10		



Important: This utility may be used only if the Entire Operations Monitor is not running.

Mass Change of Node Numbers

Natural Program Call

```
ND-MB--P LOCATION NODE-OLD NODE-NEW OWNER NETWORK NETWORK-VERSION
```

Parameters

Name	Format	Description
LOCATION	A3	All occurrences of node will be changed in master definitions of network, job or event. For other values, refer to the List of JCL Locations.
NODE-OLD	N5	Node number.
NODE-NEW	N5	Node number.
OWNER	A10	Owner filter (optional).
NETWORK	A10	Network filter (optional).
NETWORK-VERSION	A10	Network version filter (optional).

Notes

1. Both `NODE-OLD` and `NODE-NEW` must be defined and valid and of the same operating system class, i.e., you can change OS/390 to a z/OS node but not VSE to z/OS.
2. If no value is specified for `OWNER`, `NETWORK`, `NETWORK-VERSION`, an asterisk (*) is used by default.
3. You can use an asterisk (*), a greater than sign (>) or a smaller than (<) sign as a wildcard character.

If a wildcard appears in `OWNER`, `NETWORK` contains an asterisk (*).

4. Program `ND-MB--P` can be run safely (only to list objects) in a Natural session with the Natural profile parameter `DBUPD=OFF` set.

Mass Logon Processing in Batch Mode

- [Natural Program Call](#)
- [Fields in the Input Lines](#)

- Single Node Logons without using a Work File

Natural Program Call

```
ND-LB--P
```

The utility ND-LB--P must be run in a Natural batch environment. The logging is written both to SYSOUT and to the Entire Operations (SAT) Log. Passwords will not be logged.

The *input* is read from *Natural work file 1*. For each logon to be performed, one line must be defined with the format:

```
node userid group password
```

Example:

```
517 sag users sag-pwd
```

Comment lines must have an asterisk (*) as first character. The output is written to the Natural batch job SYSOUT and to the Entire Operations log.

Fields in the Input Lines

Name	Format	Description
<i>node</i>	N5	The Entire System Server node, to which the logon is to be performed.
<i>userid</i>	A20	User ID for the logon.
<i>group</i>	A20	Group for the logon. For Windows nodes, <i>group</i> must contain the domain. If no group or domain is used, enter a hyphen (-) instead. Example: 148 USER - USER-PWD
<i>password</i>	A20	The password for the logon. The <i>password</i> parameter must be enclosed in double quotes (" <i>password</i> "), if the password contains at least one blank. Passwords with trailing blanks are not supported.



Notes:

1. This utility can be used only if the Entire Operations Monitor is not running.
2. There may be one or several blanks between the parameters.
3. If no group or domain is used, enter a hyphen (-) instead.
4. For UNIX and Windows nodes, the input is case-sensitive.

5. For mainframe nodes, the text strings will be converted to upper case before the logon is performed.



Caution: Users of this utility must be aware that the input file is very security-sensitive, because it contains passwords in clear text.

Single Node Logons without using a Work File

To allow single node logons without using a workfile, the parameters can be passed on the Natural command line too.

The command line parameters are:

1. node number
2. user ID
3. group (UNIX) resp. domain (Windows)
(pass a hyphen (-) if the default group resp. no domain is to be used)
4. password

The parameters are case-sensitive.

On return, ND-LB--P provides a return code in the AIV variable +ND-LB--P-RC (I4).

It contains 0 (zero), if the logon was successful, and any other error code in case of an error.

Bulk Execution of MACRO Commands

- [Program: MAC-GE-P](#)

Program: MAC-GE-P

The program performs the MACRO command in batch mode for a given selection of job network(s).

Natural Program Call

(with the Natural parameter ID=, being set)

```
LOGON SYSEOR
MAC-GE-P
MODE,OWNER,NETWORK,NETWORK-VERSION
FIN
```

Example:

```
LOGON SYSEOR
MAC-GE-P
A,SN,A-1,v-unnamed
FIN
```

Parameters

Name	Format	Description	
MODE	A1	A	Catalog all jobs with JCL location MAC.
		M	Catalog jobs with JCL location MAC, which were not yet cataloged.
OWNER	A10	Owner filter.	
NETWORK	A10	Network filter.	
NETWORK-VERSION	A10	Network version filter	

The parameters P-OWNER, P-NETWORK and P-NETWORK-VERSION may contain selection wildcards:

```
*, >, <, ?
```

**Notes:**

1. The Program MAC-GE-P must be started in batch.
2. The Natural profile parameters LC=ON, EDPSIZE=100 must be set.
3. On mainframe systems. The Entire Operations monitor must be active. The task 90 must be running.

Result Checking

The SYSOUT of the batch job contains detailed information about the cataloging. The results must be checked.

Example:

```

Owner      Network  Version  Job          Library  Member  Result
SN         A-1      v-unnamed MAC1-541     EOR-T541 MAC1     Done
SN         A-1      v-unnamed MAC1N511    EOR-N541 MAC1     Error
>>> NAT0806 - Library not found.

```

Migration of Log Data from NOP Versions lower than 411 to NOP 412 and above

- [Program: LGMIG41W](#)
- [Data Migration \(System File Adaptation during Entire Operations version upgrade\)](#)

Program: LGMIG41W

The program reads the old Entire Operations Log created with ADAULD and converts it to the System Automation Tools log format suitable for loading with ADALOD. The whole process ADAULD + LGMIG41W + ADALOD runs about 3 times faster than conventional LG-MIG41.

Entire Operations log must be unloaded with ADAULD options:

```
MODE=SHORT , SORTSEQ=CT
```

LGMIG41W may be called online or batch. LFILE 216 (Entire Operations 411 System File 1) must be defined. The following Natural work files must be defined:

File	Type	Description
1	Input file	ADAULD of Entire Operations Log file with option: MODE=SHORT , SORTSEQ=CT
2	Output file	System Automation Tools Log file for subsequent ADALOD.
3	Input file	System Automation Tools Log FDT (System Automation Tools installation file SAT31x.SYSF).

By default LGMIG41W writes variable length records to workfile 02 with record size 32760. This size may be changed with optional parameter. For example, LGMIG41W 20000 tells LGMIG41W to limit record size to 20000 bytes.

Be sure to make corresponding settings in JCL (block and/or record sizes).

LGMIG41W works on mainframe platforms only.

LGMIG41W reports data and superdescriptor storage sizes which may be used to calculate DB extents for the System Automation Tools Log file.

Although Entire Operations may function with a different Entire Operations log file, FDT, for LGMIG41W, it must have the fields in the following order:

CP, CQ, CR, I1, Y6, CT, EP, CV, AV, AP, HP, HQ, HR, A3, FA, DB, AW, IP.

Below is a sample BS2000 JCL (ISP mode) for Entire Operations log migration:



Note: The file and DB extents are allocated exactly for 412.766 Entire Operations log record migration. This may differ mostly depending on whether extended logging is being used. The SORT and TEMP data sets were used to about 92% by ADALOD.

```

./MIG41W LOGON
/SYSFILE SYSOUT=MIG41W.OUT
/SYSFILE SYSLST=MIG41W.LST
/SYSFILE SYSIPT=(SYSCMD)
/SYSFILE SYSDTA=(SYSCMD)
/REMARK ===== UNLOAD 321 EOR-LOG =====
/FILE MIG41W.ULD.321, LINK=DDOUT1, SPACE=(19971,300)
/DO DB36473(DBDEF)
/EXEC (ADARUN,&(#ADALIB))
ADARUN PROG=ADAULD,DB=36473,DE=2000,Q=32760
/EOF
ADAULD UNLOAD FILE=13,SORTSEQ=CT,MODE=SHORT
/FILE MIG41W.ULD.321, SPACE=-999999
/STA
/REMARK ===== CONVERT EOR-LOG TO SAT-LOG FILE =====
/SETSW ON=(4,5)
/EXE $EDT
LFILE=(216,36473,61,,),STACK=(LOGON SYSEOR AAA AAA)
$WR '#PARM'
$H
/SETSW OFF=(4,5)
/SETSW ON=2
/FILE #PARM, LINK=CMPRMIN
/FILE MIG41W.ULD.321, LINK=W01, BLKSIZE=
/FILE MIG41W.ULD.411, LINK=W02, BLKSIZE=(STD,16), SPACE=(58224,300)
/FILE $SAG.SAT313.SYSF, LINK=W03, BLKSIZE=
/EXEC N31B
LGMIG41W
FIN
/FILE MIG41W.ULD.411, SPACE=-999999
/STA
/REMARK ===== LOAD SAT-LOG-FILE =====
/FILE MIG41W.ULD.411, LINK=DDEBAND, BLKSIZE=
/DO DB36473(DBDEF)
/EXEC (ADARUN,&(#ADALIB))

```

```
ADARUN  PROG=ADALOD,DB=36473,DE=2000
/EOF
ADALOD  LOAD  FILE=10,NAME='SAT311-LOG-NOP'
ADALOD      MAXISN=412766,UISIZE=642B,NISIZE=19404B
ADALOD      DSSIZE=18265B,DSDEV=2000
ADALOD      USERISN=NO,ISNREUSE=YES
ADALOD      TEMPDEV=2000,TEMPSIZE=15000B
ADALOD      SORTDEV=2000,SORTSIZE=15000B
/LOGOFF  NOSPOOL
```

Here is the text of procedure DB36473(DBDEF) referenced in JCL above:

```
/PROC  C
/FILE  ADA36473.ASS01,LINK=DDASSOR1,OPEN=OUTIN,BLKSIZE=,SHARUPD=YES
/FILE  ADA36473.DATA1,LINK=DDDATAR1,OPEN=OUTIN,BLKSIZE=,SHARUPD=YES
/FILE  ADA36473.WORK1,LINK=DDWORKR1,OPEN=OUTIN,BLKSIZE=,SHARUPD=YES
/FILE  ADAXXX.SORT1,  LINK=DDSORTR1,OPEN=OUTIN,BLKSIZE=(STD,2),SHARUPD=YES
/FILE  ADAXXX.TEMP1,  LINK=DDTEMPR1,OPEN=OUTIN,BLKSIZE=(STD,2),SHARUPD=YES
/ERAJV  #ADALIB
/STEP
/CATJV  #ADALIB
/SETJV  #ADALIB,'$SAG.ADA713.MOD'
/FILE  &(#ADALIB),LINK=DDLIB
/ENDP
```

Data Migration (System File Adaptation during Entire Operations version upgrade)

In certain cases it may be necessary to perform a data migration intentionally.

A data migration usually is necessary when upgrading Entire Operations to a new version.

Data Migration to the current Entire Operations Version

The data migration to the current Entire Operations version adapts various objects and fields, so that they can be used by the current Entire Operations version,

```
DMDRIV - P
```

It is recommended to use a Natural batch job to perform the migration. The Natural commands are:

```
LOGON SYSEOR
DMDRIV -P
FIN
```

**Notes:**

1. A data migration consists of several independent migration steps.
2. Each migration step will be performed only once. Internal control records prevent multiple executions.
3. The migration invocation may be repeated as often as necessary.
4. The migration is logged into the Natural SYSOUT.
5. The data migration may be performed only if the Entire Operations Monitor is not active.

IV

Using Owners

10

Using Owners

- Listing Owners 112
- Owner at Logon 113
- Linking Additional Owners 114
- Owners Granted Access to Individual Networks 114

Each user ID is associated with single or multiple owners in order to group network maintenance.

You can only list, access and maintain networks that belong to associated owners (unless you have administrator rights):

- The main (mandatory) owner to which your user ID is **assigned at logon**,
- The owners **linked to your User ID** and
- The owners which **granted you individual network access**.

This section provides an overview of the functions available to view and assign owners.

Related Topics:

- The concept of user/owner assignments is described in the sections *Owner* and *Owners and User IDs* in the *Concepts and Facilities* documentation.

Listing Owners

You can list all owners associated with your user ID.

> To list owners

- In the Entire Operations **Main Menu**, press PF11 (Owner).

Or:

Use the SET OWNER direct command as described in the the *Direct Commands* documentation.

An **Owner List** window similar to the example below opens:

```

16-11-08          ***** Entire Operations *****          16:24:34
Owner EXAMPLE          Main Menu +-----+ TQA5
-----+-----+-----+-----+-----+-----+-----+-----+
Main Menu          !      User NATQA5          !
                  !      Owner List          !
1  Network and Job Maintenance      20 !          !
2  Active Job Networks              21 !          !
3  Schedule Maintenance            !          !
4  Calendar Maintenance            !          !
5  Log Information                  !          !
6  Symbol Tables                    !          !
7  System Administrator Services    !          !
8  Reports                          !          !
9  Cross References                 30 !          !
10 Import/Export                    !          !
11 Help                              !          !
                  !          ** more **          !
                  !      --PF3---PF7---PF8---          !
Command => _____ !      End   Up   Down          !
                  +-----+-----+-----+-----+
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help           End                               Owner Mail

```

In the example above, all owners associated with the user ID NATQA5 are listed.

You can press PF7 (Up) or PF8 (Down) to scroll up or down in the list.

Owner at Logon

Your user ID is assigned an **Owner at Logon** (main owner) in your user profile as described in *Viewing, Adding and Modifying a User (Administration documentation)*.

When you log on to Entire Operations, your main owner (in the previous example, the owner EXAMPLE) is set for your current session by default. The display is filtered to show only those networks belonging to the main owner and its associated owners.

➤ To change the current session owner

- From the **Owner List window** in the Entire Operations **Main Menu**, select the required owner with any character and press ENTER.

Or:

Use the SET OWNER direct command as described in the *Direct Commands* documentation.

You can only change the session to an owner which is associated with your user ID (unless you have administrator rights).

Linking Additional Owners

(Administrator rights required)

In addition to the **Owner at Logon** specified in your user profile, you can link other owners to your user ID as required. See *Adding and Removing User/Owner Links* in the *Administration* documentation.

Owners Granted Access to Individual Networks

You can grant a user or an owner unrestricted or restricted access to a selected network by using the **Granting Definition** option of the network maintenance function described in [Granting Definition: Authorizing Other Users or Owners to Access a Network](#) in the section *Network Maintenance*.

Listing Granted Owners

You can view all users/owners who granted you network access by using the appropriate line commands of the **Network Maintenance** function: see [Selecting a Range of Networks to be Listed](#).

V Network Maintenance

This section provides information on how to define and maintain job networks.

For information on all definitions made at the job and event level, see the section [Job Maintenance](#).

[Maintaining Job Network Definitions](#)

[Copying Job Network Definitions; Cloning Network Versions](#)

[Displaying the Job Flow within a Network](#)

[Checking for a Loop in a Job Network](#)

[Maintaining the Usage of Network Versions](#)

[Applying Network Defaults to Jobs \(Mass Update\)](#)

[Display Network Accounting Data](#)

[Listing Jobs for a Network](#)

[Activating a Job Network Manually](#)

11

Maintaining Job Network Definitions

▪ Listing all Network Definitions	118
▪ Selecting a Range of Networks to be Listed	119
▪ Commands: Network Maintenance Screen	120
▪ Displaying and Modifying a Network Definition	121
▪ Adding a Network Definition	122
▪ Fields and Commands: Network Definition	123
▪ Operating System and Environment Defaults	126
▪ Specifying Recipients for Network Messages	133
▪ Granting Definition: Authorizing Other Users or Owners to Access a Network	134
▪ Documenting Your Networks	137
▪ Deleting a Job Network	138

Listing all Network Definitions

> To list all networks defined for an owner

- Select the **Network and Job Maintenance** option on the Main Menu.

Or:

Use the direct command `LIST NETWORKS` as described in the *Direct Commands* documentation.

A **Network Maintenance** screen similar to the example below appears:

```

17.12.13          ***** Entire Operations *****          17:16:28
Owner EXAMPLE          Network Maintenance
Selection OR_____
-----
Cmd #Run Owner      Network      Node  Description
*-----*
_      EXAMPLE      ADMIN       N0146 Administrator workshop
_      EXAMPLE      ADMIN2      N0146 Administrator workshop
_      EXAMPLE      B60-FLOW    N0194 Job Flow, BS2000
_      EXAMPLE      B60-FLOWFX  N0194 Job Flow, BS2000
_      EXAMPLE      B60-FLOW38  N0038 Job Flow, BS2000
_      EXAMPLE      B60-FX      N0031 Job Flow, BS2000
_      EXAMPLE      B60FLOW194  N0194 Job Flow, BS2000
_      EXAMPLE      B60FLOW211  N0031 Job Flow, BS2000
_      EXAMPLE      DEMO-NET    N0146 BRY-DEMO-NET
_      EXAMPLE      E01-CO-FX   N0146 Completion-Codes, Job Duration
_      EXAMPLE      E01-CO-148  N0148 Completion-Codes, Job Duration
***** m o r e *****
A Active C Copy D Delete F Flow G Grant H Check L List Jobs M Modify N Deact
P Descr R Activate S Schedule T Acct U Vers.Usage W Disp.Sched. X History
Command => _____

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Add   End           Save           Up             Down           NxtAc Menu
    
```

When one or more job networks are defined for the owner, they appear in the main information section of this screen.

Columns: Network Maintenance

The **Network Maintenance** screen contains the following columns:

Column	Description
Cmd	One-character line command input field. Possible values are listed in the command section of the screen (see also the section <i>Line Commands: Network Maintenance</i>). Note: The letter L (loop) appears between the Cmd and #Run columns if a loop was found during a loop test for the corresponding network. For further information, see the field description for Loop exists .
#Run	Number of currently active runs for the corresponding network.
Owner	Owner name of the network; the first line contains the prefix specification used to obtain the list of networks: enter an asterisk (*) and press ENTER to list networks belonging to the current owner (see screen header) and networks of all other owners for which you are authorized, or enter ABC* and press ENTER to list networks belonging to all owners whose names begin with ABC.
Network	User-defined network name. The first line contains the prefix specification used to obtain the list of networks: enter an asterisk (*) to list all networks belonging to the specified owners, or enter ABC* to list networks whose names begin with ABC. You can also use a question mark (?) as a wildcard at any position in the selection string, for example, A?C.
Node	Default execution node for the jobs in the network.
Description	Short description of the network.

Selecting a Range of Networks to be Listed

You can specify the range of networks to be listed by entering one or more of the following values in the **Selection** field on the **Network Maintenance screen** and pressing ENTER:

Value	Description
A	Only networks with at least one active run are listed.
G	Granted networks are listed for the current user and the owner selected on the Network Maintenance screen (excluding current owner).
0	Networks of the current owner are listed. Owner selected on Network Maintenance screen has no effect.
R	Number of active runs is shown in the #Run column. Using this option could increase database access.
U	Networks granted for the user.

You must specify at least either 0 or G. You can specify both 0 and G together. The default value is OR.

If you enter a question mark (?) in the first position of the **Selection** field and press ENTER, a **Network Selection** window opens where you can enter the required value(s).

Commands: Network Maintenance Screen

This section describes the commands available on the [Network Maintenance screen](#):

- [Line Commands: Network Maintenance](#)
- [Special PF Keys: Network Maintenance](#)

Line Commands: Network Maintenance

You can perform the following functions on any network listed on the [Network Maintenance](#) screen:

Cmd	Description
A	List all active jobs for the network. Use this command to maintain active jobs.
C	Copy all definitions of the selected network for a new network, or clone a network version.
D	Delete the selected network , including all jobs and all other definitions for this network. If several network versions exist, a network version selection window opens. Mark the version you want to delete.
F	Display overview of job flow within network. If several network versions exist, a network version selection window opens. Mark the version you want to delete.
G	Grant other users or owners access to this network.
H	Check for loops in the network. If several network versions exist, a network version selection window opens. Mark the version you want to delete.
L	List all jobs defined for this network. Use this command for job maintenance. If several network versions exist, a network version selection window opens. Mark the version you want to delete.
M	Modify selected network definition. If several network versions exist, a network version selection window opens. Mark the version you want to delete.
N	Deactivate active jobs , selectable by dates.

Cmd	Description
P	Description. Invoke the Entire Operations editor to write an online text description for the selected network. If several network versions exist, a network version selection window opens. Mark the version you want to delete.
R	Activate the selected network manually.
S	Network schedule definition.
T	Display network accounting data. See also Network and Job Accounting Data.
U	Maintain the usage of network versions.
W	Viewing a Network Schedule Definition as a Calendar.
X	Display the execution history (previous network runs)

Special PF Keys: Network Maintenance

You can perform the following functions from the **Network Maintenance** screen using these PF keys:

PF Key	Name	Function
PF2	Add	Add a network definition.
PF11	NxtSt	Display a system-wide list of all planned job and network activations.

Displaying and Modifying a Network Definition

➤ To view or modify a network definition

- 1 Type **M** in the line command field of the selected network on the **Network Maintenance screen** and press **ENTER**.

A **Network Modification** window opens similar to the example below opens:

```

30.11.15          ***** Entire Operations *****          11:18:46
Owner EXAMPLE          Network Maintenance
+-----+
- |                               Network Modification                               | -
  | Owner EXAMPLE   Network EXA-NET1__ Version _____ |
  | Description Example Network_____ |
  |                                                    Loop existing N
  | Default Values for the Jobs
  | Execution Node N0146 MVS/ESA          Symbol Table _____
  | JCL Node       N0146 MVS/ESA          Symbol Table Version _____
  | JCL Location   _____           Symbol Table Activation Mode X
  |                                                    Escape Act @ Sub $
  | End-of-Job Action Errors set 'not ok' Conditions ==> N
  |
  | File _____
  | VolSer _____           Password
  |                                                    defined no
  | --PF1-----PF3-----PF5---PF6---PF7---PF8---PF9---PF10---PF11----- | **
  | Help      End      Save Spec Symb SP-UR DfJb Copy  MsgRe | t
+-----+

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Add  End      Save      Up    Down          NxtSt Menu

```

The window shows the current values defined for the network.

The fields and commands available are explained in [Fields and Commands: Network Definition](#).

- 2 Press PF5 (Save) to save any changes.
- 3 Press PF3 (End) to close the window.

Adding a Network Definition

➤ To add a new network definition

- 1 Press PF2 (Add) on the [Network Maintenance screen](#).
- 2 A **Network Addition** window opens.

Define the network by entering the required values in the input fields.

The fields and commands available correspond to the fields and commands available in the [Network Modification window](#). They are described in [Fields and Commands: Network Definition](#).

- 3 Press PF5 (Save) to save your new network definition.

Fields and Commands: Network Definition

The fields in the [Network Addition/Modification window](#) are described in the following table:

Field	Description
Owner	Owner of the network.
Network	Network name. Together with owner name, uniquely identifies the network in the master database.
Version	<p>Network version.</p> <p>You may add a new network with a non-blank version. It is possible to invoke the network addition multiply for the same network, but with different versions.</p> <p>New network versions can be created by cloning too as described in Copying Job Network Definitions; Cloning Network Versions.</p> <p>For further information on network versions, see <i>Object Versioning</i> in the <i>Concepts and Facilities</i> documentation.</p>
Description	<p>Short description of the network. This text appears in the list of networks on the Network Maintenance screen.</p> <p>A longer description of the network can be added using the editor (see Documenting Your Networks).</p>
Loop existing	<p>Y Loop exists: a loop has been found in the network while linking network jobs via their input and output conditions or while defining input and output conditions of a job.</p> <p>N Loop does not exist: no loop has been found in the network, or no loop test has yet been performed, for example, with the line command H.</p>
Default Values for the Jobs	The input fields under this heading are used to specify default values for subsequent new job definitions in the network. Each default value specified here can be overridden at the job level.
Execution Node	Default node ID on which jobs within this network are submitted. This value can be modified here or overridden at the job level. To list the possible nodes, type an asterisk (*) here and press ENTER. The operating system type appears after a valid node number. To display the field help, type a question mark (?) here and press ENTER.
JCL Node	Node on which JCL can be accessed. The default is the same as for the Execution Node . The operating system type appears after a valid node number.
JCL Location	<p>Type of storage for the JCL: see List of JCL Locations.</p> <p>The default is used in new job definitions and can be overridden there.</p>

Field	Description
Symbol Table	<p>Default symbol table for those jobs in the network that use the dynamic JCL generation facility. Can be overridden at the job level and is therefore optional here.</p> <p>Enter an asterisk (*) and press ENTER to display a selection list of available symbol tables. The selected symbol table can be edited using PF7 (Symb).</p>
Symbol Table Version (symbol table)	<p>You can maintain several versions of a symbol table.</p> <p>Enter an asterisk (*) and press ENTER to display a selection list of available symbol tables.</p> <p>Reserved names:</p> <p><i>blank</i> Unknown (implicit) version.</p> <p>(<i>current</i>) Is replaced by the current version for the activation date or determination date.</p> <p>(<i>nv</i>) Is replaced by the version of the network used.</p> <p>For further information, see <i>Object Versioning</i> in the <i>Concepts and Facilities</i> documentation.</p>
Symbol Table Activation Mode	<p>A During the network activation. No symbol prompting is possible.</p> <p>X After schedule extraction. Symbol prompting can be used for scheduled networks. This is the default.</p>
Esc Act	<p>Activation Escape Character = Network default value</p> <p>This escape character is the prefix for Natural code lines and symbols to be replaced at activation time.</p> <p>Caution: Existing Dynamic JCL might become invalid after changing this escape character and applying defaults to jobs.</p>
Esc Sub	<p>Submission Escape Character = Network default value</p> <p>This escape character is the prefix for Natural code lines and symbols to be replaced at submission time.</p> <p>Caution: Existing Dynamic JCL might become invalid after changing this escape character and applying defaults to jobs.</p>
End-of-Job Action Errors set 'not ok' Conditions	<p>If Y is specified here, output condition(s) are set for the job after all end-of-job actions are performed. This includes end-of-job action (EJA) exits. Any errors in the end-of-job action processing will cause the setting of the conditions for job not ok. If the job was already set to not ok, it will remain not ok, regardless of the definition here.</p> <p>Note:</p> <ol style="list-style-type: none"> Setting this flags may cause a longer elapsed time of a network, because the condition setting waits for the termination of the other end-of-job actions. The network level setting will be overridden by job settings.

Field	Description
	If N is specified, errors during end-of-job action processing will have no impact on the job result. Job level: If the field is empty, the network level definition will be inherited at activation time. This is the default.
File	Name of the file or Natural library according to the value of the JCL Location field. For BS2000: The default pubset will be stripped from the file name prior to storing it. This allows easier migration to another default pubset.
VolSer	Volume serial number of data set (if data set is not cataloged).
Password defined	Password if the file or library specified in the Data set field is password protected: no No password is defined. yes Password is defined.

Special PF Keys: Network Addition/Modification

You can perform the following functions using these PF keys in the [Network Addition/Modification window](#):

PF Key	Name	Function
PF6	Spec	Special definitions for different operating systems and environments.
PF7	Symb	Open the Usable Symbol Tables window, and select a symbol table for browsing or modification. See also Listing Usable Symbol Tables in the section <i>Symbol Table and Symbol Maintenance</i> .
PF8	SP-UR	Define a symbol prompt user exit. See also Specifying User Exits for Symbol Modification in the section <i>Symbol Table and Symbol Maintenance</i> .
PF9	Dfjb	Apply network defaults to jobs.
PF10	Copy	Copy a network definition.
PF11	MsgRe	Specify message recipients for the network.

1. Press PF5 (Save) to save the network definition.
2. Press PF3 (Exit) to return to the **Network Maintenance** screen.
3. The new network is now listed on the **Network Maintenance** screen.

Operating System and Environment Defaults

➤ To specify operating system specific defaults

- 1 Press PF6 (Spec) in the **Network Modification window**.

The following window opens:

```

29.10.08          ***** Entire Operations *****          11:24:09
Owner SN          Network Maintenance
Selection OR_____

- +-----+-----+-----+-----+-----+-----+-----+-----+
!                                                         !
!                   Network Modification                   !
! Network      ==> A-1_____ Owner ==> SN                !
! Description  ==> Time Range Test ..._____            !
!                                                         !
!                                                         Loop existing ==> N !
! Default Values f +-----+-----+-----+-----+     !
! Execution Node !                                     ! -01_____ !
! JCL Node      ! Please select the Operating System    ! b ==> _ !
! JCL Location  !                                     ! e ==> X !
!               !           _ BS2000                   !           !
! File ==> NO !           _ MVS                         ! _____ !
! VolSer ==> __ !           _ VSE                       !           !
!               !           _ UNIX, Windows            ! no           !
! --PF1-----PF3--- !                                     ! --PF12-- ! **
! Help End ! or Environment                             ! Menu !
+-----+-----+-----+-----+-----+-----+
Command => _____ !           _ SAP                 ! _____
Enter-PF1---PF2---PF3 !                                     ! -PF11--PF12---
Help Add End +-----+-----+-----+-----+ NxtSt Menu
↵

```

2 Choose an entry and press ENTER.

Depending on the operating system selected, a **Network Defaults** window similar to the examples below opens:

For BS2000:

```

+-----+
|                                     |
|                               Network Defaults (BS2000) |
|                                     |
| Network      B60-FLOW   Version           Owner EXAMPLE |
| Description  Job Flow, BS2000 |
| JCL Node    N0194           Execution Node N0082 |
|                                     |
| Default Values for the Jobs |
| Esc Activation _           Submit User ID _____ |
|   Esc Submit  _           SYSOUT User ID _____ |
| Default User ID NOP_____ |
|   JCL User ID _____ |
|   Job Class   _____ |
| Account Number _____ |
|                                     |
| --PF1-----PF3-----PF5----- |
| Help      End       Save |
|                                     |
+-----+

```

For z/OS:

```

+-----+
|                                     |
|                               Network Defaults (z/OS) |
|                                     |
| Network      B60-FLOW   Version           Owner EXAMPLE |
| Description  Job Flow, BS2000 |
| JCL Node    N0194           Execution Node N0082 |
|                                     |
| Default Values for the Jobs |
| Esc Activation _           JCL User ID _____ |
|   Esc Submit  _           Submit User ID _____ |
|                                     |
| --PF1-----PF3-----PF5----- |
| Help      End       Save |
|                                     |
+-----+

```

For z/VSE:

```
Network Defaults (z/VSE)

Network      B60-FLOW      Version              Owner EXAMPLE
Description  Job Flow, BS2000
JCL Node     N0194              Execution Node N0082

Default Values for the Jobs
  Esc Activation _
    Esc Submit  _

--PF1-----PF3-----PF5-----
  Help      End      Save
```

For UNIX and Windows:

```
Network Defaults (UNIX, Windows)

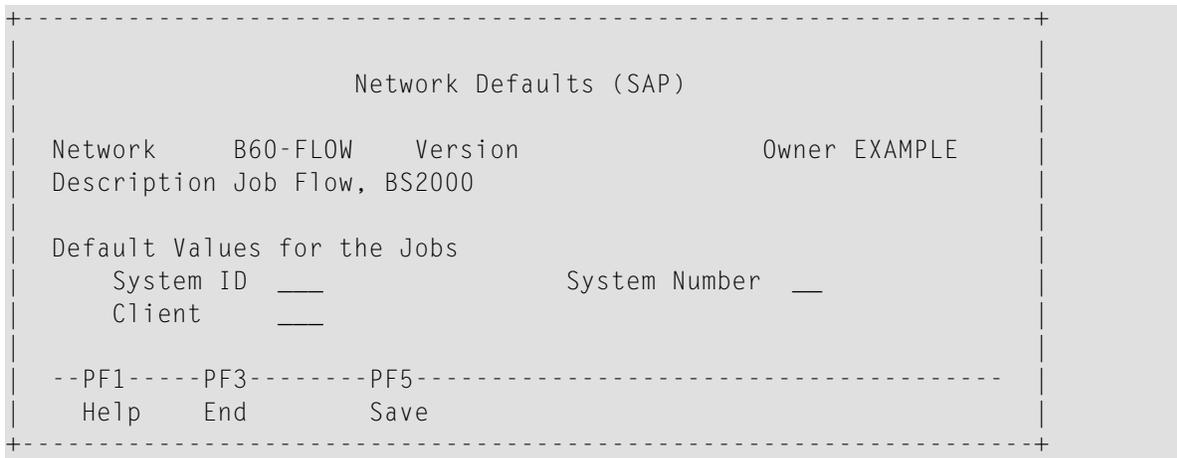
Network      B60-FLOW      Version              Owner EXAMPLE
Description  Job Flow, BS2000
JCL Node     N0194              Execution Node N0082

Default Values for the Jobs
  Esc Activation _
    Esc Submit  _
  JCL User ID  _____
  JCL  Group   _____

  Submit User ID _____
  Submit  Group _____

--PF1-----PF3-----PF5-----
  Help      End      Save
```

For SAP:



3 Enter the required values.

For more information about the specific operating system settings, see the [Input Fields: Network Defaults](#).

4 Press PF5 (Save) to save your changes and then PF3 (End) to close the window.

This section covers the following topic:

Input Fields: Network Defaults

The input fields available in the **Network Defaults** window depend on the operating system selected as indicated in the following table:

Field	Description
Esc Activation	Activation Escape Character = Network default value specific for operating system This escape character is the prefix for Natural code lines and symbols to be replaced at activation time. Caution: Existing Dynamic JCL might become invalid after changing this escape character and applying defaults to jobs.
Esc Submit	Submission Escape Character = Network default value specific for operating system This escape character is the prefix for Natural code lines and symbols to be replaced at submission time. Caution: Existing Dynamic JCL might become invalid after changing this escape character and applying defaults to jobs.
JCL User ID	(Not applicable to z/VSE) The user ID to be used for JCL loading BS2000:

Field	Description
	<p>If this field is not empty, the JCL is loaded with the rights of this BS2000 user ID. It can be overwritten with specific definitions. TSOS may only be defined if the user himself has logged on under TSOS. Default: The user ID from the fully qualified file name.</p> <p>If this field is left empty in a job definition, then the Default User ID will be inserted during the activation of the job.</p> <p>z/OS:</p> <p>JCL in z/OS will be loaded under this user ID by the Entire Operations Monitor. You can define this field only if you are logged on to the JCL node with the same user ID.</p> <p>Default: If this field is left blank, the user ID of the last modification will be used as JCL user ID.</p> <p>UNIX and Windows:</p> <p>With the authorization of this user ID, the Entire Operations Monitor loads the JCL of type TXT.</p> <p>See also the default setting User ID Definition (<i>Default Setting (1), Administration</i> documentation), Operating System User IDs and Default User ID Determination.</p>
Submit User ID	<p>(Not applicable to z/VSE)</p> <p>The user ID to be used at job start</p> <p>BS2000:</p> <p>The Entire Operations Monitor starts jobs in BS2000 under this user ID. In the network definition, this is a default value for the jobs.</p> <p>If this field is left empty in a job definition, then the Default User ID is inserted during the activation of the job.</p> <p>z/OS:</p> <p>The Entire Operations Monitor starts jobs in z/OS under this user ID. You can only define this user ID if you have logged on to the executing node with the same user ID.</p> <p>Default: If this field is empty, the user ID of the last modification is taken as submit user ID.</p> <p>UNIX and Windows:</p> <p>With the authorization of this user ID, the Entire Operations Monitor starts the script or the executable program.</p> <p>See also the default setting User ID Definition (<i>Default Setting (1), Administration</i> documentation), Operating System User IDs and Default User ID Determination.</p>
Applies to BS2000 only:	

Field	Description
Default User ID	This user ID is a default for all objects of this job network or job which are linked to a user ID.
Job Class	This job class is a default for all jobs in the network. It can be overridden by specific definitions.
Account Number	This account number is a default for the Submit User ID defined for the job network. It can be overridden by specific definitions.
Job Priority	If not empty, this job priority will be used during submission and will override a possible setting in the LOGON statement. The default value on network level will be used for new job definitions.
Run Priority	If not empty, this run priority will be used during submission and will override a possible setting in the LOGON statement. The default value on network level will be used for new job definitions.
SYSOUT User ID	This is the user ID under which internal SYSOUT files are created by Entire Operations. If you do not enter an ID here, the Submit User ID is used. See also the default setting User ID Definition (<i>Default Setting (1), Administration documentation</i>), <i>Operating System User IDs</i> and <i>Default User ID Determination</i> .
SYSOUT Cat ID	This is the catalog ID under which internal SYSOUT files are created by Entire Operations. This field is meaningful only if a SYSOUT user ID different from the submit user ID is used.
Applies to UNIX and Windows only:	
JCL Group	Applies to UNIX only (optional) If this field is left blank, the standard group of the UNIX user ID is used as defined under /etc/passwd. Otherwise, this field must contain one of the groups issued by the UNIX command groups.
Submit Group	Applies to UNIX only (optional) If this field is left blank, the standard group of the UNIX user ID is used as defined under /etc/passwd. Otherwise, this field must contain one of the groups issued by the UNIX command groups.
Applies to SAT only:	
System ID	SAP System ID (SID), as required by jexa4S.
System Number	SAP System Number, as required by jexa4S.
Client	SAP Client Symbol replacement is possible (except at logon).

Specifying Recipients for Network Messages

This function is used to specify single or multiple users to receive network-related messages that are sent when a specified event occurs. This is especially useful for sending information about the abnormal end of a job.

Among the events, which can cause a message to be sent, are:

- a calendar not defined for next year;
- a network not correctly terminated;
- symbol prompting requests.

The specific message text is generated automatically by the Entire Operations Monitor.

➤ To define a message recipient

- In the **Network Modification window**, press PF11 (MsgRe).

A **Message and Message Recipients** window similar to the example below opens:

> To add or modify a granting definition

- 1 Type G in the line command field of the selected network on the **Network Maintenance screen**.
- 2 Press ENTER.
- 3 A **Granting Definition** window similar to the example below opens:

```

17-07-10          ***** En +-----+
Owner SAGTEST          Ne !
Selection OR_____ !      Owner SAGTEST   Network B60-FLOW !
-----+-----+ !      Version (unnamed) !
Cmd #Run Owner      Network  N !      Granting Definition !
*-----* *-----* !      -----+-----+ !
G P      SAGTEST    B60-FLOW  !      Cmd Type  Name      Option !
_      SAGTEST    SAG-PROMPT !      _ Owner   SAGTEST__  O_____ !
_      1 SAGTEST    SAGNET      !      _ User_   DEMO_____  R_____ !
_      1 SAGTEST    SAGNET-HEB !      _ Owner   EXAMPLE___  D_____ !
_      SAGTEST    SAGNET-P   5 !      _ _____  _____  !
_      1 SAGTEST    SAGNETHEB !      _ _____  _____  !
_      SAGTEST    SAGNETSUB1 !      _ _____  _____  !
_      SAGTEST    SAGNETSUB2 !      _ _____  _____  !
_      SAGTEST    SAGNET2   !      _ _____  _____  !
_      !      _ _____  _____  !
_      !      _ _____  _____  !
***** !      -----+-----+ !
A Active C Copy D Delete F Flow !      D Delete !
P Descr R Activate S Schedule T !      !
Command => _____ !      Enter-PF1---PF3-----PF5-----PF7---PF8--- !
!      Help  End      Save  Up  Down !
+-----+-----+
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11--PF12---
Help  Add  End      Save  Up  Down      NxtSt Menu

```

- 4 Type in the user ID or owner you want to authorize network access and the level of access to be granted.

The columns and options available are described in *Fields: Granting Definition*.
- 5 Press PF5 (Save) to save the definition.
- 6 You can enter up to 30 different names and options. If current window is full, press PF8 (Down) to scroll forward to next window. Press PF7 (Up) to scroll back to previous window.
- 7 Press PF3 (Exit) to return to the **Network Maintenance** screen.
- 8 The network appears on the **Network Maintenance** screen when the authorized user logs on to Entire Operations. You can also use this function to delete existing authorizations.

> **To delete a granting definition**

- In the **Granting Definition window**, type D (Delete) next to the user or owner whose definition you want to remove and press ENTER.

The users and owners you can remove depend on your user authorization. For the delete restrictions that apply, see *Deleting a User/Owner Link* in the *Administration* documentation.

Fields: Granting Definition

The columns and input options in the **Granting Definition window** are described in the following table:

Column/Field	Description	
Cmd	One-character line command input field. Possible option: D to delete a specified user or owner.	
Type	Specifies the object to be authorized. Possible options:	
	Owner	All users linked to the Entire Operations owner specified in the Name field.
	User	A defined user.
	You can enter O for Owner or U for User .	
Name	The user or owner to be granted access. If the access rights are given to an owner, all linked users can obtain these access rights. Use an asterisk (*) for selection.	
Option	Level of access granted to a user or owner. You can specify one or more (for example RA for read and activate access) of the following authorization levels:	
	R	Read access to the network (no maintenance).
	W	Read and write access to the network (maintenance allowed except delete network).
	D	Read, write and delete access to the network.
	O	Read, write and delete access to the network. In addition, the specified user or owner can also allow network access to other users and owners.
	A	User is allowed to activate the network.

Documenting Your Networks

You can add a **brief description** of a job network when defining a network in the **Network Addition/Modification** window. This short description appears in the list of networks on the **Network Maintenance screen**.

If you wish to add more online documentation for a network, you can invoke the Entire Operations editor.

➤ To create or modify a detailed description for a job network

- 1 Type P in the line command field of the selected network.
- 2 Press ENTER.
- 3 The Entire Operations editor screen appears:

```

Edit Description Nw: E01-C0-148----- ==> Description saved
====>                                     Scroll====> CSR
***** ***** top of data *****
00001 Network 'E01-CONTI'
00002 -----
00003 This network consists of 6 independent jobs. Each of them forces
00004 a defined event at end-of-job time. Each job executes the
00005 program 'NOPCONTI' which resides on the installation library
00006 'NOPnnn.LOAD'. The necessary JCL which is named as 'NOPE*' is stored on
00007 the installation library 'NOPnnn.SRCE'.
00008
00009 This is a short description to enable the user to define his
00010 own test examples.
00011
00012 Log Jobname          Event at End-of-job
00013 -----            -----
00014
00015 +-----+
00016 ! E01-J01 !          Job ended ok.
00017 +-----+
00018      !
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      End      Rfind Rchan Up      Down      Impo      Left      Right Curso

```

You can enter and edit text using editor commands. For a detailed description of the Entire Operations editor, see the *Software AG Editor* documentation.

Once a text is written, it can be read by any user who is authorized to access the network. The line command `P` entered for the network in the list of networks displays the current text.

You can also display or print online documentation in the *Entire Operations Reporting Facility* (**Reports** option on the Main Menu - see the section *Reporting*).

You can also create online documentation at the job level (see the section *Job Maintenance*).

Deleting a Job Network

Only authorized users can delete a network. Only networks without active runs and version usage can be deleted.

Deleting a network also deletes all definitions made for the network on lower levels using network and job maintenance functions.

Entire Operations saves the last run number of a deleted network. The first run of a new network under the same name receives the last run number incremented by 1.

> To delete a network

- 1 Type `D` in the line command field of the selected network on the **Network Maintenance screen**.
- 2 Press `ENTER`.
- 3 If several network versions exist, a network version selection window opens. Select the version you want to delete.
- 4 A window opens in which you can confirm the deletion by entering the network name.
- 5 Type in the network name and press `ENTER` to delete the network.

12 Copying Job Network Definitions; Cloning Network

Versions

- Fields: Network Definition Copy 141

When defining a new network, you can select an existing network and all its definitions from the **Network Maintenance** screen as a model for the new network.

➤ **To copy a network definition**

- 1 On the **Network Maintenance screen**, type C in the line command field next to the network to copy, and press ENTER.

Or:

Alternatively, you can press PF10 (Copy) from the **Network Modification window** of the network to copy.

- 2 A **Network Master Definition Copy** window similar to the example below opens:

```

16.03.16          ***** Entire Operations *****          15:12:37
Owner EXAMPLE          Network Maintenance
Selection OR_____
-----
Cmd #Run Owner      Network  Node  Description
*-----*
_      EXAMPLE  ADMIN   N0146 Administrator workshop
_      EXAMPLE  ADMIN2  N0146 Administrator workshop
c P    5 +-----+
_ P    |
_ P    |          Network Master Definition Copy
_ P    |
_ P    |          From                      To
_ P    |          Owner ==> EXAMPLE_____ Owner ==> EXAMPLE
_      |          Network ==> B60-FLOW___ Network ==> _____
_      |          Version ==> _____ Version ==> _____
_ P    |
***** |          Alt. File ==> N (Y/N)          with Schedule ==> N (Y/N)
A Activ |          with Grants ==> N (Y/N)          act
P Descr |          PF1 Help          PF3 End
Command +-----+
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Add End Save Up Down NxtSt Menu
    
```

Enter the required definitions. The fields available in the window are described in **Fields: Network Definition Copy**.

- 3 When you are finished, press ENTER to copy the network.

The window closes and new network appears in the list on the **Network Maintenance** screen.

Fields: Network Definition Copy

The fields of the **Network Definition Master Copy** window are described in the following table.



Note: You cannot copy a network to an existing network, or a version to an existing version.

Field	Description				
From	<p>The fields under From contain the name of the owner, network and network version from which the new network is copied.</p> <p>You can change the names as required.</p> <p>To open a selection window of available networks, use an asterisk * as a wildcard.</p> <p>To clone a network version, make sure that the From and To owners and networks are identical, and enter a different To network version.</p>				
To	<p>The fields under To contain the name of the owner, network and network version to be used for the new network.</p> <p>You can change the name of the network and network version as required.</p> <p>To open a selection window of available networks, use an asterisk * as a wildcard.</p> <p>To clone a network version, make sure that the From and To owners and networks are identical, and enter a different To network version.</p>				
Alt. File (secondary file)	<p>If this field is marked with Y, the definition is copied from an alternate (secondary) Entire Operations system file, which has to be assigned in the Entire Operations defaults.</p> <p>Note:</p> <ol style="list-style-type: none"> 1. If an alternate data file is defined in the Entire Operations Defaults (submenu of the System Services Menu), you can copy a network definition from there by setting the Alt. File field to Y. 2. The alternate file must be an Entire Operations system file. The network definitions must be created in the current version of Entire Operations. 3. When you are copying from a secondary file or between different owners, all defined symbol tables are also copied, however, only if they do not already exist in the target environment. 				
with Schedule	<p>Possible values:</p> <table border="1"> <tbody> <tr> <td>N</td> <td>The schedule linked to the source network will not be copied (default).</td> </tr> <tr> <td>Y</td> <td>The schedule of the source network will also be copied. The current day will be excluded to prevent an unwanted immediate activation. The current schedule table will be copied from the day following the current</td> </tr> </tbody> </table>	N	The schedule linked to the source network will not be copied (default).	Y	The schedule of the source network will also be copied. The current day will be excluded to prevent an unwanted immediate activation. The current schedule table will be copied from the day following the current
N	The schedule linked to the source network will not be copied (default).				
Y	The schedule of the source network will also be copied. The current day will be excluded to prevent an unwanted immediate activation. The current schedule table will be copied from the day following the current				

Field	Description	
		day until the end of the current year. The execution history will not be copied.
with Grants	N	Other users/owners granted access to the source network will not be copied to the new network (default). The new network will only have access rights for the current user/owner. See also <i>Granting Definition: Authorizing Other Users or Owners to Access a Network</i> .
	Y	Other users/owners granted access to the source network will be copied to the new network, in addition to the current user/owner.

13

Displaying the Job Flow within a Network

This function provides a short overview of the job flow within a network. The output can be sent to the screen or to the defined printer.

➤ **To view the job flow within a network**

- 1 Enter `F` in the line command field of the appropriate network.
- 2 Press `ENTER`.
- 3 A window opens at the bottom of the screen:

```

17-07-11          ***** Entire Operations *****          14:28:30
Owner EXAMPLE          Network Maintenance
Selection OR_____
-----
Cmd #Run Owner      Network      Node  Description
*-----*
F L      EXAMPLE     B60-FLOW    42  BS2000 Job Flow
_ P      EXAMPLE     E01-CONTI   148 Completion-Codes, Job Duration
_ P      EXAMPLE     E02-IOC-01  148 I/O Conditions without events
_ P      EXAMPLE     E03-IOC-02  148 I/O Conditions with code events
_        EXAMPLE     E04-IOC-03  148 I/O Conditions with string events
_        EXAMPLE     E05-IOC-04  148 I/O Conditions with user routine event
_ P      EXAMPLE     E10-PAR-01  148 Parallel & sequential job execution
_        EXAMPLE     E20-DYN-01  148 Dynamic JCL generation
_ P      EXAMPLE     E40-REC-01  148 Cause a recovery situation
_ P      EXAMPLE     E40-REC-02  148 Recovery and Reschedule
_ P      EXAMPLE     E50 +-----+
***** ! ! *****
A Active C Copy D Dele !      Send Output to Printer ? N (Y/N) ! act
P Descr R Activate S S !      !
Command => _____ +-----+
-----
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Add   End       Save       Up       Down       NxtSt Menu

```

- 4 Enter Y if you want to send the output to the printer assigned to Natural Printer (1), as specified in your Natural profile parameters.

Or:

- 5 Enter N if you want to view the job flow of the network.

A screen similar to the example below appears:

```

17-07-11                Entire Operations                14:46:51
                        Job Flow of Network B60-FLOW    Owner EXAMPLE    Page 1
                        Version
Job      by Condition      from/to Job      Owner      Network
-----
(1)  JOB-SUBNET
.....
(2)  JOB-01
      I      E60-JOB1-0
      +-----> E60-JOB1-0 -----> (7)  JOB-019
      +<----- E60-JOB1-0 <----- (13) JOB-1-TEST
      +<----- JOB-02-OK <----- (8)  JOB-02
      +<----- E60-JOB3-0 <----- (9)  JOB-03
(3)  JOB-012
      I      E60-J012-0
(4)  JOB-013
      I      E60-J013-0
(5)  JOB-014
      I      E60-J014-0
(6)  JOB-015
      +-----> E60-J015-0 -----> (8)  JOB-02
.....
      +<----- E60-JOB1-0 <----- (2)  JOB-01
      +<----- E60-JOB1-0 <----- (13) JOB-1-TEST
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                        End                                Down                                Menu

```

6 Press PF8 (Down) or PF7 (Up) to scroll down or up in the screen.

 **Note:** You can also generate a report from a job flow overview. See the report type **Network Job Flow Display** described in the section [Reporting](#).

14

Checking for a Loop in a Job Network

If you type H in the line command input field for a network, Entire Operations runs a check of the links between the jobs.

- If a loop in the job flow is detected, the following message appears:

```
Definition Loop in xxx/yyy
```

where:

xxx is the name of the network tested for loops, and

yyy is the name of the job in the network, in which a cyclic link was first detected.

When a loop has been detected, a corresponding message is written to the Entire Operations log.

- If no loop is detected, the following message appears:

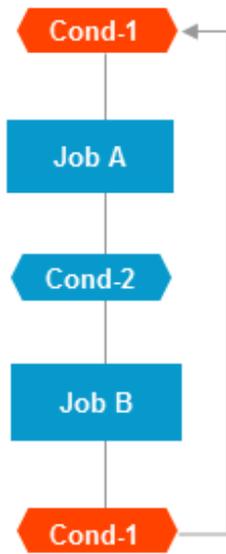
```
No Loop in Network xxx
```

- If the network contains more than 1000 jobs and no loop has been detected in these first 1000 jobs, the following message appears:

```
Loop check limit 1000 exceeded. Loop check stops.
```

Entire Operations does not perform a loop test on more than 1000 jobs.

The following figure illustrates a loop within the job flow of a network.



15

Maintaining the Usage of Network Versions

- Network Version Usage List 150
- Adding or Modifying a Version Usage Definition 151

Network Version Usage List

This function enables you to define the maintenance of date ranges, in which network versions are to be used for scheduled activations.

```

17.12.13          ***** Entire Operations *****                17:22:07
                   Network Version Usage for scheduled Activations
Owner SN          Network A-1
-----
Cmd from         to          Version   Description
-----
_ >>>>>>>> 16.12.13 v004
_ 17.12.13 20.12.13 v20131217
_ 21.12.13 31.12.13 V00002
_ 01.01.14 >>>>>>>> V00003

***** Bottom of Data *****
D Delete M Modify
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Add End Up Down Left Right
    
```

Function	Description
>>>>>>>> in the from column	Activate without start date. If not start date is defined, the current day is assumed by default.
>>>>>>>> in the to column	Activate without end date (unlimited).

Line Command	Description
D	Deletion of a definition.
M	Modification of a definition.

New definitions can be made with PF2.

 **Note:** The date ranges must not overlap.

Adding or Modifying a Version Usage Definition

This function enables you to define the date range, in which a network version is to be used for scheduled activations.

```

17.12.13          ***** Entire Operations *****          17:35:31
                  Network Version Usage for scheduled Activations
Owner SN          Network A-1
-----
Network Version ==> _____
is to be used for scheduled activations
                  from ==> _____ (empty: unlimited)
                  to ==> _____ (empty: unlimited)

Version Usage Description
-----

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

Field	Description
Network Version	A network version can be selected by using a wildcard.
Default Usage from	The start date of a default usage period of a network version. Empty: no start date defined. The current day is assumed by default.
Default Usage to	The end date of a default usage period of a network version. Empty: no end date defined (infinite time).

16 Applying Network Defaults to Jobs (Mass Update)

You can perform a mass-update of job attributes within a network.

➤ To apply defaults modified in the network definition to all jobs in the network

1 Press PF9 (DfJb) on the **Network Modification window**.

A screen similar to the example below appears:

```
Owner EXAMPLE      Network B60-FLOW  Version
Application of Network Defaults to Jobs

C Field            old      new          C Field      old  new
_ Symbol Table                    _ Exec Node 31  42
_ SymTab Version                    _ JCL  Node 31  42
_ JCL  User ID                      _ JCL  Loc      NAT
_ JCL  Group                          _ Esc Act      @
_ Submit User ID                      _ Esc Sub      $
_ Submit Group                          _ Job Prio
_ BS2000 Def.User ID                      _ Run Prio
_ BS2000 Job Class                      _ EJA  E-M      N
_ BS2000 Acct.Number
_ SYSOUT User ID                      NOP
_ SYSOUT Cat ID
_ Modifying User      (all)    NATQA5
_ File
  old
  new SYSEORU

S only if same old  A all
-----
PF1 Help      PF3 End      PF6 SAP
```

All network master definitions/fields on this screen are described in the section *Fields and Commands: Network Definition*.

- Whether or not the new value replaces the old value is determined by a line command entered in the line command input field in the window (column headed **C**). This allows you to distinguish between those jobs which have the defaults set for the network and those jobs for which specific values have been defined.

The following line commands are available:

Line Command	Description
S	Replace value only if the old value matches the old default. Specific definitions made at the job level remain unchanged.
A	Replace all values regardless of whether they are default settings or specific definitions.

Enter the required line command next to the value you want to replace or leave the line command field blank for a value you do not want to replace.

- Press **ENTER** when you are finished.

The specified values are now copied and updated as default values for all jobs of the network.

A message is returned indicating the number of jobs that have been modified.

For each job for which values were replaced, the name of the definition that changed and the new default value are listed in the Entire Operations log with a message like `Network default Activation Escape @ applied`. See also [Displaying Logged Information](#) in the section *Log Information*.

17

Displaying Network Accounting Data

- Displaying Accounting Data for Jobs in a Selected Network 156
- Special PF Key: Job Accounting Data 157

 **Note:** You can also generate reports from accounting data (see [Example of Accounting Data](#) as described in the section *Reporting*).

Displaying Accounting Data for Jobs in a Selected Network

 **Note:** This function is also available for jobs: see [Viewing Job Accounting Data](#) in the section *Schedule Maintenance*.

> To display accounting data for jobs in a selected network

- 1 Type T in the line command field of the selected network.

Press ENTER.

The following window opens in which you can specify dates, times and run numbers for the jobs:

```

17-07-12          ***** Entire Operations *****          12:21:07
Owner EXAMPLE          Network Maintenance
Selection OR_____
-----
Cmd #Run Owner      Network      Node  Description
*-----*-----*
T L      EXAMPLE    B60-FLOW    42  BS2000 Job Flow
_ P      EXAMPLE    E01-CONTI   148 Completion-Codes, Job Duration
_ P      EXAMPLE    E02-IOC-01  148 I/O Conditions without events
_ P      EXAMPLE    E03-IOC-02  148 I/O Conditions with code events
_        EXAMPLE    E04-IOC-03  148 I/O Conditions with string events
_        EXAMPLE    E05-IOC-04  148 I/O Conditions with user routine event
_ P      EXAM +-----+ ion
_        EXAM !
_ P      EXAM ! Owner EXAMPLE Network B60-FLOW !
_ P      EXAM ! Date / Run Number Selection !
_ P      EXAM !
***** ! From Date 17-07-12 00:00 to 17-07-12 12:26 ! *****
A Active C C ! From Run 1_____ to 99999 ! N Deact
P Descr R Ac ! ! story
Command => __ ! PF3 End ! ___
+-----+
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Add  End      Save      Up      Down      NxtSt Menu
    
```

The input fields are explained in [Fields: Range Specification for Accounting Data](#) in the section *Schedule Maintenance*.

2 Enter dates, times and run numbers and press ENTER.

A **Job Accounting Data** screen appears.

For explanations of the columns on the **Job Accounting Data** screen, see *Fields and Columns: Reports* in the section *Reporting*.

Special PF Key: Job Accounting Data

You can perform the following function from the **Job Accounting Data** screen using this PF key:

PF Key	Name	Function
PF4	Net	Alternate display: <ul style="list-style-type: none"> ■ Job and network data together (as show in the example of a Job Accounting Data screen). ■ Network data only.

18 Listing Jobs for a Network

➤ To list jobs defined for a network

- Type **L** in the line command input field of the selected network on the **Network Maintenance screen**. Use this line command to maintain job definitions.

If several network versions exist, a network version selection window opens first. Select a version.

For more information, see the section *Job Maintenance*.

➤ To list active jobs for a network

- 1 Type **A** in the line command field of the selected network.
- 2 Press **ENTER**.
- 3 An **Active Jobs** screen (similar to the example shown in the section *Listing All Active Jobs for a Single Network Active*) appears with a list of active jobs for the selected network.

Within this list, you can perform the same functions as if you had selected the option **Active Job Networks** from the Main Menu.

For more information, see *Listing All Active Jobs for a Single Network Active* in the section *Active Job Networks*.



Note: All job and network definitions are stored on the master database. All active jobs and networks are maintained on the active database.

19

Activating a Job Network Manually

- Fields: Network Activation 164

The Entire Operations Monitor will **activate** job networks automatically according to their defined schedules. However, it is also possible to activate a network manually from the **Network Maintenance screen**.

When a network is activated, all jobs within the network, except recovery jobs, are activated.



Note: Only authorized users can activate job networks manually.

There are several situations in which you may wish to activate a job network manually (press ENTER to activate the network):

- When no schedule has been defined for the network;
- To start the same network several times (with different run numbers);
- When the network is not scheduled for the current date;
- To reactivate a network.

➤ To activate a network manually

- 1 Type R (Activate) in the line command field of the selected network, and press ENTER.
- 2 A **Network Activation** window similar to the example below opens:

```

06.06.16          ***** Entire Operations *****          14:20:50
Owner EXAMP +-----+
Selection 0 |
-----+-----+
|                               Network Activation          |-----+
Cmd #Run Ow |
| * - |                               Owner ==> EXAMPLE      |
| _   EX |                               Network ==> B60-FLOW |
| _   EX |                               Version ==> _____|
r P   6 EX |                               No Network Versions |
| _   EX |
| _   EX |                               Last Schedule Extract ==> 06.06.16
| _   EX |                               at ==> 00:00:39
| _   EX |                               Last Execution Date ==> 06.06.16
| _   EX |                               Last Run Number ==> 2254
| _   EX |
| _   EX |                               Preferred Run Number ==> _____
| _ P   EX |                               Use Time from Schedule ==> N (Y/N)
*****          or activate at Date ==> 06.06.16__          *****
A Active C |                               Time ==> 14:20:58          Deact
P Descr R |                               JCL Check only ==> N (Y/N)          ory
Command => |                               Enter----PF1---PF3-----
|                               Activate Help End
+-----+-----+

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Add End Save Up Down NxtSt Menu
  
```

- 3 Change the input fields as required. The fields are explained in [Fields: Network Activation](#).

Press ENTER when you are finished.

If symbol prompting has been defined for the network, a [Symbol Prompting for Table screen](#) appears with a table of symbols used during dynamic JCL generation.

If required, change the symbols and symbol values. For further explanations, see [Symbol Prompting during Network or Job Activation](#) in the section [Symbol Table and Symbol Maintenance](#).

- 4 Press PF5 (Acpt) when you are finished.

If the activation is successful, the activated network gets a new run number as returned in a message.

Fields: Network Activation

The fields in the **Network Activation window** are described in the following table:

Field	Description
Owner	Owner of the network to be activated.
Network	Network to be activated.
Version	<p>Version of the network to be activated.</p> <p>If a current version exists for the current date, it will be preset as the default value. It is possible to activate any other existing version of the network.</p> <p>When using a wildcard, a network version selection window will open.</p> <p>Below the version field, an explanative text for the version will be shown (if applicable).</p> <p>Example:</p> <pre>17.12.13 - 20.12.13 - current at 18.12.13</pre>
Last Schedule Extract at	<p>Date and time of the last automatic schedule extraction for this network by the Entire Operations Monitor.</p> <p>See also Date and Time Formats.</p>
Last Execution Date	<p>Date of the last activation.</p> <p>See also Date and Time Formats.</p>
Last Run Number	Run number of the last activation.
Preferred Run Number	<p>If input is allowed for this field, you can enter the run number you want to be used for the activated network. Run numbers for further network activations increment from this number.</p> <p>If the requested run number is in use, Entire Operations assigns the next free number to the run.</p> <p>If this field contains a zero (0) or no value, Entire Operations (as usual) determines the number to be assigned to the run.</p> <p>Field input is allowed or not allowed (default) in the network default settings: see the option Activation: Allow run number setting in the section <i>Default Setting (4)</i> in the <i>Administration</i> documentation.</p>
Use Time in Schedule	<p>Enter Y (yes) here to activate the network within the defined time frames (Earliest Start, Latest Start, Deadline), even if the current date is not a scheduled date for this network. With this option, you can force the same time dependencies as if the network were scheduled and activated automatically.</p>

Field	Description
or activate at Date/Time	<p>Shows the current date and time. You can modify the date and time to force activation at any time on any date. There is no limitation for future date and time settings.</p> <p>See also Date and Time Formats.</p>
JCL Check only	<p>If set to Y (yes):</p> <p>Only a JCL check is performed for the job network or job. The required commands are automatically inserted:</p> <pre>z/OS, JES2: TYPRUN=SCAN z/OS, JES3: EXEC PGM=JCLTEST BS2000: /MODIFY-SDF-OPTIONS MODE=TEST UNIX: Script execution with set -vn Windows: Jobs are executed as Dummy due to JCL Check</pre> <p>Note:</p> <ol style="list-style-type: none"> Active conditions used by the network execution for JCL check only do not interfere with active conditions of “real” active jobs or active networks. These active conditions are marked with the prefix (C) in all screens and lists. <hr/> <p>If set to N (no):</p> <p>Normal job submission is initiated.</p> <p>This is the default.</p>

VI Job Maintenance

This section describes the functions available to create, maintain and delete Entire Operations objects of the type job.

For general information on jobs, see the section *Jobs* in the *Concepts and Facilities* documentation.

[Defining and Managing Jobs](#)

[Defining Job Types and Job Execution Features](#)

[Using a Dummy Job](#)

[Defining a Subnetwork](#)

[Defining the Parameters for an FTP Job](#)

[Defining and Managing JCL for a Job](#)

[Editing Master JCL and Natural Sources](#)

[Defining and Managing Job Input Conditions](#)

[Handling Prerequisite Resources for a Job](#)

20

Defining and Managing Jobs

- Listing Jobs 170
- Displaying and Modifying a Job Definition 174
- Adding a Job Definition 181
- Writing and Viewing Online Documentation for a Job 183
- Deleting a Job Definition 185
- Activating a Single Job Manually 185
- Displaying Job Dependencies and Connecting or Disconnecting Jobs 188
- Defining Job-Specific Log Information 194

Listing Jobs

This section describes how to list all master jobs defined for a network.

Related Topics:

- [Listing Active Jobs](#) in the section *Active Job Networks*
- [Listing Jobs of a Subnetwork](#)

> To list all jobs defined for a network

- 1 On the **Network Maintenance screen**, type L in the line command field of the network for which you wish to list all jobs, and press ENTER.

Or:

Use the direct command LIST JOBS as described in the *Direct Commands* documentation.

If several network versions exist, a **Network Version Selection** window similar to the example below opens first:

```

16-11-21          ***** Entire Operations *****          13:39:27
Owner SA +-----+-----+-----+-----+-----+-----+
Selectio !
----- !
                        Network Version Selection
Cmd #Run !  Owner SAGTEST   Network SAGNET
!          !  Version      Usage
_ P    4 !  _    V2          16-07-01 - 16-12-31 - current at 16-11-21
_      !  _    (unnamed)  unnamed version (without name)
_      !  _    V1          16-01-01 - 16-06-30
|      8 !
_      4 !
_      !
_      3 !
_      1 !
_      2 !
_ P   25 !
_ P    !
***** !
A Activ !          ** Bottom **
P Descr !  -----PF3-----PF7---PF8-----
Command !          End          Up      Down
+-----+-----+-----+-----+-----+-----+

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Add   End       Save       Up      Down          NxtSt Menu
  
```

The window shows a list of the versions defined for a network.
 If a current version exists for the current date, it will be presented in the first line.
 The other versions appear in alphabetical order after the current version.
 If necessary, use PF7 to scroll up or PF8 to scroll down.

- 2 Select the required network version by marking it with any character and press ENTER.

A **Job Maintenance** screen similar to the example below appears:

```

17-03-29          ***** Entire Operations *****          18:04:06
Job Maintenance   Owner SAGTEST   Network B60-FLOW   Version
-----
Cmd  C  R  PU  Job      Type  Loc  Description          File or Library  Member
-----
_   C1R1  JOB-01  SRV      Where it all starts
_   R1P   JOB-012 JOB BS2 Depending on Job-01  $NOP.SYSEORU
_   C1    JOB-013 JOB MAC Depending on JOB-012 SYSEORU          B60-M01
_   JOB-015 DUM      Depending on JOB-014
_   C1    JOB-019 JOB MAC Depending on JOB-01  SYSEORU          B60-M01
_   C2    JOB-02  JOB MAC Dep. JOB-15, JOB-19 SYSEORU          B60-M02
_   C1    JOB-03  NAT NAT Depending on JOB-02  SYSEORU          B60-P01
_   C1R1  JOB-04  JOB MAC Depending on JOB-03  SYSEORU          B60-M01
_   C1    JOB-05  DUM      Depending on JOB-04
_   JOB-06  JOB MAC Where it all ends  SYSEORU          B60-M02
_   C1    J2-SUB  NET      Subnetwork Job for Su SAGTEST/SAGNETSUB (D)
_   C1    SUBNETJOB1 NET      Subnetwork job for su SAGTEST/SAGNETSUB (D)
***** Bottom of Data *****
A Depend. C Copy D Delete E Edit G Pregen. I Input Cond. J JCL L Resources
M Mod. O EOJ Chk + Act P Prose R Activate S Sched.Parms U Ext.Log Z Subnet
Command => _____

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Add   End           Save           Up           Down           Menu
    
```

This screen displays all jobs defined for the selected network.

The screen columns are described in [Columns: Job Maintenance](#).

This section covers the following topics:

- [Columns: Job Maintenance List](#)
- [Line Commands: Job Maintenance](#)

Column	Description
	For possible JCL locations, see List of JCL Locations .
Description	The short description of the job. Note: The description is truncated if it exceeds 21 characters.
File or Library	Physical storage of JCL according to the JCL location . For a job of the type NET (subnetwork), this column contains the name of the owner and the name of the subnetwork defined for the job (see the jobs J2 - SUB and SUBNETJOB1 in the example of a Job Maintenance screen).
Member	The member which contains the JCL. This refers to the JCL location . For possible members, see List of JCL Locations . For a job of the type NET (subnetwork), this column contains a letter (for example, (D)) denoting the subnetwork activation mode .

Line Commands: Job Maintenance

You can perform several functions on any job listed on the [Job Maintenance screen](#) using line commands. The following line commands are available:

Line Command	Description
A	Display job dependencies (previous and following jobs). Includes connection and disconnection of jobs. See Displaying Job Dependencies and Connecting or Disconnecting Jobs .
C	Copy a job definition to a new job, including all associated definitions.
D	Delete a job definition , including its input conditions and End-of-Job checking and actions.
E	Edit JCL or Natural program (editor).
G	Pregenerate active JCL .
I	List input conditions for the job.
J	Define the JCL for the job .
L	Handle prerequisite resources for the job .
M	Modify a job definition. See Displaying and Modifying a Job Definition .
O	Define End-of-Job checking and actions .
P	Invoke the editor to write a long description for the job. See Writing and Viewing Online Documentation for a Job .
R	Activate a single job .
S	Define scheduling parameters for the job. See Maintaining Schedule Dependencies for a Job .
U	Define extended logging. See Defining Job-Specific Log Information .
Z	List jobs in subnetwork (for jobs of type NET only). See Listing Jobs of a Subnetwork .

Special PF Keys: Job Maintenance

The PF keys of the job maintenance screen have the standard behavior, except the following:

PF Key	Description
PF3	Leave the job maintenance screen. If the job maintenance screen has been invoked by the line command Z (Subnet) for a subnetwork job, you will return to the job maintenance screen of the calling network.

The following sections give detailed descriptions of the functions you can perform on each job using line commands.

Displaying and Modifying a Job Definition

➤ To view and modify a job definition

- 1 On the **Job Maintenance screen**, type M in the line command input field next to the job you want to view or modify.
- 2 Press ENTER.
- 3 A **Job Definition (Master)** window opens containing the current values for the job:

```

12.11.15          ***** Entire Operations *****          18:01:52
Job Maintenance   Owner EXAMPLE      Network B60-FLOW      Version
-----
+-----+
|                                     Job Definition (Master)                                     |
|                                                                              |
| Job Name      ==> JOB-01_____      Mod ==> SAG      24.02.15 12:16      |
| Description    ==> Where it all starts_____      | 2 |
| Job Type      ==> JOB      |
| Execution Node ==> N0082 MVS/ESA      | 1 |
|                                                                              | 1 |
| Special Type  ==> _____      Symbol Table ==> EXAM-ST1__      |
| Restartable   ==> _      Symbol Table Version ==> _____      | 1 |
|               Suffix Symbol ==> _____      | 2 |
|               Escape Characters:  Activation ==> @      | 1 |
|               Submit              ==> $      | 1 |
| End-of-Job Action Errors set 'not ok' Conditions ==> _      | ** |
| Enter-PF1---PF2--PF3--PF4---PF5---PF6---PF7---PF8---PF9-----PF12- |
|               Help  Add  End  Edit  Save  Spec  Symb  Net  JCL      Menu  |
+-----+

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Add  End          Save          Up  Down          Menu

```

You can change the values in the fields described in the section [Fields: Job Definition \(Master\)](#).



Note: Depending on your job type or operating system, you may have to define additional parameters. Press PF6 (Specials) to open a window with the additional parameters. For further information, see [Displaying Job Dependencies and Connecting or Disconnecting Jobs](#).

- 4 Press PF5 (Save) to save all changes.
- 5 Press PF3 (End) to return to the [Job Maintenance screen](#).

This section covers the following topics:

- [Fields: Job Definition \(Master\)](#)

- [Special PF Keys: Job Definition \(Master\)](#)

Fields: Job Definition (Master)

The fields in the **Job Definition (Master) window** are described in the following table.



Note: The fields and tabbed pages available depend on the **Job Type**, the **Special Type** and the operating system of the **Execution Node** specified for the job.

Field	Description
Job Name	<p>Job name.</p> <p>Entire Operations recognizes a job by this name. It can, but need not be the same name as the JCL source used by the job (the operating system recognizes a job by its JCL name).</p> <p>Note: Job names should start with a character. Job names starting with a digit are tolerated, but are not recommended.</p>
Description	<p>Short description of the job.</p> <p>The maximum input is 50 characters.</p> <p>This text appears in the list of jobs on the Job Maintenance screen.</p> <p>Note: The description is truncated on this list after 21 characters.</p> <p>You can add a long description for the job: see Writing and Viewing Online Documentation for a Job.</p>
Job Type	<p>The job type as defined to Entire Operations. For further information, see the section Job Types.</p>
Special Type	<p>Leave this field empty if this is a normal job or enter one of the following values:</p> <ul style="list-style-type: none"> ▪ (empty field) Normal job without special condition. ▪ C - Cyclic execution (This is a replacement and extension of the former CYC (cyclic) job type.) <p>Execution at certain intervals. If this type is specified, a cyclic interval must be defined in the job schedule parameters. This is checked during job activation and before job start.</p> <p>Each job type can be provided with this attribute.</p> <p>Note: You can use the reserved condition <code>P-STOPCYC - jobname</code> to interrupt a cyclic job execution loop. See also reserved condition names in <i>Restrictions for Condition Names</i>.</p>

Field	Description						
	<ul style="list-style-type: none"> <p>■ D - Execute as dummy Execution as a dummy job. This allows you to prevent the job from being executed without having to delete the definition. You can also set this type in an active job before a repetition.</p> <p>If this value is deleted in an active job, then the active JCL is generated, provided that it is already present. However, this does not have a resubmission of the job as a consequence.</p> <p>To resubmit the job, use the line command R in the list of active jobs in this case as well. See <i>Resubmitting an Active Job</i> in the section <i>Active Job Networks</i>.</p> <p>■ P - Stop Job Job type STC: Stops a started task.</p> <p>Job type SRV: Stops a Windows service.</p> <p>■ R - Recovery Job For a recovery job. You must also use this special type for jobs that are later activated with the job import API NOPUJIXN (see the section <i>API Routines</i>).</p> <p>Note: Recovery jobs are not activated during a normal job network activation. They are activated only if they are defined for another job and if a certain event makes a recovery necessary.</p> 						
Execution Node	<p>The job is submitted on the node specified in this case. The default is the value specified in the job network definition. You can enter a different node for the job in this case. The name of the operating system appears after the node number.</p> <p>Note: In this field, you can also enter PMPA preceded by an escape character, for example, \$PMPA. See <i>\$PMPA</i> in <i>Predefined Symbols</i> in the section <i>Symbol Table and Symbol Maintenance</i>).</p>						
Symbol Table	<p>Name of the symbol table to be referenced for substitution of variables in dynamic JCL.</p> <p>Enter an asterisk (*) and press ENTER to list available symbol tables for network owner. Select a name from the list. Alternatively, you can enter a new name and press PF7 (Symb) to define a new symbol table. See <i>Modifying a Symbol Table Master</i> in the section <i>Symbol Table and Symbol Maintenance</i> for more information.</p> <p>Note: If no symbol table is defined on the job level, the symbol table on the network level (if defined) will be activated during network activation or job activation (if defined).</p>						
Symbol Table Version	<p>Version of the symbol table to be used.</p> <p>Reserved names (will be replaced).</p> <table border="1" data-bbox="402 1604 1472 1766"> <tbody> <tr> <td data-bbox="402 1604 932 1682">current</td> <td data-bbox="932 1604 1472 1682">Current version for the activation date or determination date.</td> </tr> <tr> <td data-bbox="402 1682 932 1730">nv</td> <td data-bbox="932 1682 1472 1730">Version of the using network.</td> </tr> <tr> <td data-bbox="402 1730 932 1766">svn</td> <td data-bbox="932 1730 1472 1766">Symbol table version of the using network.</td> </tr> </tbody> </table>	current	Current version for the activation date or determination date.	nv	Version of the using network.	svn	Symbol table version of the using network.
current	Current version for the activation date or determination date.						
nv	Version of the using network.						
svn	Symbol table version of the using network.						
Suffix Symbol (optional)	<p>If several instances of the job are to be activated in parallel, this field must contain a symbol name which is in the defined symbol table. The symbol itself must contain the suffixes to be</p>						

Field	Description
	<p>appended to active job names and output conditions. The sum of the lengths (prefix+suffix) must not exceed the maximum field lengths for job names or conditions.</p> <p>For example, if the master job name is <code>PARA</code>, and the symbol contains <code>001, 003, 012</code>, then the active jobs <code>PARA001</code>, <code>PARA003</code> and <code>PARA012</code> are created. If the suffix symbol value is not defined in a network and the network has a calling network (if it is a subnetwork), then the search for the suffix symbol is performed upward up to the highest level.</p> <p>If Suffix Symbol does not contain any values, the job is activated as a temporary dummy job. For information about the definition of multiple symbol values, see Defining Multiple Symbol Values in the section <i>Symbol Table and Symbol Maintenance</i>. For information about the assignment of multiple symbol values to Suffix Symbol, see Symbol Replacement with Multiple Symbol Values.</p>
Restartable	<p>Only applies to BS2000.</p> <p>The job can be restarted automatically. Possible values:</p> <ul style="list-style-type: none"> ■ Y - Restart The job is to be restarted without any recovery after a system crash. ■ N - No restart No automatic restart after a system crash. ■ R - BS2000/RESTART The job is to be restarted when repeated. SYSOUT files are not renamed. <p>Automatic SYSOUT rerouting can be delayed with the following line right after the <code>/LOGON</code> instruction:</p> <pre data-bbox="334 1173 1385 1205">/REMARK EOR-SYSOUT-DIRECT=LATER</pre> <p>Later, the following line must appear:</p> <pre data-bbox="334 1310 1385 1341">/REMARK EOR-SYSOUT-DIRECT=NOW</pre> <p>This allows user-defined actions before using a SYSOUT file.</p> <p>Additional Restart Criteria:</p> <p>Job submission time must be before the last IPL time and the job must not be terminated. The field MonJV must contain \$R. The system session number at the job check time must be different from the session number at submission time.</p>
Mod	User ID, date and time of the last modification of the job definition.

Field	Description
End-of-Job Action Errors set 'not ok' Conditions	<p>Job output conditions. Possible values:</p> <ul style="list-style-type: none"> ■ N - No impact on the job result Errors during End-of-Job action processing will have no impact on the job result. ■ Y - Errors impact the job result Errors during End-of-Job action processing will have an impact on the job result. <p>Output condition(s) for the job will be set after all End-of-Job actions are performed. This includes End-of-Job action (EJA) exits. Any error in the End-of-Job action processing will cause the setting of the conditions for job not ok. If the job was already set to not ok, it will remain not ok, regardless of the definition here.</p> <p>Note:</p> <ol style="list-style-type: none"> 1. Setting this flags can cause a longer elapsed time of a network, because the condition setting waits for the termination of the other End-of-Job actions. 2. The network level setting will be overridden by job settings. <ul style="list-style-type: none"> ■ empty field - Inherited from the network If the field is empty, the job will inherit the condition from its network at activation time. This is the default.
Milestones	<p>Milestone job types.</p> <p>You can set single or multiple milestone attributes (types) for a job that performs an important task within a job network, such as tracking significant events or controlling the successful completion of other jobs.</p> <p>Possible values for user-defined milestones:</p> <ul style="list-style-type: none"> ■ B - Network Begin The milestone job performs as the first network job. ■ E - Network End The milestone job performs as the last network job. ■ I - Other The milestone job performs in no particular job order. <p>Possible values for system milestones dynamically set by Entire Operations during job submission (applies to active jobs only):</p> <ul style="list-style-type: none"> ■ 1 - Network Begin The milestone job performs as the first network job. ■ 2 - Network End The milestone job performs as the last network job. <p>You can use the application programming interface NOPUMI1N (see the section <i>API Routines</i>) to define your own milestones.</p>

Field	Description
	For further information on milestone jobs, see the section <i>Event Store Milestones</i> in the <i>System Automation Tools</i> documentation.
Escape Characters:	This is the activation escape character . It is a prefix for Natural code lines and symbols to be replaced at activation time. If you change this character, dynamic JCL can become invalid.
Activation	See also <i>Symbol Escape Characters (Symbol Maintenance)</i> and <i>Notes about Escape Characters</i> .
Escape Characters:	This is the submission escape character . It is a prefix for symbols to be replaced at submission time. If you change this character, dynamic JCL can become invalid.
Submit	Default is the dollar sign (\$). See also <i>Symbol Escape Characters (Symbol Maintenance)</i> and <i>Notes about Escape Characters</i> .

Note about Escape Characters

The job escape characters will always be used for symbol replacements and macro JCL.

When a job is created or when an escape character is reset, the default value is taken in the following order from:

1. **Network defaults (operating system specific)** if defined for the current operating system;
2. **Network definition** if non-blank;
3. Global operating system specific defaults for z/OS and z/VSE, BS2000, UNIX and Windows (*Administration* documentation).

See also *Symbol Escape Characters* in the section *Symbol Maintenance*.

Special PF Keys: Job Definition (Master)

You can perform the following functions using these PF keys in the **Job Definition (Master) window**:

PF Key	Name	Function
PF4	Edit	Edit the JCL. You can edit the job's JCL directly from here. Make sure that you have defined JCL in advance. The JCL definition can be invoked with PF9 from this screen.
PF6	Spec	Define job execution features. Depending on whether you press PF6 in the Specials for Job Type window or in the Specials, Execution window, you can define job execution features specific to the job type or operating system. The input fields available are described in <i>Fields: Job Type Specific Execution Features</i> and <i>Fields: Operating System Specific Execution Features</i> .

PF Key	Name	Function
PF7	Symb	Open the Usable Symbol Tables window, and select a symbol table for browsing or modification. See also Listing Usable Symbol Tables in the section <i>Symbol Table and Symbol Maintenance</i> .
PF8	Net	Job type NET : See Defining a Subnetwork .
PF9	JCL	Define the JCL for a job .

Adding a Job Definition

When defining a new job, you can either enter all required attributes individually, or copy all attributes from an existing job definition.

➤ To add a job definition

- 1 On the **Network Maintenance** screen, type L in the line command field of the network for which you wish to add a job and press ENTER.

Or:

Use the direct command `ADD JOB` as described in *Direct Commands* documentation.

The **Job Maintenance** screen appears.

- 2 Press PF2 (Add).

A **Job Definition (Master)** window similar to the example below opens:


```

17.12.13          ***** Entire Operations *****          16:59:26
Job Maintenance   Owner SN          Network A-1          Version v20131217
-----
Cmd C R PU Job          Type Loc Description          File or Library,Member
*-----
c   C5 PU A-1          JOB PDS missing member test  SN.SYSF.SRCE          0000000
_   C5   A-2          DUM   ic def test
_   BBBB          JOB
_   +-----+
_   !
_   !          Job Master Definition Copy          ! SS-T0
_   !          !          ! SS-T1
_   C   !          !
_   !   From          To          ! DL001
_   !   Owner    ==> SN_____   Owner    ==> SN          !
_   !   Network  ==> A-1_____   Network  ==> A-1          ! 0990A
_   !   Version  ==> v20131217_   Version  ==> v20131217  ! 0990B
_   !   Job      ==> A-1_____   Job      ==> _____  !
***** !   Alt. File ==> N (Y/N)          ! *****
A Depend !          !          ! s
M Mod. 0 ! PF3 End          ! t
Command +-----+
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Add End          Save          Up          Down          Menu

```

You can copy the job under another name from the same or another network or network version defined for the same or another owner. You can only specify another owner if you are authorized to access the networks of the owner.

- 3 Press ENTER to save the new definition and close the window.

The new job appears in the list of jobs in the [Job Maintenance screen](#) for the owner specified for the new job.



Note: Copying a job does not include the definitions made at job level via the [Job Maintenance screen](#), such as input conditions and End-of-Job handling. Use the available line commands to create these definitions for each new job.

Writing and Viewing Online Documentation for a Job

You can add a short description of a job when defining a job in the [Job Definition \(Master\) window](#). This short description appears in the list of jobs on the [Job Maintenance screen](#).

If you wish to add more online documentation for a job, proceed as described in the following instruction.

➤ **To create, view or modify a long job description**

- 1 On the **Job Maintenance screen**, type P in the line command input field next to the required job.
- 2 Press ENTER.

The description is displayed on an editor screen similar to the example below:

```

Edit Description Jb: ABC3A Nw: EORACCT----- Columns ←
001 072
=====>                                SCROLL====> CSR
***** ***** top of data *****
00001 Job 'ABC3A'
00002 -----
00003 This job forces a defined event at end-of-job time and executes the
00004 program 'EORCONTI' which resides on the installation library
00005 'EORnnn.LOAD'. The necessary JCL which is named 'EORE*' is stored in
00006 the installation library 'EORnnn.SRCE'.
***** ***** bottom of data *****
    
```

The example above shows the screen with 6 lines of text.

(If no documentation exists for the selected job, the editor screen appears without text.)

Write new or replace existing text using editor commands which are briefly described in the online help. For detailed descriptions of all editor functions, see *Software AG Editor* in the *Natural* documentation.

- 3 Press PF3 when you are finished.

The text is saved and the editor window closes.

On the **Job Maintenance** screen, the letter P in the **P** column next to the modified job indicates that a long description exists for this job.

Once online documentation is written, it can be read by any user who is authorized to access the network to which the job belongs. Use the line command P to display the current text.

You can also display or print online documentation by using the **Reports** facility described in the section [Reporting](#). If long job descriptions exist, they are included in the **Network Description (detailed)** report as shown under **item 5** of the *Example of Network Description (detailed)*.

Deleting a Job Definition



Note: Deleting a job definition also deletes all definitions made at the job level, including its input conditions and End-of-Job checking and actions. The original JCL is not deleted.

> To delete a job definition

- 1 On the **Job Maintenance screen**, type `D` in the line command input field next to the job you want to delete.
- 2 Press `ENTER`.

A window prompts you to confirm the deletion by entering the job name.
- 3 Press `ENTER` to delete the job definition.

Activating a Single Job Manually

Just as you can perform an ad-hoc manual **activation** of a job network at any time to allow it to run outside of its scheduled times, you can also manually activate any job at any time, regardless of its position within a job network.

The activated job is assigned its own run number (in sequence with the network run number) and, if you do not change the date or time, is submitted immediately by the Entire Operations Monitor. If the job contains symbols as placeholders for variables, symbol prompting takes place as described in the section *Symbol Prompting during Network or Job Activation*. If other symbol tables are required, they will be prompted too.

The differences between the manual activation of a single job and the manual activation of a job network are that when a single job is activated:

- no check for defined input conditions for the job is performed;
- no output conditions are set by the job.

All other End-of-Job actions defined for the job are executed.

> To activate a job manually

- 1 Type `R` in the line command input field for the selected job in the **Job Maintenance screen**.
- 2 Press `ENTER`.

A **Job Activation** window similar to the example below opens:

```

06.06.16          ***** Entire Operations *****          18:18:36
Job Maintenance   Owner EXAMPLE   Network B60-FLOW   Version
-----
Cmd C R PU Job      Type Loc Description          File or Library,Member
*---+-----+-----+-----+-----+
_      P ABC3      |                               |                               |
_      HEB-      |                               |                               |
_      C1 HEB-      |                               |                               |
r      C1 JOB-      |                               |                               |
_      C1 P JOB-      |                               |                               |
_      C1 JOB-      |                               |                               |
_      C1 P JOB-      |                               |                               |
_      C1 JOB-      |                               |                               |
_      C1 JOB-      | Preferred Run Number ==> _____ |                               |
_      C3 JOB-      | Use Time from Schedule ==> N (Y/N)   |                               |
_      C4 JOB-      | or activate at Date ==> 06.06.16__  |                               |
_      C1 JOB-      |                               |                               |
_      C1 JOB-      |                               |                               |
*****          |                               |                               |
*****          |                               |                               |
A Depend. C Cop | Enter----PF1---PF3----- | CL L Resources
M Mod. 0 EOJ Ch | Activate Help End          | t.Log Z Subnet
Command => _____+-----+-----+-----+-----+

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Add End Save Up Down Menu
  
```

This section covers the following topics:

- [Fields: Job Activation](#)

Fields: Job Activation

The fields of the [Job Activation window](#) are described in the following table:

Field	Description
Owner	Owner of network in which job is to be activated.
Network	Network of the job that is to be activated.
Version	Network version of the job that is to be activated.
Job	Job to be activated.

Field	Description
<p>Preferred Run Number</p>	<p>If input is allowed for this field, you can enter the run number you want to be used for the activated job. Run numbers for further job activations increment from this number.</p> <p>If the requested run number is in use, Entire Operations assigns the next free number.</p> <p>If this field contains a zero (0) or no value, Entire Operations (as usual) automatically assigns a run number.</p> <p>Field input is allowed or not allowed (default) in the network default settings: see the option Activation: Allow run number setting in the section <i>Default Setting (4)</i> in the <i>Administration</i> documentation.</p>
<p>Use Time in Schedule</p>	<p>Enter Y (yes) here to activate the job within the defined time frames (earliest start, latest start, deadline), even if the current date is not a scheduled date for this network. With this option, you can force the same time dependencies as if the network was scheduled and activated automatically.</p> <p>For more information on network time frames, see <i>Activating a Job Network Manually</i> in the section <i>Network Maintenance</i>.</p>
<p>or activate at Date/Time</p>	<p>Shows the current date and time. You can modify date and time to force activation at any time on any date. There is no limitation for future date and time settings.</p> <p>See also <i>Date and Time Formats</i>.</p> <p>Note: No symbol prompting is performed for later activations.</p>
<p>JCL Check only</p>	<p>If set to Y (yes):</p> <p>Only a JCL check is performed for the job network or job. The required commands are automatically inserted:</p> <pre>z/OS, JES2: TYPRUN=SCAN z/OS, JES3: EXEC PGM=JCLTEST BS2000: /MODIFY-SDF-OPTIONS MODE=TEST UNIX: Script execution with set -vn Windows: Jobs are executed as Dummy due to JCL Check</pre> <p>Note:</p> <ol style="list-style-type: none"> Active conditions set or reset by job execution for JCL check only do not interfere with active conditions of “real” active jobs or active networks. These active conditions are marked with the prefix (C) in all screens and lists. <hr/> <p>If set to N (no):</p> <p>Normal job submission is used for network activation.</p> <p>An actual JCL check is not performed. This is the default.</p>

Displaying Job Dependencies and Connecting or Disconnecting Jobs

The **Job Maintenance** facility allows you to see which jobs must run before a specific job can start, and which jobs run after it.

You can trace job dependencies through whole networks and display cross-network job links.

You can connect or disconnect jobs from the same or different networks.



Note: An overview of the job flow for the whole network is available from the **Network Maintenance** screen described in *Displaying the Job Flow within a Network*.

- [Displaying Job Dependencies for a Specific Job](#)
- [Connecting Jobs from the Same or Different Networks](#)
- [Disconnecting Jobs](#)

Displaying Job Dependencies for a Specific Job

➤ To display job dependencies for a specific job

- 1 On the **Job Maintenance screen**, type `A` in the line command input field next to the required job.
- 2 Press `ENTER`. A **Job Dependencies** window opens:

```

23.10.08          ***** Entire Operations *****          15:35:15
Owner SN          Job Maintenance          Network BIG-1
+-----+-----+-----+-----+-----+-----+-----+
!
!           Predecessors           Job Dependencies           Successors           !
! -----+-----+-----+-----+-----+-----+-----+ !
! M Network   Job           Output Condition           !
!   Input Condition           Network   Job           M !
! _ BIG-1     JOB-013       => E60-J014-0           !
!   E60-J013-0           =>           BIG-1     JOB-015   _ !
!           +-----+-----+-----+-----+-----+-----+ !
!           => Ow   SN           =>           =>           _ !
!           Nw   BIG-1       =>           =>           _ !
!           Job JOB-014           =>           =>           _ !
!           +-----+-----+-----+-----+-----+-----+ !
!           =>           =>           =>           _ !
!           =>           =>           =>           _ !*
! -----+-----+-----+-----+-----+-----+-----+ !
! X Chain   D Disconnect           X Chain   D Disconnect           !
! Enter----PF1----PF2-----PF3-----PF5-----PF7---PF8----- !
E !           Help Connect End           Save           Up           Down           !-
+-----+-----+-----+-----+-----+-----+-----+

```

The selected job name and network name appear in the middle of the window with its dependencies displayed in the two smaller windows on either side of the selected job name:

- The **Predecessors** window on the left of the selected job contains the job and network names of the jobs that must run before the selected job can run. Directly beneath these are the linking conditions.
- The **Successors** window on the right of the selected job contains the job and network names of the jobs for which the selected job is a prerequisite. Directly above these are the linking conditions.

You can scroll both lists by using PF7 (Up) and PF8 (Down).

Line Commands: Job Dependencies

In the **M** column of the **Job Dependencies window**, you can enter the following line commands to perform the described functions on any job displayed in the window:

Line Command	Description
D	Disconnect the job from a predecessor or successor job. See also <i>Disconnecting Jobs</i> .
X	Chain. Move the selected job into the center of the window and display its dependencies.

Special PF Keys: Job Dependencies

You can perform the following function from the **Job Dependencies window** using this PF key:

PF Key	Name	Function
PF2	Connect	Connect the job to a predecessor job. For the standard case, event Job OK. <i>See Connecting Jobs from the Same or Different Networks.</i>

Connecting Jobs from the Same or Different Networks

You can connect two jobs within the same or different networks without defining logical conditions for them by using the Entire Operations default condition.

This function is also useful to link jobs across networks. In this case, you should check the input condition reference of the successor job. Use a reference different from RUN.



Note: After connecting jobs within a network, a loop check is performed for the affected network. The same conditions apply as described in *Checking for a Loop in a Job Network* in the section *Network Maintenance*, with one exception: if a loop is detected in the job flow, no corresponding message appears.

➤ To connect two jobs

- Press PF2 (Connect) in the **Job Dependencies window** of the job you wish to connect.

A **Job Connection** window opens in which you can define a job which must precede the selected job:

```

10.02.10          ***** Entire Operations *****          16:31:28
Owner SN          Job Maintenance          Network A-1

+-----+-----+-----+
! EOR1141 - No Dependencies found          !
!      Predecessors          Job Dependencies          Successors          !
! -----+-----+-----+          !
! M Network      Job      I      I      Output Condition          !
!      Input Condition      I      I      Network      Job      M          !
! -              I              I=>          !
!              =>I              I          !
! - +-----+-----+          !
! -              !              !          !
! -              !              Job Connection          !
! -              !              !          !
! -              !      from  Owner: SN_____  to  Owner: SN          !
! -              !      Network: A-1_____  Network: A-1          !
! -              !      Job: _____  Job: D1          !
! -              !      Event: OK          !
! ----          !              !          !
! X Ch          !      PF3 End          !          !
! Ente +-----+-----+          !
E !              Help  Connect  End      Save      Up      Down          !
+-----+-----+-----+

```

The identifiers of the selected job appear in the protected fields on the right in the window.

This section covers the following topics:

- [Fields: Job Connection](#)

Fields: Job Connection

The input fields in the **Job Connection** window have the following meaning:

Field	Description
from/ Owner	Enter owner of job to run as predecessor.
from/ Network	Enter network name of job to run as predecessor.
from/ Job	Enter name of job to run as predecessor.
Event	OK Job ended ok The condition <i>predecessor-job-OK</i> is added as an input condition to the successor job for the event job ok.
	NO Job ended not ok The condition <i>predecessor-job-NOT-OK</i> is added as an input condition to the successor job for the event job not ok.

Field	Description
	If an output condition already exists for a mentioned event, it is used for the linkage (instead of creating a new condition).

Disconnecting Jobs

➤ **To disconnect two jobs linked by conditions**

- 1 In the **Job Dependencies window**, type D in the line command input field of the second job in the sequence.
- 2 Press ENTER.

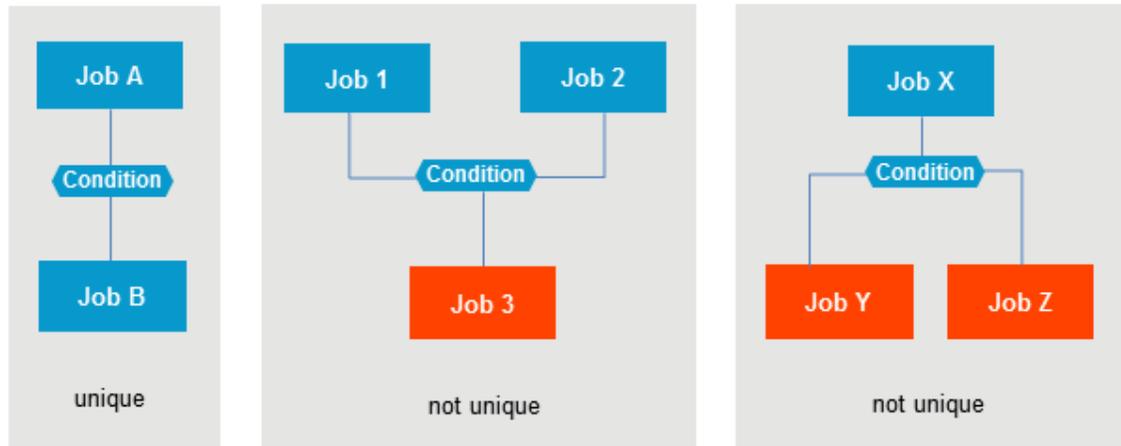
A confirmation window opens:

```

23.10.08          ***** Entire Operations *****          15:35:41
Owner SN          Job Maintenance          Network BIG-1
+-----+-----+-----+-----+-----+-----+-----+-----+
! NOP1457 - Several Predecessor Jobs          !
!           Predecessors          Job Dependencies          Successors          !
! -----+-----+-----+-----+-----+-----+-----+-----+ !
! M Network      Job           Output Condition          !
!   Input Condition          Network      Job      M !
! _ BIG-1        JOB-012          => E60-J013-0          !
!   E60-J0 +-----+-----+-----+-----+-----+-----+-----+ !
! _ BIG-1      !           !           !           !           !
!   E60-J0 ! Please confirm          !           !           !           !
! _          ! the Disconnection from JOB-014          !           !
! _          ! by entering 'Y' ====> _          !           !
! _          !           !           !           !           !
! _          ! PF3 End          !           !           !           !
! _          +-----+-----+-----+-----+-----+-----+-----+ !
! _          =>          !           !           !           !           !
! _          +-----+-----+-----+-----+-----+-----+-----+ !
! X Chain D Disconnect          X Chain D Disconnect          !
! Enter-----PF1---PF2-----PF3-----PF5-----PF7---PF8----- !
E !           Help Connect End          Save          Up      Down          !-
+-----+-----+-----+-----+-----+-----+-----+-----+
  
```

- 3 Enter Y.
- 4 Press ENTER to perform the disconnection and close the window. You can disconnect any two jobs linked by conditions using this function. There is no need to delete any defined conditions.

 **Note:** No disconnection is performed if the use of the conditions is not unique, for example:



 **Note:** Remove the condition definitions manually if the link is not unique.

Disconnection from a Predecessor Job

When disconnecting from a predecessor job, the linking input condition definition of the job displayed in the center (of the **Job Dependencies** window) is deleted.

Disconnection from a Successor Job

When disconnecting from a successor job, the linking output condition definition of an event definition of the job displayed in the center (of the **Job Dependencies** window) is deleted.

 **Note:** The input condition definitions of the successor job remain unchanged. After the output condition definition is deleted, the successor job can wait for a condition which is never satisfied.

Restrictions:

- A linking condition is deleted, only if it links no more than two jobs.
- If the X line command (Chain) was invoked for jobs in different networks, a disconnection of this type is not possible.

Defining Job-Specific Log Information

Entire Operations allows you to define which information is to be logged in addition to the Entire Operations default system log.

This information is defined at the job level and can be job SYSOUT, JCL or selected operating system messages from the SYSOUT.

This section covers the following topics:

- [Specifying Extended Log Information](#)
- [Fields: User-Defined/Extended Log](#)

Specifying Extended Log Information

➤ **To specify extended log information**

- 1 Type U in the line command input field of the selected job on the **Job Maintenance screen** and press ENTER.

An **Extended Log** screen opens in which you can specify the information to be logged:

```

16.01.14          ***** Entire Operations *****          15:00:28
                        Extended Log
Owner SN          Network A-1          Version V00003          Job IEFBR14
-----
      Log SYSOUT ==> Y          (Y/N)
      Files ==> _____
      Log JCL ==> Y          (Y/N)

Log System Messages  M Code      M Code      M Code      M Code
_ IEF403I            _____
_ IEF404I            _____
                    _____
                    _____
                    _____

                        S Message Contents Selection

Command=> _____
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help          End          Save          Menu
    
```

- Specify the information to be logged.

The input fields and options are explained in *Fields: User-Defined/Extended Log*.

- Press PF5 (Save) to save your entries.
- Press PF3 (End) to return to the **Job Maintenance screen**.

The logged information can be viewed after job termination using the **Log Information** facility described in the section *Log Information*.

Fields: User-Defined/Extended Log

The input fields of the **Extended Log** screen are described in the following table:

Field/Column	Description	
Log SYSOUT	Specifies logging of job SYSOUT files after job termination. Possible values:	
	Y	Logs SYSOUT.
	N	No SYSOUT files are logged.
	SYSOUT logging can also be defined as SYSOUT action as described in the section <i>End-of-Job Checking and Actions</i> . If the SYSOUT file or spool data set exceeds a given line limit, the log display will be truncated: see SYSOUT Line Limit in <i>Default Setting (2)</i> in the <i>Administration</i> documentation.	
Files	(z/OS only) Input valid only if Log SYSOUT is specified. Enter number(s) of SYSOUT file(s) to be logged. If left blank and Log SYSOUT is specified, all files are logged.	
Log JCL	Specifies logging of JCL after job termination. This is useful for dynamically generated JCL or if JCL is frequently modified for different job runs. Possible values:	
	Y	Logs JCL.
	N	No JCL is logged.
	See also <i>JCL Log</i> .	
Log System Messages:	Specifies operating system messages to be logged. Enter a message code in a field under the Code column. If you enter S (Message Contents Selection) in the field headed M preceding a specified code and press ENTER, the Message Contents Selection window opens in which you can define additional selection strings. The message will be logged if no additional string is specified, or if at least one specified string will be found in the message.	

Message Contents Selection

```

+-----+
!
!           Message Contents Selection           !
!
! Network A-1           Version V00003           Job IEFBR14           !
!
! Log Message IEF403I  , if it contains one of the !
! following strings:                               !
! _____ !
! _____ !
! _____ !
! _____ !
! _____ !
!
! PF1 Help   PF3 End                               !
+-----+

```

System messages will be logged if they are selected by their message code, and one of the following conditions is true:

- No message contents selection is specified. (No selection strings defined.)
- A message contents selection is specified, and at least one of the strings is found within the pre-selected message.

JCL Log

The JCL log is generated from the following sources:

BS2000	From the ENTER file. (The ENTER file is deleted after copying into the Entire Operations Log.)
z/OS	If Entire Operations runs on a mainframe: from the SYSOUT. If Entire Operations does not run on a mainframe: from the active JCL.
z/VSE	From the active JCL.
UNIX	From the effectively submitted shell script.
Windows	From the effectively submitted BAT file.

21

Defining Job Types and Job Execution Features

- Available Job Types 198
- Defining Job Type Specific Execution Features 200
- Defining Operating System Specific Execution Features 204

You define a job according to the **job type** you require and the operating system on which your Entire Operations environment runs.

The functions and specification options available to define a job depend on the following entries in the **Job Definition (Master) window**:

- The job type entered in the **Job Type** field,
- The job type entered in the **Special Type** field, and
- The operating system of the node entered in the **Execution Node** field.

Depending on the job type, the special type and the operating system of the execution node entered in the **Job Definition (Master) window**, a special window is provided to define individual execution criteria for the job.

This section describes the types of jobs you can define in Entire Operations and the type specific features provided to execute a job.

Available Job Types



Note: Only the job types **JOB**, **SRV** and **STC** result in operating system jobs when submitted.

You can define the following job types to Entire Operations:

Job Type	Description
DAT	A UNIX, Windows or DAT text file.
Text File	<p>Jobs of type DAT (data file generation) can be used to create text files in the same way as active JCL is generated.</p> <p>Instead of a submission, the created text is just being written to a text (data) file.</p> <p>Symbol replacement, symbol functions, text block inclusions can be used like in other JCL.</p> <p>If the DAT output file is to be created on a mainframe node, no trigraph encoding will be performed during JCL loading.</p> <p>If the DAT output file is to be created on a UNIX or Windows node, trigraph encoding will be performed during JCL loading, and trigraph decoding will be performed during DAT file writing.</p> <p>For more information, see Trigraph Encoding for JCL Submission on UNIX and Windows in the section <i>System Overview</i>.</p>

Job Type	Description
DUM Dummy Job	<p>A dummy job has no JCL and is not submitted to the operating system.</p> <p>You can use dummy jobs to set input conditions with a time range, to provide a time window for other jobs or as placeholders for future or seldom used jobs, etc.</p> <p>You can define an estimated running time for a dummy job (see the field Estimated Elapsed Time). The dummy job is then executed for exactly this amount of time without burdening the system.</p> <p>Execution node:</p> <p>Even if a dummy job itself does not need it, it may be necessary to define an execution node with a Submit User ID. This is the case if any prerequisite check, End-of-Job check or End-of-Job action uses an Entire System Server node. The Monitor determines whether a node access is required, and performs Entire Systems Management logons only if they are necessary for any action.</p> <p>See also <i>Using a Dummy Job</i>.</p>
FTP File Transfer Job	<p>This job type can be used for file transfers in batch mode using the File Transfer Protocol (FTP). The JCL for the file transfer job will be generated by Entire Operations, depending on the FTP parameters in the JCL definition.</p>
JOB Standard Job	<p>This includes all standard operating system jobs with their own JCL. Jobs of the type JOB (Standard Job) are submitted as batch jobs. The JCL of JOB-type jobs can contain symbols as placeholders for variables defined in the symbol table specified for the job. These symbols are replaced by their current value at job activation or at job submission time. See also <i>Symbol Replacement in JCL</i>.</p> <p>If the job is executed on a UNIX or Windows node, command line parameters can be passed by the symbol CMDLINE-job. See <i>Predefined Symbols for Command Line Parameters</i> in the section <i>Symbol Table and Symbol Maintenance</i>.</p>
NAT Natural Program	<p>This is a Natural subprogram.</p> <p>It has no JCL and is executed directly by the Entire Operations Monitor.</p> <p>A subprogram, for example, is used to code user exit routines (see the relevant section) used by a job.</p> <p>See also <i>Editing Master JCL and Natural Sources</i>.</p>
NET Subnetwork	<p>This job type enables you to execute a complete network within a main network. See <i>Defining a Subnetwork</i>.</p> <p>The concept of subnetworks is described in detail in the section <i>Subnetworks</i>.</p>
SAP Job within SAP System	<p>The job will be executed within the SAP runtime system.</p> <p>See the SAP-specific network defaults described in <i>Input Fields: Network Defaults (Operating System and Environment Defaults)</i> and <i>Job Execution Specials for Job Type SAP</i>.</p>

Job Type	Description
SRV Windows Service	<p>With this job type, you can define a service under Windows. A service must first be defined in Windows before it can be started.</p> <p>For further information, see:</p> <ul style="list-style-type: none"> ■ Job Execution Specials for Job Type SRV ■ Passing Command Line Parameters <p>Stopping a Windows service: Create a copy of the main SRV job, and define the special type P described in <i>Fields: Job Definition</i>.</p>
STC Started Task	<p>A started task is a special type of batch job on z/OS or z/VSE. Started tasks have static JCL which can reside in any PROCLIB (z/OS) or POWER RDR (z/VSE).</p> <p>For z/OS: only PDS is a valid JCL location for this job type.</p> <p>Entire Operations can activate and process started tasks like any other job type. See also Passing Command Line Parameters.</p> <p>Stopping a started task: Create a copy of the main STC job, and define the special type P described in <i>Fields: Job Definition</i>.</p>
WPS Windows PowerShell Script	<p>Applies to Windows only.</p> <p>This job type is for Windows PowerShell Scripts (with extension .ps1).</p> <p>The Windows PowerShell (powershell.exe) will be used as command interpreter instead of the old Windows command interpreter. The script must conform to the PowerShell.</p> <p>Command line parameters can be passed by the symbol CMDLINE - job. See Predefined Symbols for Command Line Parameters in the section <i>Symbol Table and Symbol Maintenance</i>.</p>

Defining Job Type Specific Execution Features

➤ To define job type specific execution features

- 1 In the **Job Type** and **Special Type** fields of the **Job Definition window**, enter the required types. Press ENTER.
- 2 Press PF6 (Spec).

A **Specials for Job Type** window similar to the example of the job type DAT below opens:

```

Master Job Definition
Specials for Job Type DAT

Owner ==> SN                Job Type ==> DAT
Network ==> E41103-01       Exec Node ==> 194 BS2000
Job ==> DAT-BS2

Submit User ID ==> SN_____
Target Location ==> BS2      Target overwrite ==> Y
Target File ==>
$SN.E41103-01.OUT.1_____
Member ==> _____
Member Type ==> _____

Enter-PF1-----PF3-----PF5-----PF6-----PF12--
      Help       End       Save       Spec       Menu
    
```

Or:

For the job type NET, press PF8 (Net).

A **Job: Subnetwork Definition window** opens.

All type specific fields are described in *Fields: Job Type Specific Execution Features*.

If there are no type specific fields available, an appropriate message occurs indicating that this function is not available.

This section covers the following topics:

- [Fields: Job Type Specific Execution Features](#)

Fields: Job Type Specific Execution Features

Job Type	Field	Description
NET Subnetwork	JCL Node	Node where the JCL of the job (if defined) is located.
	Subnetwork Owner	Owner of the subnetwork. Use a wildcard to select an owner.
	Subnetwork Name	Name of the subnetwork. Use a wildcard to select a network.
	Subnetwork Version	Version of the subnetwork if defined.
	Subnetwork Run	Run number of the subnetwork (read-only field).

Job Type	Field	Description	
	Subnet Activation Mode	Activation time: see Time of Activation of a Subnetwork in the <i>System Overview</i> .	
		Possible values:	
		A (or blank)	At the moment of the activation of the calling job.
		S	At the moment of the start of the calling job.
	Append Mult. Suffix	Enter Y to append the suffix of the calling multiple job to the name of the subnetwork.	
		The resulting name must not have more than 10 characters. Otherwise, the name of the active job network is the same as the name of the master job network. Uniqueness is ensured by different run numbers.	
	Use Subnetwork Time Frames	Possible values:	
		N	Use the time frame from the calling job of the type NET (default).
		Y	Use the time frame defined for the subnetwork itself.
	Subnetwork Input Condition	Default input condition automatically set for a job of the type NET. See also Link to the Main Network .	
Subnetwork Output Condition	Default output condition automatically set for a job of the type NET. See also Link to the Main Network .		
DAT Text File	Submit User ID	The user ID which will be used to write a DAT text output file.	
	Target Location	AJC	Generate active JCL only.
		BS2	BS2000 text file.
		LMS	BS2000 LMS file.
		NAT	Natural source object.
		PDS	z/OS partitioned data set.
		TXT	UNIX or Windows text file.
	VSE	z/VSE sublibrary member with type.	
	Target overwrite	This option is available only for some target locations.	
Possible values:			
N		Append the output to target file.	
Y	Overwrite target file with the output.		

Job Type	Field	Description
	Target File	Text file to which the DAT output is to be written. Symbol replacement is possible. Additionally to symbols with activation escape characters, also symbols with submit escape characters will be resolved. Both resolutions will be performed at submission time.
	Member	The target locations LMS, NAT and PDS also require the definition of a target file member. Symbol replacement is possible. Additionally to symbols with activation escape characters, also symbols with submit escape characters will be resolved. Both resolutions will be performed at submission time.
	Member Type	BS2000, LMS: The LMS member type. For possible input values, see the description of the Type field in <i>Fields: Operating System Specials for JCL</i> . z/VSE: Member type of the z/VSE library.
SAP Job within SAP System	System ID	SAP System ID (SID), as required by jexa4S.
	System Number	SAP System Number, as required by jexa4S.
	Client	SAP Client Symbol replacement is possible (except at logon).
	SAP Job	Name of the job in the SAP system.
	Job Count	Job count assigned by the SAP system. This is a read-only field and applies to active (submitted) jobs only.
	Target Server	SAP Target Server, as required by jexa4S.
	SAP User ID	User ID (for the Job) in the SAP system.
	ciph. Password	Password in the SAP system. The ciphering is to be performed with the jexa4S utility encryptpw.
	ext. User ID	External user for SAP jobs. Default: Submit User ID of the Entire Operations job.
SRV Windows Service	Service Name	The internal name of a Windows service. Note: You can obtain the internal name of a Windows service by using the following Windows command line command: <code>sc getkeyname <external-name></code> .

Defining Operating System Specific Execution Features

➤ To define operating system specific execution features

- 1 In the **Job Type** field of the **Master Job Definition window**, enter the required **job type** and press ENTER.
- 2 Press PF6 (Spec).

If no job type specific features are available, the **Specials, Execution** window similar to the example below opens:

```

+-----+
!                                     !
!               Master Job Definition !
!               BS2000 Specials, Execution !
!                                     !
!           Owner ==> EXAMPLE           Job Type ==> JOB !
!           Network ==> B60-FLOW         Exec Node ==> 31 BS2000 !
!           Job ==> JOB-01               !
!                                     !
!   Default User ID ==> _____   Submit Password ==> !
!   Submit User ID ==> NOP_____   defined ==> no !
!   Account Number ==> _____   SYSOUT User ID ==> _____ !
!   BS2000 Job Class ==> _____   SYSOUT Cat ID ==> _____ !
!   Job Priority ==> _____   Collect SYSLST ==> N !
!   Run Priority ==> _____   Share SYSOUT ==> N !
!                                     !
!   MonJV ==> _____ !
!   Password ==> _____   defined ==> no !
!                                     !
! Enter-PF1-----PF3-----PF5-----PF12-- !
!           Help           End           Save           Menu !
+-----+

```

If job type specific features are available, the **Specials for Job Type** window opens first and you press PF6 (Spec) from this window.

The input fields provided depend on the operating system (here: BS2000) of the execution node specified for the job.

All operating system specific fields are described in [Fields: Operating System Specific Execution Features](#).

In general, these fields correspond to the operating system specific fields provided for network maintenance.

If no execution features can be specified for the operating system of the specified execution node, the following message occurs:

Function not available for ...

For the job type DAT, execution features are only available for the target location LMS on BS2000 only.

This section covers the following topics:

- [Fields: Operating System Specific Execution Features](#)

Fields: Operating System Specific Execution Features

The operating system specific input fields provided for job execution are described in the following table.

Operating System	Field	Description
BS2000	Default User ID	Valid logon user ID defined for BS2000. This ID has no meaning for other operating systems. Unqualified file and job variable names in this job definition are prefixed with this BS2000 user ID. Note: The user ID TSOS can be defined only if the user defining the ID is working under TSOS.
	Submit User ID	Jobs in BS2000 are submitted under this user ID by the Entire Operations Monitor. This ID has no meaning for other operating systems. Note: The user ID TSOS can be defined only if the user defining the ID is working under TSOS. In the Monitor standard values/submit user type in the access control system, certain checks of the Submit User ID can be defined. See also the field Submit Security User Type in the section <i>Monitor Defaults</i> in the <i>Administration</i> documentation. If this field is left blank, then the default user ID is inserted at job activation. See also the default setting User ID Definition (<i>Default Setting (1)</i> , <i>Administration</i> documentation), and the sections Operating System User IDs and Default User ID Determination .
	Account Number	Account number to be used for the BS2000 Submit User ID . If you leave this field blank, the account number is taken automatically from the BS2000 JOIN entry (only if the Entire Operations Monitor is under TSOS or under the same user ID as the job).

Operating System	Field	Description
		Symbol replacement is performed in this field if the activation escape character is used.
	BS2000 Job Class	Valid job class for job submission defined for BS2000. Symbol replacement is performed in this field if the activation escape character is used.
	Job Priority	If not empty, this job priority will be used during submission and will override a possible setting in the LOGON statement. The default value on the network level will be used for new job definitions.
	Run Priority	If not empty, this run priority will be used during submission and will override a possible setting in the LOGON statement. The default value on the network level will be used for new job definitions.
	Share SYSOUT	Make BS2000 SYSOUT file shareable. If you enter Y here, the internal temporary SYSOUT file can be accessed from other BS2000 user IDs.
	Submit Password	Password for job submission. (Required only for BS2000 password encryption where submit node Entire System Server version is less than Version 3.4.1).
	SYSOUT User ID	User ID under which internal SYSOUT files are created by Entire Operations. If you leave this field blank, the Submit User ID is used. See also the default setting User ID Definition described in <i>Default Setting (1)</i> in the <i>Administration</i> documentation.
	SYSOUT Cat ID	Enter the SYSOUT catalog ID. This is the catalog ID under which internal SYSOUT files are created by Entire Operations. This field is meaningful only if you specify a SYSOUT User ID different from the Submit User ID .
	Collect SYSLST	If you enter Y here, and if the job contains a SYSLST assignment (like /SYSFILE SYSLST=. . . or /ASSIGN-SYSLIST TO-FILE=. . .), the SYSLST output is appended to the SYSOUT collection by Entire Operations. The SYSLST output is then also available in the SYSOUT online display.
	MonJV	Name of the BS2000 Monitor Job Variable to be used at job submission. If you leave this field blank, Entire Operations generates an internal unique name if the product JOB VARIABLES is available.
	Password	Password for the defined BS2000 Monitor Job Variable. Only alphanumeric passwords are supported.
z/OS	Submit User ID	In z/OS, the Entire Operations Monitor starts jobs under this user ID. You can only define this user ID if you are logged on to the executing node with the same user ID.

Operating System	Field	Description
		<p>Default: If this field is empty, the user ID from the last Submit User ID change is taken.</p> <p>In the Monitor standard values/job start user type in the access control system, certain checks of the job start user ID can be defined. See also the field Submit Security User Type in the section <i>Monitor Defaults</i> in the <i>Administration</i> documentation.</p> <p>See also the default setting User ID Definition (<i>Default Setting (1)</i>, <i>Administration</i> documentation), and the sections <i>Operating System User IDs</i> and <i>Default User ID Determination</i>.</p>
UNIX and Windows	Submit User ID	<p>The Entire Operations Monitor will set the user ID of the shell script to this value. The user ID will be converted to lower case automatically.</p> <p>In the Monitor standard values/job start user type in the access control system, certain checks of the job start user ID can be defined. See also the field Submit Security User Type in the section <i>Monitor Defaults</i> in the <i>Administration</i> documentation.</p> <p>See also the default setting User ID Definition (<i>Default Setting (1)</i>, <i>Administration</i> documentation), and the sections <i>Operating System User IDs</i> and <i>Default User ID Determination</i>.</p>
	Submit Group	<p>UNIX: If this field is empty, the user's default UNIX group (from <code>/etc/passwd</code>) is used. Otherwise, this field must contain one of the UNIX groups, which is visible in the output of the UNIX groups command.</p> <p>Windows: You can specify the user's domain in this field.</p>
	SYSOUT Node	<p>Only if the SYSOUT is to be copied to BS2000:</p> <p>Node to which internal SYSOUT files are to be copied by Entire Operations. The SYSOUT node must be different from the execution node, and it must be a BS2000 node.</p> <p>See also <i>Defining the Handover of SYSOUT Files from UNIX/Windows to BS2000</i>.</p>
	SYSOUT Cat ID	<p>Only if the SYSOUT is to be copied to BS2000:</p> <p>Catalog ID under which internal SYSOUT files are copied by Entire Operations.</p> <p>See also <i>Defining the Handover of SYSOUT Files from UNIX/Windows to BS2000</i>.</p>
	SYSOUT User ID	<p>Only if the SYSOUT is to be copied to BS2000:</p> <p>User ID under which internal SYSOUT files are copied by Entire Operations.</p> <p>See also the default setting User ID Definition (<i>Default Setting (1)</i>, <i>Administration</i> documentation), and the sections <i>Operating System User IDs</i> and <i>Default User ID Determination</i>.</p>

Operating System	Field	Description	
	Command Line obligatory	Possible values:	
		Y	The existence of the symbol <i>CMDLINE-job</i> is obligatory. If it is missing or empty, the job will not be submitted. See <i>Predefined Symbols for Command Line Parameters</i> in the section <i>Symbol Table and Symbol Maintenance</i> .
		N	A command line is not obligatory for the job (default).

22

Using a Dummy Job

- Dummy Job Execution by the Monitor 210
- Permanent Dummy Jobs 210
- Temporary Dummy Jobs 210
- Excluding a Job from Actual Execution 211
- End-of-Job Actions after Execution as a Temporary Dummy Job 212
- Supervision of Running Jobs 213

The execution of a dummy job means that the job is running without job control and without its own action within Entire Operations. Dummy jobs can have an expected run time, which they will be waiting in the system. Dummy jobs will always terminate with the state `o.k.`.

Dummy Job Execution by the Monitor

You can directly submit and execute a dummy job by using a prerequisite checking routine. This increases the throughput of the Entire Operations Monitor and reduces overhead caused by changing job queues.

Permanent Dummy Jobs

Permanent dummy jobs are declared with the job type `DUM` (**Dummy Job**) in the master job definition.

Permanent dummy jobs will execute with the estimated elapsed time, which is defined for the job. This does not block any queues within the Entire Operations Monitor.

Temporary Dummy Jobs

In principle, you can execute any type of job as a temporary dummy job. An active job that fulfills any of the conditions described under *Dummy due to Condition* is always executed as a dummy for a particular run.

Temporary dummy jobs with the job type `JOB` (**Standard Job**) have a master JCL definition (see *Defining and Managing JCL for a Job*).

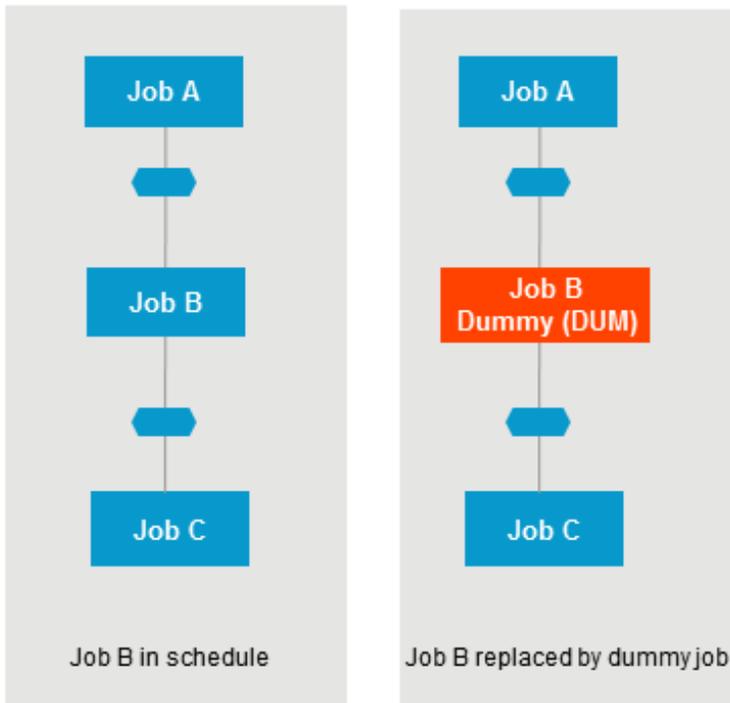
Temporary dummy jobs always have the execution time zero, regardless of which estimated elapsed time is defined for the job.

During its activation or during a prerequisite check, a job can be put into the state `temporary dummy job` for the following reasons:

Dummy due to Condition	Description
Definition	If the special type D (Execute as dummy) is set in the master job definition, the JCL definition can be maintained, but the execution as a dummy job can be forced.
Schedule	Depending upon the schedule of the network or upon a calendar, a job can be submitted as a temporary dummy job on particular days. For further information, see <i>Maintaining Schedule Dependencies for a Job</i> .
Prerequisite	It can be defined that the existence or non-existence of an active prerequisite results in execution as a temporary dummy job. For further details, see <i>Possible References for Input Conditions</i> .
Recovery Action	If a job is not to be submitted again in the case of a defined recovery action. For further details, see <i>Defining Recovery Action</i> in the section <i>End-of-Job Checking and Actions</i> .
JCL check	Execution of the job control in the mode <i>JCL check</i> . Thus, jobs (see <i>Fields: Job Activation</i>) or Job Networks (see <i>Fields: Network Activation</i>) are activated for the JCL check.
Empty JCL	If the loading process of the job control reveals that the JCL is empty, then the job will be submitted as a temporary dummy job. This state can result from the JCL generation by means of macro instructions, for instance.
Multiple Suffixes	If the activation is to take place by means of multiple suffixes: if the multiple symbol used does not contain any occurrences, then a dummy job will be activated as a placeholder in the network during activation.

Excluding a Job from Actual Execution

You can exclude a job from actual execution on certain schedule days without disturbing the job network structure. If a job is not to be activated, it can be converted into a dummy job just for this run. For example:



 condition

You can use the **Schedule Dependency** feature (see the section *Schedule Maintenance*) to define job activation or input condition usage to be checked only on certain days.

You can also define schedule dependencies for input conditions. For more information, see [Defining Schedule Dependencies for an Input Condition](#) in the section *Schedule Maintenance*.

End-of-Job Actions after Execution as a Temporary Dummy Job

For **temporary dummy jobs**, the End-of-Job actions are partially carried out as if the jobs were running in normal mode.

Several End-of-Job action definitions contain a field **Execute if temp. Dummy** (described in the section [End-of-Job Checking and Actions](#)) which can be used to execute the End-of-Job action according to this definition if the job was executed as a temporary dummy job.

This section covers the following topics:

- Carrying out the End-of-Job Actions
- SYSOUT Actions
- End-of-Job Action User Exit

Carrying out the End-of-Job Actions

The End-of-Job actions are executed as follows (prioritized in top-to-bottom order):

Conditions	Conditions are always set or deleted (even if the job is a temporary dummy job).
Dummy on account of Schedule	No further End-of-Job actions are executed.
Dummy for other Reasons	End-of-Job actions are executed.

SYSOUT Actions

Actions that refer to the SYSOUT file are not executed if no SYSOUT file exists. This case will apply if the job is executed as a temporary dummy job.

End-of-Job Action User Exit

End-of-Job action user exits will not be performed for temporary dummy jobs.

Supervision of Running Jobs

Running jobs are checked as to whether they have exceeded their predefined deadline time.

If this is the case, a message:

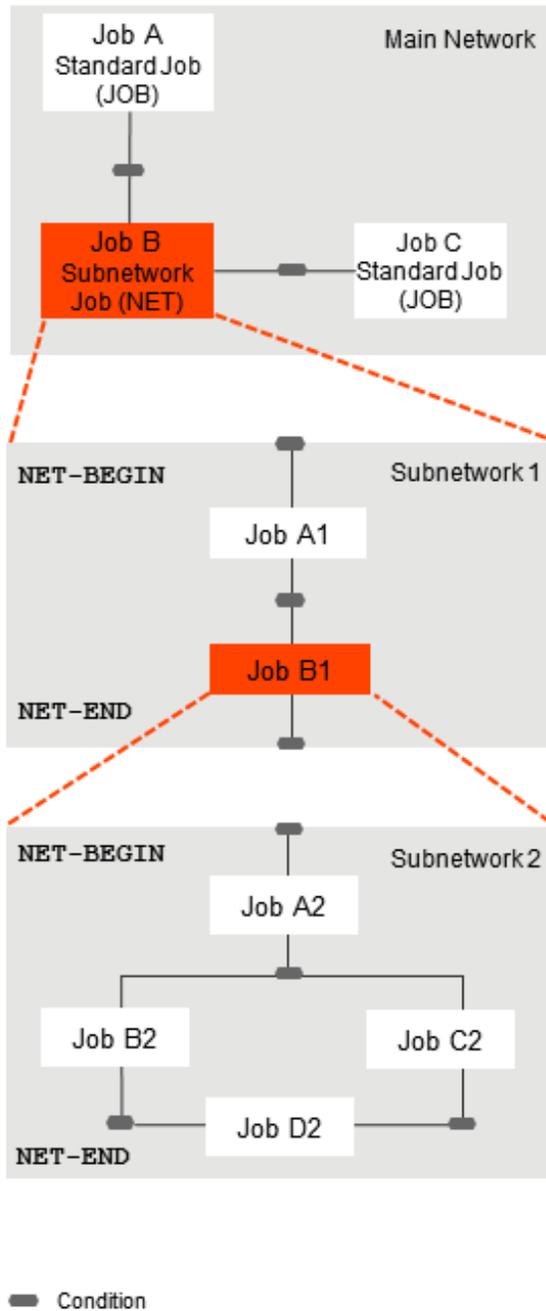
- is sent to all mailboxes linked to the network;
- is sent to Con-nect (if linked);
- is written to the log;
- appears on the **Active Jobs** or **All Active Jobs** screen.

23

Defining a Subnetwork

- Link to the Main Network 217
- Subnetwork Activation and Execution 218
- Restrictions for Subnetworks 219
- Adding or Modifying a Subnetwork Definition 220
- Listing Jobs of a Subnetwork 221

The job type NET (subnetwork) allows you to define a whole network as a job. The subnetwork must already be defined. It must have an input condition NET-BEGIN and an output condition NET-END as illustrated in the following example:



The same subnetwork can be defined in different jobs of the main network.

The subnetwork can be activated together with the calling network (or job of the type NET) or at the moment the job of the type NET is started.

Link to the Main Network

The following reserved conditions are used to link a subnetwork to a main network and control the connection between the calling NET job and the subnetwork called.

Reserved Condition	Description
NET-BEGIN	You must specify NET-BEGIN to start the subnetwork and run the first job of the subnetwork.
NET-END	You must specify NET-END to run the last job of the subnetwork and indicate the end of the subnetwork.
NET-END-NOTOK	<p>This condition is optional.</p> <p>If specified and fulfilled during subnetwork termination, the job is treated as ended not ok.</p> <p>If NET-END-NOTOK is specified in the subnetwork but NET-END is missing, the status of the NET job changes to Sub-NW executing - error indicated. In this case, the global message Job ended not ok is sent if the corresponding option is enabled in the Global Messages for Events settings (see the <i>Administration</i> documentation).</p> <p>If NET-END-NOTOK is reset in the subnetwork, the status of the NET job changes back to Subnetwork executing. In this case, the global message mentioned above is sent again.</p>
NET-END-OK	<p>This condition is optional.</p> <p>If specified at least once in your subnetwork and fulfilled during subnetwork termination, the NET job (that is, the whole subnetwork) is treated as ended ok.</p> <p>This has an impact on the release of resources, which are allocated on the network level. You can force the resource release by setting this condition. See also the deallocation mode set in the prerequisite resource definition.</p>

Notes on Subnetwork Usage

1. If the condition NET-END-NOTOK is found together with NET-END, the calling job is set to the status not ok.
2. Jobs of the type NET can use all kinds of prerequisite checks and can have all kinds of End-of-Job actions.
3. The scheduling parameters of a job of the type NET are passed on to the subnetwork. See the section *Scheduling a Job*.
4. The calling job of a subnetwork (job of type NET) should have an own symbol table definition.

5. You can also define specials for jobs of the type `NET` by pressing `PF6` (Spec) on the job definition screen.

Subnetwork Activation and Execution

A subnetwork is activated together with the main network. The subnetwork is assigned its own **run numbers**.

If the calling job of the type `NET` (subnetwork) becomes a temporary dummy job due to any reason, no subnetwork activation will be performed (if the subnetwork was not activated already).

Reserved symbols pass information to the jobs of the subnetwork about which job invoked the subnetwork. These symbols are described in *Predefined Symbols for Subnetworks and Recovery Jobs* in the section *Symbol Table and Symbol Maintenance*.

If the subnetwork has a standard symbol table, these values are entered as symbols in the active copy of this table.

This enables you to trace, even through several steps, from where the subnetwork was invoked.

If the job of the type `NET` can be activated (satisfies all input conditions), the condition `NET-BEGIN` of the subnetwork is set and the subnetwork starts to work: the `NET` job is then executing. It remains in this status until the subnetwork has set the condition `NET-END`. Only at this point are the End-of-Job checking and actions performed which are defined in the invoking network.

If an execution error occurs in the subnetwork, the execution of the invoking network is also blocked. If the subnetwork sets the condition `NET-END` before its actual termination (or at its very start), the invoking network continues to run, and a parallel asynchronous execution of the networks is possible. However, the user him/herself then becomes responsible for later synchronizing the subnetwork with the invoking network.

This section covers the following topics:

Time of Activation of a Subnetwork

Subnetworks can be activated at two different times:

- **At the time of the activation of the calling network**

The subnetwork is activated together with the calling network (i.e., with the job type `NET`). It will be available from this point in time with **run number** and active JCL. This is the default.

- **At the start time (submission) of the calling job of the type `NET`**

The subnetwork is only activated if the calling job of the type `NET` is really started. This can prevent a subnetwork from even being activated if the calling job is not executed at all later on.

Moreover, the effort for job activations and loading the JCL is thus compensated for, and the waiting-time for prerequisites within the subnetwork will be shorter. Note, however, that the active subnetwork including active JCLs will only exist from a later time onwards.

The subnetwork activation mode can be defined as follows:

- As a global default value (see **Subnetwork Activation Mode** in *Default Setting (2)* in the *Administration* documentation).
- In the definition of the calling job of the type NET (see **Subnet Activation Mode**).

Symbol Table Activation for Subnetworks

The order of symbol table activation for subnetworks is:

1. Set caller info into the active table (P-C-... symbols).
2. Activate symbol table(s).
3. Perform symbol modification exit.

The order of symbol table activation for subnetworks is the same as for main networks.

Restrictions for Subnetworks

This section covers the following topics:

- [Recursion Checking](#)
- [Maximum Nesting Level](#)

Recursion Checking

Subnetworks can in turn be invoked within subnetworks; however, a subnetwork cannot invoke itself, because this can cause an infinite recursion.

The definition of a network as its own subnetwork is prohibited. If this is attempted, the error message `Recursive Subnetwork Definition prohibited` is issued.

Maximum Nesting Level

The maximal supported nesting level for subnetworks is 5.

Adding or Modifying a Subnetwork Definition

➤ To define a subnetwork

- 1 Press PF8 (Net) in the **Job Definition window**.

A **Job: Subnetwork Definition** window similar to the example below opens:

```

+-----+
!                                     !
!               Job: Subnetwork Definition               !
!                                     !
! Owner SAGTEST      Network SAGNET      Job JOB-SUBNET  Run      !
!                                     !
! Description        ==> Subnetwork Job of Type NET      !
! JCL Node           ==> _____                      !
!                                     !
! Subnetwork Owner   ==> SAGTEST____ Use Subnetwork Time Frames ==> N !
! Subnetwork Name    ==> SAGNET2____ Subnet Activation Mode    ==> D !
! Subnetwork Version ==> _____ Append Mult. Suffix       ==> N !
! Subnetwork Run     ==>                                     !
!                                     !
! Subnetwork Input  Condition ==> NET-BEGIN              !
! Subnetwork Output Condition ==> NET-END                !
!                                     !
! Enter-PF1-----PF3-----PF5-----                  !
!      Help      End      Save                          !
+-----+
! Enter-PF1---PF2--PF3--PF4---PF5---PF6---PF7---PF8---PF9-----PF12- !
!      Help  Add  End  Edit  Save  Spec  Symb  Net  JCL      Menu  !
+-----+

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Add  End      Save      Up    Down      Menu
    
```

- 2 Enter the required values. The input fields provided in the window are described in *Fields: Job Type Specific Execution Features*.
- 3 When you are finished, press PF5.

Listing Jobs of a Subnetwork

If a master or active job has the type **NET**, and if the subnetwork is defined properly, you can invoke the master or active job list of the defined subnetwork directly.

If the subnetwork was not defined yet, the subnetwork definition screen will be invoked instead.

➤ To list jobs of a subnetwork

- 1 On a **Job Maintenance**, an **Active Jobs** or **All Active Jobs** screen, type Z (Subnet) in the line command input field next to a job of the type **NET**.

The name of the job owner and the subnetwork defined for the job are indicated in the **File or Library** column.

- 2 Press **ENTER**.

The jobs of the subnetwork defined for the selected **NET** job (here: **SAGNETSUB1**) are now listed on the **Job Maintenance** screen, as shown in the following example:

```

17-12-12          ***** Entire Operations *****          13:24:15
Job Maintenance   Owner SAGTEST   Network SAGNETSUB1 Version
-----
Cmd C R PU Job      Type Loc Description          File or Library      Member
*-----*
_      JOB-1      JOB NAT Where it all starts    SYSEORU              B60-M01
_  C1  SUBNETJOB2 NET      Subnetwork Job for Su SAGTEST/SAGNETSUB (D)

***** Bottom of Data *****
A Dep. B Browse C Copy D Del. E Edit G Pregen. I Input Cond. J JCL L Resources
M Mod. O EOJ Chk + Act P Prose R Activate S Sched.Parms U Ext.Log Z Subnet
Command => _____

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Add   End           Save           Up           Down           Menu

```



Note: If you leave the subnetwork job list with **PF3**, you will return to the calling network list.

24

Defining Parameters for an FTP Job

- Adding an FTP Parameter Definition 224
- JCL Generation for FTP Jobs 227

Jobs of the type **FTP** use JCL generated by Entire Operations, depending on the defined FTP (File Transfer Protocol) parameters.

Adding an FTP Parameter Definition

➤ To add or modify an FTP parameter definition

- 1 In the **Job Definition (Master) window** of the required job, enter **FTP** in the **Job Type** field and press **PF9** (JCL).

Or:

On the **Job Maintenance screen**, type **J** (JCL) in the line command input field next to the job of the type **FTP** whose definition you want to modify, and press **ENTER**.

A **Job Definition - FTP Attributes** screen similar to the example below appears:

```

24.10.08          ***** Entire Operations *****          09:12:22
                   Job Definition - FTP Attributes
                   ↵
                   ↵
Owner    ==> SN                      Run ==>                               ↵
Network  ==> FTP-002                 Execution Node ==> N0517 Linux          ↵
Job      ==> LIN-01                   Symbol Table ==> FTP-002             ↵
----- ↵
Remote ... ↵
Host     ==> $HOST_____ ↵
User ID  ==> $FUSER_____ Account ==> _____ ↵
Group    ==> _____ Password ==> def. Y ↵
Remote Directory ↵
$REMOVEDIR_____ ↵
File _____ ↵
test_____ ↵
File 2 (Target) ↵
----- ↵
Local Directory ↵
localdir_____ ↵
FTP Type ==> _ Function ==> reget___ File Type ==> A ↵
----- ↵
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      End      Save      Symb

```

- 2 Enter the required values.

The input fields are explained in [Fields: FTP Parameter Definition](#).

The special PF key function available is described in [Special PF Keys: FTP Parameter Definition](#).

- 3 When you are finished, press PF5 to save your entries.

This section covers the following topics:

- [Fields: FTP Parameter Definition](#)
- [Special PF Keys: FTP Parameter Definition](#)

Fields: FTP Parameter Definition

The fields on the [Job Definition - FTP Attributes screen](#) are described in the following table.

Field	Description
Remote Host	Name of the remote host to be used for the file transfer. Example: host1.company.net Symbols can be used.
Remote User ID	User ID for the FTP login on the remote host. Symbols can be used. Note for UNIX: If you use .netrc (on the execution node's host) to specify user IDs and passwords for remote hosts, you may leave this field empty.
Remote Account	Account for the FTP login on the remote host. Required for some platforms only. Symbols can be used.
Remote Group	Group or domain, respectively, for the FTP login on the remote host. Required for some platforms only. Symbols can be used.
Remote Password	Password for the FTP login on the remote host. Note for UNIX: If you use .netrc (on the execution node's host) to specify user IDs and passwords for remote hosts, you may leave this field empty.
def.	Y A password is defined
	N A password is not defined.
Remote Directory	Directory on the remote host, which contains (or is the target) of the file(s) to be transferred. Symbols can be used.
File	File(s) to be transferred. Symbols can be used.

Field	Description	
File 2 (Target)	(optional) File name for the target. Use this field only if the file is to be renamed on the target machine. Symbols can be used.	
Local Directory	Directory on the local host (execution node), which contains (or is the target) of the file(s) to be transferred. Symbols can be used.	
FTP Type	FTP type Valid input value: F (standard FTP)	
Function	File transfer function to be used. Examples: get, put, mget, mput	
File Type	File type. Valid file types are:	
	A	ASCII for text files.
	I	Binary for other files.

Special PF Keys: FTP Parameter Definition

The following special PF key is provided on the [Job Definition - FTP Attributes screen](#):

PF Key	Name	Function
PF7	Symb	Display the symbol table specified in the Symbol Table field . You can define or change the symbol table.

JCL Generation for FTP Jobs

Submit User ID

If no **Submit User ID** is defined for a z/OS FTP job, Entire Operations tries to use the network's default **Submit User ID**. If this fails, Entire Operations attempts to use the **execution node's** default **Submit User ID**.

Job Card z/OS

The z/OS job card generation can be done manually by defining the symbol `FTP-JOBC1-jobname`.

A job card continuation can be defined optionally in `FTP-JOBC2-jobname`.

If neither symbol is present, the z/OS job card will be generated as

```
//submit-userid JOB ...
```

"site" Commands

If the predefined symbols `FTP-SITE1-jobname` and `FTP-SITE2-jobname` are defined, one or two FTP "site" commands will be generated, appended by the contents of these symbols.

FTP Function

If the FTP function is `put`, and if the target file is to be renamed, a sequence of `put` (with the origin name) and `rename` is generated now.

FTP Parameters

The FTP parameter generation can be done manually by defining the symbol `FTP-PARM1-jobname`.

The default is:

For z/OS:	"-e -i -v -n"
For UNIX, Windows:	"-d -i -v -n"

See also [Predefined Symbols](#) in the section *Symbol Table and Symbol Maintenance*.

25

Defining and Managing JCL for a Job

▪ Job Control for Jobs under BS2000	230
▪ Job Control for Jobs under UNIX	230
▪ Job Control for Jobs under Windows	231
▪ Defining Master JCL for a Job	234
▪ Displaying Master JCL	238
▪ JCL Locations	239
▪ Defining Operating System Dependent JCL Specifications	243
▪ Handling JCL during Job Submission	246
▪ Symbol Replacement in JCL	250
▪ Importing JCL into a Natural Library	250
▪ Pregenerating Active JCL	253

A job needs job control language (JCL) instructions to perform a task. The required JCL is contained in the JCL member of a library/file of the operating system, or in a Natural object contained in a Natural library/system file.

This section provides information on defining a JCL for a master job and editing the JCL source referenced by a job.



Note: FTP jobs do not have a standard JCL definition. They have an [FTP parameter definition](#) instead.

Related Topics:

- [Editing Master JCL and Natural Sources](#)
- [Regenerating Active JCL](#) in the section *Active Job Networks*

Job Control for Jobs under BS2000

This section covers the following topics:

- [Naming Conventions for Work Files](#)
- [User Exit for Work File Names](#)

Naming Conventions for Work Files

The name generation for work files under BS2000 is described in the section *Naming Conventions for Work Files* of the *Installation and Setup* documentation.

User Exit for Work File Names

Names of BS2000 work files can also be generated via an exit (see [Generating SYSOUT File Names for BS2000](#) in the section *API Routines*).

Job Control for Jobs under UNIX

The environment variable \$EOR_WORK of Entire System Server/UNIX contains the name of the Entire Operations work directory. Within this directory, the work files are hierarchically stored.

Job Control for Jobs under Windows

This section covers the following topics:

- [Job Control Characteristics under Windows](#)
- [Components Required](#)
- [Executing Operating System Functions](#)
- [File Names](#)
- [SYSOUT Redirection](#)
- [Entire Operations Work Directory](#)
- [Environment Variables](#)
- [Job Control \(JCL\)](#)
- [Job Start and Job Control](#)
- [End-of-Job Check](#)
- [End-of-Job Actions](#)

Job Control Characteristics under Windows

The job control system Entire Operations also runs in a Windows environment, with the following characteristics:

- Operating system neutral modeling of job networks.
- Entire Operations' functionality and flexibility is also transparently available on Windows.
- Support of DOS batch files and executables (EXE).
- It is not necessary to directly enter Windows DOS commands.
- Can also run in mixed mainframe/Windows/UNIX environments.
- Job control on several Windows machines at the same time.

Components Required

■ Entire Operations Monitor

The monitor can run under the operating systems BS2000, z/OS, z/VSE, or UNIX, concurrently controlling jobs executed under BS2000, z/OS, z/VSE and UNIX.

■ Entire System Server - For Mainframes, UNIX and Windows

Required to access mainframe, UNIX or Windows operating systems.

On every machine to be controlled, an Entire System Server/Windows server has to be installed. This is installed as a Windows service and is administrated with the Windows services administration.

■ **Entire Net-work/EntireX Broker**

Used as the transport layer.

Executing Operating System Functions

For executing operating system functions, there is a server of type Entire System Server/Windows on each Windows node. This server runs as a background Windows process.

The Entire Operations Monitor and the Entire Operations online application use the following components for server communication:

- The System Automation Tools (SAT) communication layer
- The EntireX Broker for transmitting client/server requests
- Entire Net-work as the transport layer
- Up to 740 Windows nodes can be served concurrently

File Names

As there is no backslash (\) available on mainframes, Windows file names can be written with a slash (/) as an alternative as well if the character sequence +F+ directly precedes the file name. This applies to file names within JCL as well.

Example:

Original Windows	c:\jcl\script1.bat
Alternative representation	+F+c:/jcl/script1.bat

SYSOUT Redirection

From the Entire Operations Monitor, all jobs are started which redirect their output to a file. The SYSOUT files are stored in the Entire Operations work directory. If the job is repeated, the old SYSOUT file is renamed.

Entire Operations Work Directory

The Entire System Server/Windows environment variable `%EOR_WORK%` contains the name of the Entire Operations work directory. Within this directory, the work files are stored hierarchically.

The name generation for work files under BS2000 is described in *Naming Conventions for Work Files Created by Entire Operations, Windows* in the section *Installing Entire Operations on Mainframes* in the *Installation and Setup* documentation.

The name of the work directory for an active network is available in the predefined symbol `P-NADIR`. You may store application-specific work files there, provided there are no naming conflicts with files created by Entire Operations.

Work files created by Entire Operations or by the application are deleted by the Entire Operations Monitor on network or job deactivation.

Environment Variables

You can use Windows environment variables freely within file names. This corresponds to the behavior of BAT files. You can combine environment variables and variables from symbol tables.

Job Control (JCL)

The Windows JCL can be stored in any place. Among other places, it can be stored in Natural text objects or in mainframe files. Symbol replacement and JCL generation (jobs with JCL location `MAC`) are available.

Job Start and Job Control

The Entire Operations Monitor starts jobs based on conditions and on time. Accounting data is retrieved and stored. You can manually cancel the job from the online environment.

End-of-Job Check

Entire Operations adds some echo commands to the Windows job control. This allows control of certain messages in the `SYSOUT`:

- Starting and ending message with timestamp
- Elapsed job time

These messages are used to check whether a job did run completely. Additionally, accounting information is retrieved. For checking the job, strings can be searched for in `SYSOUT` and End-of-Job check routines can be used.

End-of-Job Actions

You can send messages (for example by e-mail) to other users from the Windows node. In the Windows node definition, a program for sending mail can be defined. It is required that this program can be started from the DOS command prompt. One example for such a program is the shareware wsendmail. All other forms of news broadcasting, such as the Entire Operations Mailbox, can still be used.

You can define printing of files and of SYSOUT lists as an End-of-Job action. For each Windows node, a Windows print command with a variable for the file name can be defined. User exits can help start further actions.

Defining Master JCL for a Job

> To define master JCL for a job

- 1 On the **Job Maintenance screen**, type J (JCL) in the line command input field next to the job for which you want to add or modify a JCL definition.

Or:

In the **Job Definition (Master)** window of the required job, press PF9 (JCL).

- 2 Press ENTER.

A **Job: JCL Definition (Master)** window similar to the example below opens:

```

17-03-09          ***** Entire Operations *****          11:55:22
Job Maintenance   Owner SAGTEST   Network SAGNET   Version
-----
+-----+
!
!                   Job: JCL Definition (Master)                   !
!
! Job Name          ==> TEST-JOB           Mod ==> NATQA5   17-03-09 11:54   !
! Description       ==> Job for JCL Test                                     !
! Job Type          ==> JOB                                                     !
! Execution Node    ==> 55522 MVS/ESA           Symbol Table ==> EXAM-TABLE !
!
! JCL Location      ==> NAT                   JCL Load Mode ==> A           !
! JCL Node          ==> 55522 MVS/ESA                                     !
! File/NatLib       ==> SYSEORU_____                                     !
! Member           ==> JCLTEST_ _____                                     !
! VolSer            ==> _____ (if not catlgd) Password ==>         !
!                                                           defined ==> no   ! *
! Enter-PF1-----PF3--PF4---PF5---PF6---PF7---PF8-----PF12-   !
!           Help      End  Edit  Save  Spec  Symb  Impo           Menu   !
+-----+
!           Help  Add  End  Edit  Save  Spec  Symb  Net  JCL      Menu   !
+-----+

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
           Help  Add  End           Save           Up    Down           Menu

```

- 3 Enter the required values. The input fields and special PF keys available are explained in *Fields: Master JCL Definition*.

After having entered all values, you can continue with one of the functions provided by the PF keys described in *Special PF Keys: Master JCL Definition*.

- 4 When you are finished, press PF5 (Save) to save your entries.

If you want to edit the defined JCL, proceed as described in *Editing Master JCL and Natural Sources*.

This section covers the following topics:

- [Fields: Master JCL Definition](#)

- [Special PF Keys: Master JCL Definition](#)

Fields: Master JCL Definition

The fields in the [Job: JCL Definition \(Master\) window](#) are described in the following table:

Field	Description			
JCL Location	<p>Type of JCL storage location.</p> <p>For valid input values, see JCL Locations.</p> <p>In a network definition, this field serves as a default for the entire network. See also the section .</p>			
JCL Node	<p>JCL for the job is read from this node.</p> <p>The job is started on the execution node. The default is the value specified in the job network definition (see Fields and Commands: Network Definition), which can be overwritten here.</p> <p>Note: In this field, you can also enter PMPA preceded by an escape character, for example, \$PMPA. See \$PMPA in the section <i>Symbol Table and Symbol Maintenance</i>.</p>			
JCL Load Mode	Possible values:			
	<table border="1"> <tbody> <tr> <td>A</td> <td>Load JCL at activation time (default).</td> </tr> <tr> <td>S</td> <td> <p>Load JCL at submission time. Use this only if prerequisites for JCL are not available at activation time.</p> <p>Note:</p> <ol style="list-style-type: none"> 1. JCL load errors will be detected very late. 2. Network execution may slow down, because of the time necessary for JCL load before submission. 3. JCL pregeneration also works with submission time JCL load. 4. If resources are defined, they will be allocated before a JCL load at submission time. If the JCL load fails, the allocated resources will be freed immediately. 5. JCL loading at submission time is mainly intended to allow the generation of macro JCL as late as possible. 6. For other types of JCL, we still recommend submission time symbol replacement, which causes less system overhead. </td> </tr> </tbody> </table>	A	Load JCL at activation time (default).	S
A	Load JCL at activation time (default).			
S	<p>Load JCL at submission time. Use this only if prerequisites for JCL are not available at activation time.</p> <p>Note:</p> <ol style="list-style-type: none"> 1. JCL load errors will be detected very late. 2. Network execution may slow down, because of the time necessary for JCL load before submission. 3. JCL pregeneration also works with submission time JCL load. 4. If resources are defined, they will be allocated before a JCL load at submission time. If the JCL load fails, the allocated resources will be freed immediately. 5. JCL loading at submission time is mainly intended to allow the generation of macro JCL as late as possible. 6. For other types of JCL, we still recommend submission time symbol replacement, which causes less system overhead. 			

Field	Description
File/NatLib	<p>Storage type of the JCL corresponding to the job type.</p> <p>For Natural programs and dynamic JCL (storage type NAT): the Natural library where the program or the variable JCL resides. The Natural library must reside in the Entire Operations FUSER system file and must not have the same name as the Entire Operations library.</p> <p>For z/VSE: library and sub-library associated by a period (.).</p> <p>For all others: the file name.</p> <p>You can list the file or library names by entering an asterisk (*) as a wildcard character. To list, for example, all file or library names beginning with XA, enter XA* in this field and press ENTER. To get the complete list, enter an asterisk (*) in the field and press ENTER. You can select the required library or file from this list. You can also delete a library or file from this list.</p> <p>For BS2000: The default pubset will be stripped from the file name prior to storing it. This allows easier migration to another default pubset.</p> <p>For UNIX: Be aware of upper and lower case. You may use environment variables and symbols within names.</p> <p>Symbols may be used. If a JCL file contains the activation escape character (see <i>Escape Characters</i>), a symbol replacement will be performed:</p> <ul style="list-style-type: none"> ■ at JCL editing (temporary, using the master symbol table). ■ at JCL loading (permanently, using the active symbol table). <p>For detailed information, see the section <i>Symbol Table and Symbol Maintenance</i>.</p>
Member	<p>Name of the Natural object or library/file member of the operating system that contains the JCL. When you add a new job definition, the name of the job is entered in this field by default.</p> <p>You can list the member names by entering an asterisk (*) as a wildcard character. To list, for example, all member names beginning with XA, enter XA* in this field and press ENTER. To get the complete list, just enter an asterisk (*) in the field and press ENTER. You can select the required member from this list. You can also delete a member from this list.</p> <p>New Members:</p> <p>You can also enter the name of a non-existing member if, for example, no JCL has yet been defined for the job. If the member does not exist, the message ** Member not in the Library * appears next to the field when you press ENTER or PF5, but Entire Operations accepts the member name assuming a member with this name will be created.</p> <p>Before activating a network and before starting a job, Entire Operations checks whether a member has been defined. If there is no member, an error message is issued.</p> <p>Symbols may be used. If a JCL file contains the activation escape character (see <i>Escape Characters</i>), a symbol replacement will be performed:</p> <ul style="list-style-type: none"> ■ at JCL editing (temporary, using the master symbol table).

Field	Description	
	<ul style="list-style-type: none"> at JCL loading (permanently, using the active symbol table). <p>This applies to all JCL locations which support members: see List of JCL Locations.</p> <p>For detailed information, see the section Symbol Table and Symbol Maintenance.</p> <p>Note for BS2000:</p> <p>JCL location PRC (Callable procedure): If no member is specified, the JCL file must be a BS2000 sequential file.</p>	
VolSer	Volume serial of the file.	
	Only required if the file has not been cataloged (z/OS only).	
Password	The file password if the file has been protected by a password.	
defined	yes	A file password is defined.
	no	A file password is not defined.

Special PF Keys: Master JCL Definition

You can perform the following functions using these PF keys in the [Job: JCL Definition \(Master\) window](#):

PF Key	Name	Function
PF4	Edit	Edit the defined JCL. See Editing Master JCL and Natural Sources .
PF6	Spec	Define special parameters for operating system dependent JCL specifications .
PF7	Symb	Display the symbol table specified in the Symbol Table field . You can define or change the symbol table.
PF8	Impo	Import the job control (only master definitions). Allows to convert the JCL from an operating system file format (for example PDS, TXT) to Natural source format or a macro language format.

Displaying Master JCL

➤ To display the master JCL source defined for a job

- Proceed as described in [To edit JCL of a standard operating system job or a Natural source](#) in the section *Using the Editor*.

JCL Locations

This section covers the following topics:

- [List of JCL Locations](#)
- [Restrictions for Job Types](#)
- [Location on another Node](#)
- [Replacing Symbols](#)
- [JCL Location PRC: BS2000 Procedure](#)
- [JCL Frames for BS2000 DO Procedures](#)

List of JCL Locations

Entire Operations offers a broad range of possible locations for its original (master) JCL.

The following table describes the physical locations you can specify for JCL in the **JCL Location** field of the [Job: JCL Definition \(Master\)](#) window.

The locations NAT (**Natural source**) and MAC (**Macro Natural source**) are available on all operation systems supported by Entire Operations. Other locations are available for individual operating systems only as indicated in the following table:

JCL Location/Input Option	Applies to Operating System	Description
<i>blank field</i> (No JCL required)	n/a	None (no JCL required).
BS2 SAM or ISAM	BS2000	SAM or ISAM file. See also BS2000 in <i>Fields: Operating System Specials for JCL</i> .
EXE Executable program	UNIX, Windows	Executable, for direct file execution. Not intended for shell scripts or BAT files. You can pass parameters to the executable file with the predefined CMDLINE - job symbol. See Predefined Symbols for Command Line Parameters in the section <i>Symbol Table and Symbol Maintenance</i> .
LIB Librarian	z/OS	Librarian.

JCL Location/Input Option	Applies to Operating System	Description
LMS LMS library	BS2000	LMS library. See also BS2000 in <i>Fields: Operating System Specials for JCL</i> .
MAC Macro Natural source	all	For macro JCL. A Natural source object will be used.
NAT Natural source	all	Natural source object. See also Restriction for Job Types .
PDS Partitioned data set	z/OS	Member of a partitioned data set. See also Restriction for Job Types .
PRC Callable procedure	BS2000	BS2000 procedure. See JCL Location PRC: BS2000 Procedure for more information.
RDR z/VSE reader queue	z/VSE	z/VSE reader queue for job type STC (Started Task) .
VSE z/VSE sublibrary	z/VSE	z/VSE member in z/VSE sublibrary.
TXT Text file	UNIX, Windows	Text file. See also UNIX and Windows in <i>Fields: Operating System Specials for JCL</i> .

Restrictions for Job Types

- Jobs of the type [STC \(Started Task\)](#) must have the location PDS on z/OS.
- Jobs of the type [NAT \(Natural Program\)](#) must have the location NAT.
- Jobs of the type [DUM \(Dummy Job\)](#) must not have any JCL location.
- Jobs of the type [FTP \(File Transfer Job\)](#) do not require a JCL location. Entire Operations generates the FTP JCL itself, depending on the FTP parameters defined on the [Job Definition - FTP Attributes screen](#).

During definition, it is checked whether the defined JCL exists at all. If the JCL cannot be found, a warning appears in the Job JCL Definition (Master) window.

Location on another Node

The JCL can be saved on another machine, not identical with the executing machine and even on another operating system, not identical with the executing operating system.

Replacing Symbols

In all JCL locations, you can easily replace symbols as described in [Symbol Replacement](#) in the section *Symbol Table and Symbol Maintenance*.

JCL Location PRC: BS2000 Procedure

If this location has been defined, Entire Operations generates a BS2000 ENTER job in the active JCL calling this procedure.

The following rules are applied:

- If no JCL member is specified, the JCL file must be a BS2000 sequential file. If a JCL member is specified, the JCL file must be a BS2000 LMS library.
- The BS2000 job name adopts the Entire Operations job name and if it is longer than 8 characters, it cuts the name to 8 characters.
- SDF statements (`CALL-PROC`) are used.
- The symbol table for which the job has been defined must contain all parameters of the BS2000 procedure. The symbol values are used for calling the BS2000 procedure.
- Positional and keyword parameters are supported.
- Numbers and names of the parameters are automatically retrieved from the procedure (`PROC / BEGIN-PROC` statement).
- The generated `CALL-PROCEDURE` statement has the parameter `LOGGING=*YES`.
- BS2000 job submissions will be performed with the generated line `/MODIFY-SDF-OPTIONS CONTINUATION=*NEW-MODE`.

JCL Frames for BS2000 DO Procedures

For BS2000 procedures, a user-defined JCL frame can be used. It must be saved under the name `PRCFRAME` in the library `SYSEOR`. If it is not found, Entire Operations generates a standard frame.

Special Commands

Command	Description
#ESC - FRAME <c>	<p>Defines the escape character the JCL frame uses to replace symbols.</p> <p>This command must be at the top and is compulsory.</p>
#CALL - PROC	<p>The procedure call (/CALL - PROC) is generated here.</p> <p>Adding User-Defined Parameters to the /CALL-PROC Statement</p> <p>It is possible to add user-defined parameters to the CALL - PROC statement in the PRCFRAME text object. The following applies:</p> <ul style="list-style-type: none"> ■ The #CALL - PROC line must look like #CALL - PROC -. The minus sign indicates a continuation. In the generated JCL, it will be placed in column 72. ■ The user-defined continuation(s) follow the #CALL - PROC line. They must begin with the following characters: / , . <p>Example:</p> <pre>#CALL - PROC - / , LOGGING=YES</pre>

Example

```
#ESC - FRAME $
/.SN$P-RUN LOGON SN,1
/REMARK === PRC FRAME EXAMPLE
/REMARK $AAAA YYYYYYYYYYYYYY
#CALL - PROC
/STA L
$BBBB
/LOGOFF
```



Notes:

1. All symbols used in the JCL frame must be contained in the active symbol table of the job. The active symbol table must continue to contain all symbols for calling the procedure.
2. If a symbol does not exist, job activation is discontinued.

Defining Operating System Dependent JCL Specifications

Some operating systems or JCL locations may require some additional definitions.

➤ **To define JCL parameters specific to an operating system**

- In the **Job: JCL Definition (Master) window**, press PF6 (Spec).

Depending on the operating system, a **Specials** window opens similar to the example of BS2000 below:

```

1 +-----+
0 !                                     !
!                                     !
!               Master Job Definition !
!               BS2000 Specials, JCL  !
!                                     !
!               Owner ==> SN           !
!               Network ==> BS2-EX-2   !
!               Job ==> JOB-1-NOP      !
!                                     !
!               Job Type ==> JOB Loc ==> BS2 !
!               JCL Node ==> 194 BS2000 !
!                                     !
!               Default User ID ==> NOP_____ !
!               JCL User ID ==> NOP_____ !
!                                     !
!               LMS Member for         !
!               ==> _____          !
!               LMS Member Version ==> _____ Type ==> _____ !
!                                     !
! Enter-PF1-----PF3-----PF5-----PF12-- !
!               Help       End       Save       Menu !
+-----+
+-----+
Command => _____
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Add  End       Save       Up    Down       Menu

```

(If no window for operating system specific definitions is available, the message Function not available for ... occurs instead.)

The fields contained in the window are explained in [Fields: Operating System Specials for JCL](#).

This section covers the following topics:

- [Fields: Operating System Specials for JCL](#)

Fields: Operating System Specials for JCL

The operating system specific input fields provided in the [Specials window](#) for JCL are described in the following table.

Operating System	Field	Description
BS2000	Default User ID	<p>Valid logon user ID defined for BS2000. This ID has no meaning for other operating systems. Unqualified file and job variable names in this job definition are prefixed with this BS2000 user ID.</p> <p>Note: The user ID TSOS can be defined only if the user defining the ID is working under TSOS.</p> <p>See also the default setting User ID Definition described in <i>Default Setting (1)</i> in the <i>Administration</i> documentation.</p>
	JCL User ID	<p>If this field is not empty, the JCL is loaded under the rights of this BS2000 user ID and may be overridden by specific definitions. TSOS may only be defined, if the defining user is working under TSOS.</p> <p>Default: If this field is left blank, the user ID from the fully qualified JCL file name is used as JCL User ID.</p> <p>See also the default setting of User ID Definition (<i>Default Setting (1), Administration</i> documentation), and the sections Operating System User IDs and Default User ID Determination.</p>
	LMS Member for	<p>Applies to the JCL locations LMS (LMS library) and PRC (Callable procedure).</p> <p>This field can be used specially for long member names of up to 64 characters.</p> <p>If the JCL location is LMS and you leave this field blank, the short member field is used.</p> <p>JCL location PRC: If no member is specified, the JCL file must be a BS2000 sequential file.</p>
	LMS Member Version	<p>Applies to the JCL locations LMS (LMS library) and PRC (Callable procedure).</p> <p>Version of the specified LMS member.</p> <p>This must be exactly the same as given in LMS itself. Leading zeroes must be specified in the same way.</p>
	Type	<p>Applies to the JCL locations LMS (LMS library) and PRC (Callable procedure).</p> <p>LMS member type.</p> <p>Possible values:</p>

Operating System	Field	Description
		S: Source program
		J: Procedure or ENTER job
		D: Document
		X: Data of any format
z/OS	JCL User ID	JCL in z/OS will be loaded under this user ID by the Entire Operations Monitor. You may define this field only if you are logged on to the JCL node with the same user ID. Default: If this field is left blank, the user ID of the last modification is used as JCL User ID . See also the default setting User ID Definition (<i>Default Setting (1), Administration</i> documentation), and the sections Operating System User IDs and Default User ID Determination .
	VolSer	Volume serial number of the file where the JCL resides. Only required if the file has not been cataloged.
z/VSE	Library	Name of the z/VSE library within the selected file.
	Sublib	Name of the z/VSE sublibrary within the selected file. You can use an asterisk (*) as a wildcard to open a selection window.
	Member	Name of the member where the JCL for the job resides. You can use an asterisk (*) as a wildcard to open a selection window.
	Member Type	Member type of the z/VSE library.
	Volser	Volume serial number of the file where the JCL resides. This is a required parameter for z/VSE.
	VSAM Catalog	Name of the VSAM catalog for the selected file.
UNIX and Windows	JCL User ID	The Entire Operations Monitor will load JCL with the access rights of this user ID. See also the default setting User ID Definition (<i>Default Setting (1), Administration</i> documentation), and the sections Operating System User IDs and Default User ID Determination .
	JCL Group	UNIX group or Windows domain (optional) . UNIX: If this field is blank, the user's default UNIX group (from <code>/etc/passwd</code>) is used. Otherwise, this field must contain one of the UNIX groups, which is visible in the output of the UNIX <code>groups</code> command. Windows: The Windows domain to be used for the logon.

Operating System	Field	Description
	UNIX Group (optional)	(Not relevant for Windows) If this field is blank, the user's default UNIX group (from <code>/etc/passwd</code>) is used. Otherwise, this field must contain one of the UNIX groups, which is visible in the output of the UNIX <code>groups</code> command.

Handling JCL during Job Submission

This section covers the following topics:

- [JCL Modifications during Submission](#)
- [Notes about the JCL Header Generation](#)
- [Trigraph Encoding for JCL Submission on UNIX and Windows](#)

JCL Modifications during Submission

Jobs defined and scheduled in Entire Operations are submitted automatically under the control of the Entire Operations Monitor. During the submission process, the submitted JCL can be handled in any of the following ways:

- Completion or modification of the job JCL according to the Entire Operations defaults;
- Checking of all submitted JCL by a global user exit (to be defined in the Entire Operations defaults);
- Insertion of header information as comments into the submitted JCL. This is always performed. The header information can be viewed in the job `SYSOUT` as shown in the following example:

```

Job SNNPEX (1355) Type SM File 2----- Columns 001 072
====>                                     SCROLL====> CSR
***** ***** top of data *****
00001      1 //NOPE01J1 JOB ,SN,CLASS=G,MSGCLASS=X,MSGLEVEL=(1,1),
00002          // TYPRUN=SCAN
00003          *** $ACFJ219 ACF2 ACTIVE DAEF
00004          *** =====
00005          ***              S O F T W A R E   A G
00006          ***              E N T I R E   O P E R A T I O N S
00007          ***
00008          *** Owner:      SN          Run:          328
00009          *** Network:   SN-2       Symbol Table:  EXAM-ST1
00010          *** Job:      CHECK-TIME  Escape Act:   Sub: $
00011          ***              Submit Userid: SN
00012          ***
00013          *** 29.10.08 15:15 activated          by SN
00014          *** 29.10.08 15:15 created/modified by SN
00015          *** 29.10.08 15:15 submitted
00016          *** =====
00017          *** Symbols replaced at Activation Time:
00018          ***

```



Note: In BS2000, the LOGON statement is checked. If nothing else was specified at job definition, the LOGON parameters, account-number, job-class, monitoring job variable (with password, if specified) are taken from here. Job priority, run priority and CPU time limit can also be given via LOGON statement.

- Insertion of information about all replaced symbols and their current values if any symbols were replaced. For example:

```

Job SNNOPLEX (1355) Type SM File 2----- Columns 001 072
====>                                     SCROLL====> CSR
00015      *** 29.10.08 15:15 submitted
00016      *** =====
00017      *** Symbols replaced at Activation Time:
00018      ***
00019      *** Symbol   : UID
00020      *** Owner   : SN   Symbol Table: EXAM-ST1
00021      *** Modif.  : SN   at 22-06-07 12:31
00022      *** Value   : SN
00023      *** Symbol   : CLASS
00024      *** Owner   : SN   Symbol Table: EXAM-ST1
00025      *** Modif.  : SN   at 22-06-07 12:32
00026      *** Value   : G
00027      *** Symbol   : MSGCLASS
00028      *** Owner   : SN   Symbol Table: EXAM-ST1
00029      *** Modif.  : SN   at 22-06-07 12:32
00030      *** Value   : X
00031      *** Symbol   : P-EXECUTION-NODE
00032      *** Value   : 146
00033      *** Symbol   : JOBLIB
    
```

- Replacement of symbols in the JCL with their current values at submission time.

■ **For z/OS only:**

If a submission time symbol replacement error occurs, a JCL error will be forced to prevent from an execution of the job. Lines similar to the following will force a JCL error:

```
// ##### Entire Operations Symbol Replacement Error #####
```

The job will remain in an error status such as follows:

```
JobId 51058 - Symbol Replacement Error
```

End-of-Job processing will not be reached, because this is treated as a submission error.

See also [Working with Entire System Server Nodes](#) in the section *Operating System User ID*.

Notes about the JCL Header Generation

1. The **Submit User ID** is always shown.
2. The created/modified user ID is modified only if the job definition or JCL was modified. Activation or re-activation does not count as a modification.

Trigraph Encoding for JCL Submission on UNIX and Windows

Trigraph encoding is used to avoid translation failures during ASCII/EBCDIC text conversion. An ASCII character for which no equivalent EBCDIC character exists is then substituted by a trigraph and the text can be encoded successfully.

Entire Operations uses trigraphs when generating JCL for UNIX and Windows. For Entire System Server execution nodes on UNIX and Windows, trigraph encoding is switched on by default using the question mark (?) as the default escape character.

You can switch on and off trigraph encoding in the JCL with the following command:

#EOR-TRIG=YES	Enables trigraph encoding.
#EOR-TRIG=NO	Disables trigraph encoding.

The command can be placed at the beginning of a line or within a comment line.

Example

```
...
# #EOR-TRIG=NO
ls example.???
# #EOR-TRIG=YES
if ??( a == b ??) then
...
```

Supported UNIX and Windows Trigraphs

Entire System Server execution nodes on UNIX and Windows support the trigraphs listed in the following table. A trigraph always begins with two escape characters; the table below assumes the question mark (?) to be the default escape character.

ASCII	Trigraph	Remarks
[??(
\	??/	
]	??)	
^	??'	
{	??<	
	??_	
}	??>	
~	??-	
@	??%	
`	??;	
!	??:	

ASCII	Trigraph	Remarks
\f	??+	Form feed
\t	??&	Tab

Symbol Replacement in JCL

Symbols for replacement can be defined for all storage locations of master JCL.

Escape characters for symbols can be assigned system-wide, and they can also be defined individually for each job. Defaults for escape characters can be defined for each operating system.

Escape characters for symbol replacement must not conflict with other characters used in the JCL.

For detailed information, see the following sections in *Symbol Table and Symbol Maintenance*:

- [Defining Symbols and Symbol Values](#)
- [Symbol Replacement](#) and [Defining Escape Characters](#)

Importing JCL into a Natural Library

Entire Operations can read JCL from various origins. A facility is provided for copying JCL from other locations into a Natural library. If the required JCL should remain in its original location, the function described in this section must not be used.

Reasons for Importing JCL into a Natural Library

Importing JCL to a Natural library may be required for one of two reasons:

- Your job definition specifies a JOB-type job with JCL location NAT and the required JCL is in any operating system file. This function copies the JCL into a Natural library unchanged;
- You wish to migrate standard JCL to Entire Operations' dynamic JCL format, when you define a MAC-type job and JCL location NAT. The parameter section for dynamic JCL generation is automatically inserted at the head of the otherwise unchanged JCL. The Natural source thus created can be modified to take full advantage of dynamic JCL such as parameter substitution, dynamic code generation, etc.



Note: The editor command `MACRO` is always required to compile dynamic JCL into internal code. It must be used after the import to make the macro JCL executable. The macro JCL can be tested with the editor command `TEST`. See also [Special Editor Commands for Macro Sources](#).

This section covers the following topics:

- Using the Import Function

Using the Import Function

➤ To import JCL into a Natural source object

- 1 In the **Job: JCL Definition (Master) window**, press PF8 (Impo).

A **JCL Import from File to Natural Source** window similar to the example below opens:

```

17-03-09          ***** Entire Operations *****          11:55:22
Job Maintenance   Owner SAGTEST   Network SAGNET   Version
-----
+-----+
!                                     !
! +-----+                           !
!!                                     !
!!          JCL Import from File to Natural Source          !
!!                                     !
!! From                                     !
!! Location ==> NAT                               Node ==> 55522 MVS/ESA !
!! File ==> _____                               !
!! VSE Lib ==> _____ Sublib ==> _____ VSAM Cat ==> _____ !
!! Member ==> _____ Type ==> _____ Volser ==> _____ !
!! LMS Member                                     Password ==> _____ !
!! ==> _____                               !
!! Member Version ==> _____                               !
!! To                                     ! *
!! Library ==> SYSEORU                               !
!! Member ==> JCLTEST                               Overwrite (y/n) ==> N !
+ !                                     !
!! Enter Continue   PF3 End                               !
+ +-----+
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Add   End       Save       Up    Down                               Menu

```



Note: The **JCL Location** in the **Job: JCL Definition (Master) window** must be set to NAT.

- 2 Enter the required values:
 - In the fields under the heading **From**, enter the origin **File** and **Member** names. Enter a **Volser** name only if the file is not cataloged. You can use an asterisk (*) as a wildcard to open a selection window with file names.
 - In the fields under the heading **To**, enter the target Natural **Library** and **Member** (object) name.

Enter Y (Yes) or N (No) next to the **Overwrite (y/n)** option to determine whether or not an existing object is replaced by the imported source.

- When you are finished, press ENTER.

If the target Natural object does not already exist, the function is performed.

If the target object already exists and you entered Y for **Overwrite (y/n)**, the existing object is overwritten.

If the target object already exists and you entered N for **Overwrite (y/n)**, the object is not overwritten and an Overwrite Option required for existing JCL message occurs instead.

For a summary of all fields available in the window, see also [Fields: JCL Import from File to Natural Source](#).

Fields: JCL Import from File to Natural Source

The input fields of the **JCL Import from File to Natural Source** window are described in the following table:

Field	Description
From ...	
Location	JCL location (see JCL Locations).
Node	Node of the file containing the JCL to be imported.
File	Name of the file containing the JCL to be imported.
VSE Lib	Name of the z/VSE library within the selected file.
Sublib	Name of the z/VSE sublibrary within the selected file. You can use an asterisk (*) as a wildcard to open a selection window.
VSAM Cat	Name of the VSAM catalog for the selected file.
Member	Name of the member where the JCL for the job resides. You can use an asterisk (*) as a wildcard to open a selection window.
Type	BS2000: LMS member type. Possible values: S, J, D or X. z/VSE: Member type of the z/VSE library.
Volser	Enter the volume serial number of the file where the JCL resides. This is a required parameter for z/VSE.
Password (optional)	Password if the file is password-protected. BS2000: Only the first 4 bytes are used as Read password (alphanumeric only).

Field	Description
LMS Member	(BS2000 only) This field can be used specially for long member names. If the JCL location is LMS and you leave this field blank, the short member field is used.
Member Version	(BS2000 only) Enter the version of the specified LMS member. This must be exactly the same as given in LMS itself. Leading zeros must be specified in the same way.
To ...	
Library	Enter the name of the Natural library containing the object in which the imported JCL is to be stored.
Member	Enter the name of the Natural object in which the imported JCL is to be stored.
Overwrite (y/n)	Enter Y (Yes) if you want existing JCL in the specified Natural object to be overwritten. Default is N (No): an existing object source is not overwritten.

Pregenerating Active JCL

For performance reasons, it might be necessary to pregenerate active JCL to be used for job submission. This function is conceived mainly for macro jobs, whose dynamic JCL generation can become time-consuming, because macro language must always be executed. JCL can also be pre-generated for all other job types.



Note: In the case of a **resubmission of the active job**, the pregenerated JCL is not reloaded if the active JCL of the job has been edited in the meantime.

- [Pregenerating, Editing and Removing Active JCL](#)

Related Topic:

- [Regenerating Active JCL](#) in the section *Active Job Networks*

Pregenerating, Editing and Removing Active JCL

➤ To pregenerate active JCL

- 1 Type G in the line command field of the selected job in the **Job Maintenance screen**.
- 2 Press ENTER.

The following window opens:

```

24.10.08          ***** Entire Operations *****          09:20:08
+-----+-----+ R-169066
!
!           Active JCL Pregeneration          ! Member
!
! Owner SN      Network R-169066   Job J-1      ! JC117573
!
! Due to performance reasons, it is possible to pregenerate
! active JCL, which will be used for the job submission.
!
! Warning:
! Please use this function always after
! - you modified the definition of the JCL location,
! - you edited the master JCL,
! - you modified the linked Symbol Table.
!
! Please proceed with one of the following PF Keys:
! PF3      End
! PF4      Edit pre-generated JCL
! PF5      Pre-generate active JCL
! PF6      Remove pre-generated active JCL
E !
+-----+-----+
! 1--PF12---
! Menu

```

3 Press PF5 to pregenerate active JCL.

If no generation errors occur, the message `Function performed` indicates that active JCL has been pregenerated for the job.

4 Press PF4 if you want to open the Entire Operations editor and change the pregenerated JCL.

5 Press PF6 if you want to remove the pregenerated JCL.

The message `Object deleted` indicates that the pregenerated JCL has been removed.

6 Press PF3 to return to the [Job Maintenance screen](#).

26

Editing Master JCL and Natural Sources

- General Editing Considerations 256
- UNIX and Windows: Handling of Tab Characters (H'09') within JCL Lines 257
- Usage of Text Objects in JCL 257
- Using the Editor 257
- Editing Macro Sources for Dynamic JCL Generation 261
- Locking of Natural Sources 270

JCL and Natural source objects can be edited with the Software AG Editor.

By editing the master JCL, you change the master definition that affects each future executed active job.

Related Topics:

- [Editing Active JCL](#) in the section *Active Job Networks*
- [Editor](#) in the section *System Overview*

General Editing Considerations

Automatic Logon to the Operating System Server

Before JCL is edited, Entire Operations checks whether the user may access the JCL file. If it is an operating system file, then an automatic logon with the defined JCL user ID ([BS2000](#) and [UNIX and Windows](#)) will be carried out, if required. If this automatic logon does not function (e.g., if a password is required), then this will be displayed to the user. In this case, he must first logon to the corresponding operating system server.

Logon to a Node

While editing the master JCL, an implicit logon is triggered to establish the connection to the required node. The logon is not prompted if you have already logged on to this node within this session.

For further information, see the [logon function](#) described in the section *Using Entire Operations*.

z/VSE: Editing JCL from z/VSE/Power RDR

You can edit a unique member with `DISP=L` only. After issuing the editor `SAVE` command, the edited member is submitted to the RDR queue. The old member is deleted. If you create a new member, a default `JOB` statement is created.

Windows: File Names

File names in Windows can be written in an alternative notation if the backslash character (`\`) is not available, like in mainframe environments.

UNIX and Windows: Text Files

If a file to be edited or loaded as JCL is not a text file, the error message `E0R0325 - No Text File` is returned. For JCL loading, this is treated as a permanent error.

UNIX and Windows: Handling of Tab Characters (H'09') within JCL Lines

- If a master or active JCL is edited, each tab character is converted to a single blank before the editing. This prevents strange effects during editing. The JCL is written back without tab characters in any case.
- Extended logging, JCL logging: for the extended logging of JCL tab characters are also converted to blanks.
- If Entire Operations is running on a mainframe, the middleware translation tables convert tab characters to blanks anyway. This means that each UNIX and Windows JCL is being submitted without tab characters.
- If Entire Operations is running on UNIX or Windows, tab characters remain in master and active JCL, as long as the master or active JCL is not edited from within Entire Operations.

Usage of Text Objects in JCL

Entire Operations allows the inclusion of text objects into JCL. The text objects can have their own local parameters. Text objects can invoke other text objects, i.e., nesting is possible.

Text object inclusion can be used not only for macro jobs with **JCL location** MAC (macro Natural source), but also for other JCL locations.

For information about including a text object, see *Inserting Text Objects into JCL*.

Using the Editor

This section describes how to edit JCL and Natural sources including user exits.

➤ To edit JCL of a standard operating system job or a Natural source

- 1 Type E in the line command field of the selected job in the **Job Maintenance screen**. Press ENTER.

Or:

In the **Job Definition (Master) window**, press PF4 (Edit).

The Entire Operations editor screen appears showing the first page of the JCL or Natural source as shown in the following examples.

(If the selected job has no JCL or Natural source object defined, a blank editor screen appears.)

- Example of JCL for the **location PDS** (partitioned data set):

```

Edit NOP.EXAMPLE.JCL(NOPE02J3)----- Columns 001 072
====>                                     Scroll====> CSR
***** ***** top of data *****
00001 //NOPE02J1 JOB @UID,CLASS=@CLASS,MSGCLASS=@MSGCLASS,MSGLEVEL=(1,1)  ←
00002 //* ***** ←
00003 //* IF STEP02 IS NOT EXECUTED ←
00004 //* THEN CONDITION 'E02-J01-OK' ==> JOB 'E02-J02' WILL BE EXECUTED ←
00005 //* ELSE CONDITION 'E02-J01-NOK' ==> JOB 'E02-J03' WILL BE EXECUTED ←
00006 //* END-IF ←
00007 //* ***** ←
00008 //JOB LIB DD DSN=@JOB LIB,DISP=SHR ←
00009 //STEP01 EXEC PGM=NOPCONTI ←
00010 //*STEP02 EXEC PGM=NOPCONTI,PARM='U202' ←
00011 // ←
***** ***** bottom of data *****
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      End  Quit  Rfind Rchan Up    Down  Symbo Left  Right Curso
    
```

The example above shows the JCL member NOPE02J3 contained in the data set NOP.EXAMPLE.JCL.

- Example of JCL for the job location MAC (macro Natural program):

```

EDITNAT:SYSE0RU(E51-M01)->Subprogram->Struct->Esc:'@'----- Columns 001 072
====>                                         Scroll====> CSR
***** ***** top of data *****
==msg -caution- profile changed to "caps off" (from "caps on") because data ←
==msg contains lower case characters.                                         ←

00010 @ DEFINE DATA PARAMETER USING NOPXPL-A                                  ←
00020 @ LOCAL      /* LOCAL VARIABLES START HERE                             ←
00030 @ 1 #JOBNAME  (A10)                                                    ←
00040 @ 1 #DATE    (D)                                                       ←
00050 @ 1 #WEEKDAY (A10)                                                    ←
00060 @ END-DEFINE                                                            ←
00070 @ * -----                                                            ←
00080 @ MOVE P-JOB TO #JOBNAME                                               ←
00090 @ EXAMINE #JOBNAME FOR '-' DELETE                                       ←
00100 @ MOVE P-ACTIVATION-TIME TO #DATE                                       ←
00110 @ MOVE EDITED #DATE (EM=N(10)) TO #WEEKDAY                             ←
00120 //EORE51J1 JOB  ,@P-OWNER,MSGCLASS=@MSGCLASS,CLASS=@CLASS             ←
00130 //*                                                                      ←
00140 //STEP01 EXEC PGM=NOPCONTI,PARM='C=0000,T=00:00:10'                   ←
00150 //STEPLIB DD DISP=SHR,DSN=@STEPLIB                                       ←
00160 @ IF #WEEKDAY = 'Wednesday'                                             ←
00170 //* -- THIS STEP IS EXECUTED ON WEDNESDAYS ONLY --                     ←
00180 //WEDNESDY EXEC PGM=NOPCONTI,PARM='C=0004,T=00:10:00'                 ←
00190 //STEPLIB DD DISP=SHR,DSN=@STEPLIB                                       ←
00200 @ END-IF                                                                ←
00210 //*                                                                      ←
***** ***** bottom of data *****

```

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

The example above shows the macro Natural source E51-M01 contained in the Natural library SYSE0RU.

- 2 Write or edit the JCL or Natural source as required using Entire Operations editor commands and PF keys.

You can also write or edit JCL or a Natural source outside Entire Operations by using another editor available at your installation.

Use the appropriate PF keys or editor commands to navigate through the source and edit the text. See [Special PF Keys: Editor Screen](#) and [Editing Macro Sources for Dynamic JCL Generation](#).

The Entire Operations online help facility provides brief information on the main Entire Operations editor commands and features. The Software AG Editor is described in detail in the *Editors* documentation of Natural for Mainframes.

This section covers the following topics:

- [Editor SAVE Command](#)
- [Special PF Keys: Editor Screen](#)

Editor SAVE Command

If you have edited JCL of the type [PDS](#) or [NAT](#), you can save the JCL to another member of the same file or library by using the syntax:

```
SAVE member
```

For example: SAVE MEMBER2

Special PF Keys: Editor Screen

PF Key	Name	Description
PF4	Quit	Leave the source unchanged and exit the editor.
PF5	Rfind	Place the cursor on the next occurrence of the search string specified with the FIND editor command (for example: FIND <i>search-string</i>).
PF6	Rchan	Replace the next occurrence of the text string specified with the CHANGE editor command (for example: CHANGE <i>search-string new string</i>).
PF7	Up	Scroll up in the source.
PF8	Down	Scroll down in the source.

PF Key	Name	Description
PF9	Symbo	Open the Usable Symbol Tables window to select a symbol table for browsing or modification. See also <i>Listing Usable Symbol Tables</i> in the section <i>Symbol Table and Symbol Maintenance</i> .
PF10	Left	Scroll left in the source.
PF11	Right	Scroll right in the source.
PF12	Curso	Place the source line entered at the editor command prompt at the top of the source. You can omit leading blanks.

The editor commands mentioned in the table and the complete command syntax that applies are described in detail in the *Software AG Editor* section in the *Editors* documentation of Natural for Mainframes.

Editing Macro Sources for Dynamic JCL Generation

Jobs with **JCL location** MAC (macro Natural source) use the Entire Operations MACRO facility, which allows the use of variables anywhere in the JCL. Dynamic generation of JCL refers to the automatic substitution of these variables with their current values either when the job network is activated or when the job is submitted. The same job can thus have different job cards for different activation.

Dynamic JCL can be defined for jobs using the edit function in the Job Maintenance facility of Entire Operations. To convert existing JCL to the Entire Operations MAC format, use the JCL-IMPORT function in the job definition with JCL location as NAT (Natural source). In all cases, the MACRO editor command must be used to generate the final JCL. The TEST editor command can be used to test the generated JCL.



Note: The TEST command activates only job and network symbol tables, i.e., the macro test may fail with `symbol not found` but still run successfully when executed by monitor during loading of active JCL, when more symbol tables (e.g., calling symbol tables) are available.

This section covers the following topics:

- [Special Editor Commands for Macro Sources](#)
- [Escape Character for Natural Source Lines](#)
- [Parameter List](#)
- [Variables in Dynamically Generated JCL](#)
- [Inserting Text Objects into JCL](#)
- [Parameters for Included Text Objects](#)
- [Nested \(Recursive\) #EOR-INCL Statements](#)
- [Replacement of Parameters within the Text Object](#)

- [Examples of #EOR Statements](#)

Special Editor Commands for Macro Sources

The following special editor commands are available for a macro source:

Command	Description
MACRO	<p>Compile the macro JCL into an executable Natural object. This object will be executed by Entire Operations during JCL load. It will generate the executable active JCL. (This command is similar to the STOW command used for standard Natural programs.)</p> <p>Note: The MACRO command can be executed in batch mode, e.g., for whole networks.</p> <p>See also Bulk Execution of MACRO Commands.</p>
TEST	<p>Generate object for testing purposes. The generated JCL will show up in a new window.</p> <p>Note:</p> <ol style="list-style-type: none"> 1. The macro source is not saved during execution of the MACRO command. 2. The generated object is not compiled with the Natural STOW command during execution of the TEST command.

Run Number used for the TEST Command

The reserved run number used during the TEST command is -3. During TEST, a temporary active symbol table with this run number will be used. This avoids any influence on the defined master symbol table. In the log display, this reserved run number will not be displayed.

Escape Character for Natural Source Lines

Natural source lines must be preceded by the **activation escape character** of the job (see *Symbol Escape Characters* in the section *Symbol Table and Symbol Maintenance*).

The activation escape character at the beginning of a line distinguishes the line as a Natural statement from the JCL. The use of Natural statements provides full Natural functionality in dynamic JCL, including access to Adabas and Entire System Server. This means you do not need to learn any special control statements. All Natural statements used in dynamic JCL must be coded in structured mode.

Variables are user-defined and can be used in any part of the JCL, including the file name and control card. Variables are distinguished in the JCL by preceding them with an escape character: the activation escape character denotes variables to be substituted at activation time; the **submit escape character** denotes variables to be substituted at job submission time (see *Symbol Escape Characters* in the section *Symbol Table and Symbol Maintenance*).



Note: These escape characters can be changed by the system administrator. However, this should only be done if absolutely necessary, for example for countries with a different alphabet. It is not advisable to use escape characters which have a defined meaning in a particular operating system or which are already defined as escape characters, e.g., \$ in BS2000 or UNIX.

Parameter List

Each JCL for a job with JCL location MAC (macro Natural source) must start with the declaration of the common user exit parameter list `NOPXPL-A`. The parameter `P-CALL-PLACE` (see the section *User Exits*) will contain MAC:

```
§ DEFINE DATA PARAMETER USING NOPXPL-A
```

The first character (here: §) is the **activation escape character** of the job.

Variables in Dynamically Generated JCL

You can use four types of variables:

- A variable from the symbol table specified for the job;
- A variable from the parameter section (P- . . .);
- A local variable defined in this Natural program, which can be computed in your program (L- . . .);
- A Natural system variable (TIME, DATE, etc.) which is distinguished by an asterisk (*), for example, *TIME and *DATE.



Note: All variables with prefixes other than P-, L- or * are assumed to be found in the symbol table. Symbol replacement itself (without embedding Natural statements) is also available for standard JCL (JOB (**Standard Job**) type jobs). See *Symbol Replacement* in the section *Symbol Table and Symbol Maintenance*.

This section covers the following topics:

- [Sample JCL](#)
- [#GET-SYMBOL](#)
- [#SET-SYMBOL](#)
- [#SET-SYMBOL -M](#)
- [General Notes](#)

- Examples of Using the Activation Escape Character

Sample JCL

The following sample JCL illustrates the use of variables in the dynamically generated JCL of a job with JCL location MAC (macro Natural source):

```

$ DEFINE DATA PARAMETER USING NOPXPL-A
$ LOCAL      /* ... ALL LOCALS SHOULD START WITH 'L-'
$ 1 L-01     (A30)
$ 1 CLASS   (A01)      /* FROM SYMBOL TABLE, FOR #GET-SYMBOL
$ END-DEFINE
$ * -----
#GET-SYMBOL CLASS
$ COMPRESS P-NETWORK P-JOB INTO L-01
//SNOPEX JOB , $P-OWNER,MSGCLASS=$MSGCLASS,CLASS=$CLASS
//STEP01 EXEC PGM=NOPCONTI,PARM='C=0004'
//STEPLIB DD DISP=SHR,DSN=$STEPLIB
/* DEVICE: $*DEVICE, INIT-USER: $*INIT-USER, TIME: $*TIME
/* L-01 : $L-01
$ IF CLASS = 'K'      /* SYMBOL USED IN STATEMENT -> #GET-SYMBOL
/* THE CLASS IS $CLASS
$ ELSE
/* ANOTHER MSGCLASS FOUND
$ END-IF
/*
    
```

The variables used in the JCL are assumed to have the following current values:

Symbol Table Variables

STEPLIB	NOP.SYSF.DEV.LOAD
CLASS	K
MSGCLASS	X

Natural System Variables

*DEVICE	BATCH
*INIT-USER	EORMON

Variables from the Parameter Section

P-NETWORK	EX131A
P-JOB	EX-1-24
P-OWNER	SN

When the job is activated, Entire Operations substitutes the variables with their current values. The following JCL is generated as a result:

```
//SNNPEX JOB ,SN,MSGCLASS=X,CLASS=K
//STEP01 EXEC PGM=NOPCONTI,PARM='C0004'
//STEPLIB DD DISP=SHR,DSN=NOP.SYSF.DEV.LOAD
//* DEVICE: BATCH, INIT-USER: EORMON
//* L-01 : EX131-A EX-1-24
//* THE CLASS IS K
/*
```

#GET-SYMBOL

#GET-SYMBOL can be used within macro JCL to put an active symbol value into a local variable.

The symbol is taken from where it is found first in the symbol search path.

The syntax is as follows:

```
#GET-SYMBOL variable [value]
```

value is optional. If it is not specified, the variable will be filled with the contents of a symbol with the same name.

value must be of format A (alphanumeric). To convert *value* to a numeric field, use the Natural VAL system function.

```
#GET-SYMBOL variable(index)
```

With this syntax, an instance of a multiple value symbol can be obtained. The variable must be defined as a local array with a sufficient array size.

Example:

```
#GET-SYMBOL L-SYMBOL(L-IND)
```

#SET-SYMBOL

#SET-SYMBOL can be used within macro JCL to set an active symbol and its value from a local variable or string.

The symbol will be set in the active symbol table which belongs to the active job.

For #SET-SYMBOL, a job must have a locally defined symbol table.

The syntax is as follows:

```
#SET-SYMBOL symbol [value]
```

value is optional. If it is not specified, the symbol will be filled with the contents of a local variable with the same name.

#SET-SYMBOL -M

#SET-SYMBOL can be used within macro JCL to set a master symbol and its value from a local variable or string.

The symbol will be set in the master symbol table which belongs to the active job.

For #SET-SYMBOL-M, a job must have a locally defined symbol table.

The syntax is as follows:

```
#SET-SYMBOL-M symbol [value]
```

value is optional. If it is not specified, the symbol will be filled with the contents of a local variable with the same name.

The value may contain symbols itself. Refer to the [examples](#) in the following section.

General Notes

Text strings must not contain blanks.

Blanks are used as separators for the parameters of #GET-SYMBOL and #SET-SYMBOL.

Examples of Using the Activation Escape Character

The activation escape character is §.

Statement	Description
#GET-SYMBOL J	Move the contents of symbol J to the local variable J.
#GET-SYMBOL #J \$BB	Move the contents of symbol BB to the local variable #J.
#GET-SYMBOL #J '\$BB'	Move the contents of symbol BB to the local variable #J.
#GET-SYMBOL L-MULT '\$?MV<M1, §I>'	Move the result of a symbol function call into the local variable L-MULT.
#SET-SYMBOL J	Set the active symbol J to the contents of the local variable J.
#SET-SYMBOL I #A	Set the active symbol I to the contents of the local variable #A.
#SET-SYMBOL AA 'text'	Set the active symbol AA to the value <i>text</i> .
#SET-SYMBOL BB §I	Set the active symbol BB to the contents of symbol I.
#SET-SYMBOL BB '\$I'	Set the active symbol BB to the contents of symbol I.
#SET-SYMBOL CC '\$D.§E'	Concatenate symbol values of D and E, and put the result into the active symbol CC.
#SET-SYMBOL-M DD 'value'	Set the master symbol DD to the value <i>value</i> .
#SET-SYMBOL-M TAGYMMDD '&!D<T, YMMDD>'	Sets the result of the symbol function &!D<T, YMMDD> (e.g. 150211) into the master symbol TAGYMMDD.

Inserting Text Objects into JCL

You can insert Natural text objects anywhere in your JCL.

This feature is not limited to jobs with JCL location MAC (macro Natural source), but can also be used from within standard JCL.

The meta commands described here will be stripped from the active JCL and the submitted JCL.

The syntax is as follows:

```
#EOR-INCL LOC=NAT LIB=library MEM=text-object
```

Parameters of the #EOR-INCL Statement

Parameter	Description
LOC	Location NAT Natural text object. This is the only location which is currently allowed for text objects.

Parameter	Description
LIB	Library of the text object.
MEM	Name of the text object.

If the text object cannot be read, the JCL generation will be aborted with an error message.

Symbol replacement is possible within the `#EOR-INCL` statement.

Parameters for Included Text Objects

You may invoke included text objects with specific parameters.

For each parameter you want to pass to the text which is included by a `#EOR-INCL` statement, you must code a `#EOR-PARM` line in front of the `#EOR-INCL` statement.

These parameters will be in effect only for the text included (and all nested inclusions) in the following `#EOR-INCL` statement.

The syntax is as follows:

```
#EOR-PARM parameter = value [parameter = value]...
```

Multiple parameters can be passed on the `#EOR-PARM` line. This is limited by the line size.

Parameter values implicitly have the format A (alphanumeric).

Parameter values may contain blanks. In this case, they must be included in apostrophes (' ') or double quotes (" ").

Nested (Recursive) #EOR-INCL Statements

You may use nested `#EOR-INCL` statements inside text objects which have already been included. These nested inclusions may have their own parameter lists.



Note: Be aware that the total number of inclusions for a JCL is limited by the Natural editor buffer size for the Natural task executing.

Replacement of Parameters within the Text Object

Parameters for text objects can be used within the included text object like any other symbol. They have precedence over the symbols of the current symbol table of the job (and the other symbol tables in the search hierarchy).

Therefore, parameters for included text objects can temporarily override symbols with the same name. The scope of a parameter is only the included text object, and the text objects which are invoked from within this text object by nested calls.

The lifetime of parameters defined via `#EOR- PARM` is JCL load time only. Normal symbol replacement can be used within included text objects like everywhere else in the JCL.

Examples of #EOR Statements

Example 1:

```
...
#EOR-INCL LOC=NAT LIB=JCLLIB MEM=$OBJECT
...
```

Include the Natural text object the name of which is in the symbol `OBJECT` from the library `JCLLIB`.

Example 2:

```
...
#EOR-PARM DBID = 9
#EOR-PARM TEXT-1 = "a string with blanks"
#EOR-INCL LOC=NAT LIB=USERLIB MEMBER=BLOCK1
/* JCL statement
#EOR-PARM DBID = 10
#EOR-PARM TEXT-1 = "another string"
#EOR-INCL LOC=NAT LIB=USERLIB MEMBER=BLOCK1
...
```

Include the Natural text object `BLOCK1` from the library `USERLIB`. The parameters `DBID` and `TEXT-1` are passed to the text object with different values for the two invocations.

Locking of Natural Sources

If you start Natural with the profile parameter `SLOCK` set to `PRE`, a lock avoids that two or more users can concurrently edit a Natural source object from an Entire Operations, an Entire Operations GUI Client or within a native Natural environment.

If `SLOCK` is set to `POST` or `OFF`, no locking is performed but the Entire Operations editor detects that several users edit a source at the same time.

You can use the Natural system command `UNLOCK` to check or unlock locked Natural source objects.

For detailed information on `SLOCK` and `UNLOCK`, refer to the relevant sections in the *Natural Parameter Reference* documentation.

Additional Prerequisites for Source Locking

- In a UNIX or Windows environment, the Natural Development Server must be installed in the current Natural FNAT system file.
- If you want to avoid concurrent source editing among users in a local Entire Operations GUI Client environment, in addition to Natural, you also have to start the Natural RPC server with `SLOCK` set to `PRE`.

27

Defining and Managing Job Input Conditions

▪ Listing Input Conditions	272
▪ Adding and Modifying a Master Input Condition	275
▪ Input Condition References	281
▪ Global Conditions	284
▪ Input Condition: User Exit	284
▪ Input Condition: Multiple Suffixes	288
▪ Input Condition: File Existence	291
▪ Input Condition: Mailbox	294
▪ Input Condition: Symbol Value	296
▪ Input Condition: User Switch - BS2000	299
▪ Input Condition: Job Variable - BS2000	301
▪ Listing Jobs Linked to an Input Condition	304
▪ Deleting an Input Condition Definition	306
▪ Displaying Currently Active Conditions	306

Jobs within a job network are linked by user-defined logical conditions. Logical conditions (see also the *Concepts and Facilities* documentation) are variables within Entire Operations and describe job dependencies.

During execution of networks and submission of jobs, Entire Operations automatically checks the status of logical conditions and triggers system actions accordingly. Alternatively, logical conditions can be set by an **API routine** (see the relevant section).

In order to link two jobs, an input condition (prerequisite) defined for one job must also be defined as output (End-of-Job) condition for the preceding job.



Note: It is possible to define a maximum of 40 input conditions per job. If you need more input conditions, you must use intermediate dummy jobs to collect the conditions. See also *Job Execution as a Dummy Job*.

Related Topics:

- *Defining Schedule Dependencies for an Input Condition* in the section *Schedule Maintenance*
- *Maintaining Active Job Conditions* in the section *Active Job Networks*
- *Logical Conditions* in the *Concepts and Facilities* documentation.

Listing Input Conditions

> To list input conditions

- 1 Select option 1 on the **Main Menu** to display the **Network Maintenance** screen.
- 2 Issue the line command **L** for the network containing the job to be linked.
- 3 Type **I** in the line command input field of the job with which you wish to associate the input condition.
- 4 Press **ENTER**.

An **Input Conditions Maintenance** screen similar to the example below appears:

```

29.03.16          ***** Entire Operations *****          15:44:24
                   Input Conditions Maintenance
Owner EXAMPLE    Network B60-FLOW  Version          Job JOB-03
-----
C Condition      Reference Type      Sched.Dep.  U ex. Library  Exit
_ E60-JOB2-01    RUN      true      Y
_ EX-2           RUN      true      Y
_ EX-3           RUN      true      Y
_ E05-UR01-NOK   RUN      true      Y

***** Bottom of Data *****
D Delete  E Edit  Exit  M Modify  S Sched.Dep.  W Where used

Command => _____

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Add   End   ACond Save      Up   Down                      Menu
    
```

All input conditions defined for the job are listed on the screen.

The columns and commands available on the screen are explained in [Columns: Input Conditions Maintenance](#) and [Commands: Input Conditions Maintenance](#).

This section covers the following topics:

- [Columns: Input Conditions Maintenance](#)
- [Commands: Input Conditions Maintenance](#)

Columns: Input Conditions Maintenance

The following table explains the column headings for the data listed on the [Input Conditions Maintenance](#) screen:

Column	Description
C	One-character line command input field. The line commands available are described in Line Commands: Input Conditions .
Condition	User-defined condition name. See also Restrictions for Condition Names and Global Conditions .

Column	Description																																				
Reference	Reference date used to refer to a certain occurrence of the input condition. For possible entries, see Possible References for Input Conditions in the section Input Condition References .																																				
Type	User-defined condition type expressed as any of the following: <table border="1"> <tr> <td>true</td> <td>Condition must exist for the job to be submitted.</td> </tr> <tr> <td>false</td> <td>Condition must not exist for the job to be submitted.</td> </tr> <tr> <td>exclusive</td> <td>Exclusive use of the condition.</td> </tr> <tr> <td>destruct.</td> <td>Condition will be destroyed after use.</td> </tr> <tr> <td>dummy</td> <td>If the condition is satisfied, the job is started as a temporary dummy job.</td> </tr> <tr> <td>extern +</td> <td>Condition from another network must exist.</td> </tr> <tr> <td>extern -</td> <td>Condition from another network must not exist.</td> </tr> <tr> <td>Exit</td> <td>Condition depends on the result of a user exit.</td> </tr> <tr> <td>File +</td> <td>File must exist.</td> </tr> <tr> <td>File -</td> <td>File must not exist.</td> </tr> <tr> <td>User Sw +</td> <td>User switch must exist (BS2000 only).</td> </tr> <tr> <td>User Sw -</td> <td>User switch must not exist (BS2000 only).</td> </tr> <tr> <td>Job Var.</td> <td>Condition depends on a job variable (BS2000 only).</td> </tr> <tr> <td>Symbol</td> <td>Condition depends on the value of a symbol in a symbol table.</td> </tr> <tr> <td>Mailbox +</td> <td>Condition must exist; prompted in mailbox.</td> </tr> <tr> <td>Mailbox -</td> <td>Condition must not exist; prompted in mailbox.</td> </tr> <tr> <td>Recov.tmp.</td> <td>Condition is used for recovery (temporary - created by Entire Operations Monitor; for active jobs only).</td> </tr> <tr> <td>Sched.Dep.</td> <td>If the condition is defined to be schedule-dependent, a short summary of the dependency appears in this column. For more information, see Defining Schedule Dependencies for an Input Condition in the section Schedule Maintenance.</td> </tr> </table>	true	Condition must exist for the job to be submitted.	false	Condition must not exist for the job to be submitted.	exclusive	Exclusive use of the condition.	destruct.	Condition will be destroyed after use.	dummy	If the condition is satisfied, the job is started as a temporary dummy job.	extern +	Condition from another network must exist.	extern -	Condition from another network must not exist.	Exit	Condition depends on the result of a user exit.	File +	File must exist.	File -	File must not exist.	User Sw +	User switch must exist (BS2000 only).	User Sw -	User switch must not exist (BS2000 only).	Job Var.	Condition depends on a job variable (BS2000 only).	Symbol	Condition depends on the value of a symbol in a symbol table.	Mailbox +	Condition must exist; prompted in mailbox.	Mailbox -	Condition must not exist; prompted in mailbox.	Recov.tmp.	Condition is used for recovery (temporary - created by Entire Operations Monitor; for active jobs only).	Sched.Dep.	If the condition is defined to be schedule-dependent, a short summary of the dependency appears in this column. For more information, see Defining Schedule Dependencies for an Input Condition in the section Schedule Maintenance .
true	Condition must exist for the job to be submitted.																																				
false	Condition must not exist for the job to be submitted.																																				
exclusive	Exclusive use of the condition.																																				
destruct.	Condition will be destroyed after use.																																				
dummy	If the condition is satisfied, the job is started as a temporary dummy job.																																				
extern +	Condition from another network must exist.																																				
extern -	Condition from another network must not exist.																																				
Exit	Condition depends on the result of a user exit.																																				
File +	File must exist.																																				
File -	File must not exist.																																				
User Sw +	User switch must exist (BS2000 only).																																				
User Sw -	User switch must not exist (BS2000 only).																																				
Job Var.	Condition depends on a job variable (BS2000 only).																																				
Symbol	Condition depends on the value of a symbol in a symbol table.																																				
Mailbox +	Condition must exist; prompted in mailbox.																																				
Mailbox -	Condition must not exist; prompted in mailbox.																																				
Recov.tmp.	Condition is used for recovery (temporary - created by Entire Operations Monitor; for active jobs only).																																				
Sched.Dep.	If the condition is defined to be schedule-dependent, a short summary of the dependency appears in this column. For more information, see Defining Schedule Dependencies for an Input Condition in the section Schedule Maintenance .																																				
U	Usage. For active input conditions with schedule dependency only. One of the following values appears: <table border="1"> <tr> <td>Y</td> <td>Condition is being used for the current activation.</td> </tr> <tr> <td>N</td> <td>Condition is not being used for current activation.</td> </tr> <tr> <td colspan="2">Schedule is always dynamically tested for this display.</td> </tr> </table>	Y	Condition is being used for the current activation.	N	Condition is not being used for current activation.	Schedule is always dynamically tested for this display.																															
Y	Condition is being used for the current activation.																																				
N	Condition is not being used for current activation.																																				
Schedule is always dynamically tested for this display.																																					
ex.	Y Condition must exist.																																				

Column	Description	
	N	Condition must not exist.
Library	Natural library where a defined user exit resides.	
Exit	Natural user exit to set the condition (see the section <i>Input Condition: User Exit</i>).	

Commands: Input Conditions Maintenance

This section describes the line commands and special PF keys available on the **Input Conditions Maintenance screen**.

Line Commands: Input Conditions Maintenance

Next to each input condition listed on the screen, you can enter one of the following line commands to perform the functions described in the table below:

Line Command	Description
D	Delete an input condition definition.
E	Edit user exit to set input condition.
M	Modify input condition definition.
S	Add/modify schedule dependency. See <i>Defining Schedule Dependencies for an Input Condition</i> in the section <i>Schedule Maintenance</i> .
W	Display jobs for which the condition is used as input or output condition.

Special PF Keys: Input Conditions Maintenance

You can perform the following function using this special PF key:

PF Key	Name	Function
PF4	ACond	Display list of currently active conditions.

Adding and Modifying a Master Input Condition

» To add an input condition definition for a job

- 1 Press PF2 (Add) on the **Input Conditions Maintenance screen**.

A **Master Input Condition** window similar to the example below opens:

```

16-09-20          ***** Entire Operations *****          14:37:37
+-----+-----+
Owner E !
!
!           Master Input Condition Addition
!
----- !
C Condi !      Owner   ==> EXAMPLE___ ! it
!      Network  ==> B60-FLOW___   Version ==>
!      Condition ==> _____
!      Reference ==> _____
!
!      Usage (mark with Y or N)
!      Must Exist: Y   Exclusive: _   Destroy after usage: _
!      Activation Mode ==> N
!
!      Depending on
!      Exit           ==> _           Multiple Suffixes ==> _
***** !      File Existence ==> _           Mailbox           ==> _ ! *****
D Delet !      User Sw.(BS2000) ==> _           Symbol Value       ==> _
!      Job Var.(BS2000) ==> _
!
Command ! -PF1---PF2---PF3-----PF5-----PF9---PF10---PF12--
!      Help  Add   End       Save           Xref  ScDep  Menu
+-----+-----+

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Add   End   ACond Save           Up   Down           Menu

```

- 2 Enter the required values. The input fields are explained in *Fields and Selection Options: Input Condition*.

When you are finished, press PF5 (Save) to save your entries.

- 3 If you want to add another input condition, press PF2.

The **Master Input Condition Addition** window is cleared and you can enter a new input condition definition.

- 4 Press PF3 (End) to return to the **Input Conditions Maintenance** screen.

All new input conditions are listed on the screen.

- 5 If you defined a user exit to set the condition, you can edit the routine by using the line command E (Edit Exit) for the condition on the **Input Conditions Maintenance** screen.

For more information, see *Defining and Editing an Input Condition of the Type User Exit*.

➤ **To modify an input condition definition for a job**

- 1 On the **Input Conditions Maintenance screen**, type the line command M next to the input condition you want to modify.

A **Master Input Condition window** similar to the previous example opens.

- 2 Change the entries as required. The input fields are explained in *Fields and Selection Options: Input Condition*.

When you are finished, press PF5 (Save) to save your changes.

- 3 Press PF3 (End) to return to the **Input Conditions Maintenance** screen.



Note: After an input condition has been defined or modified, a loop check is performed for the network. The same conditions apply as described in *Checking for a Loop in a Job Network* in the section *Network Maintenance*, with one exception: if a loop is detected in the job flow, no corresponding message appears.

- [Fields and Selection Options: Input Condition](#)
- [Special PF Keys: Input Condition](#)

Fields and Selection Options: Input Condition

The input fields in the **Master Input Condition window** are described in the following table.

Field	Description
Owner	The job owner is assumed as the default. You can select a different owner. The field may contain symbols.
Network	The job network is assumed as the default. You can select a different network. The field may contain symbols.
Version	Version of the network.
Condition	Name assigned to the condition. The condition name and its reference date uniquely identify an active condition. See also <i>Restrictions for Condition Names</i> .
Run	Current run number (for active jobs only).
Reference	Reference date to specify which occurrence of this definition the job uses. For possible input values, see <i>Possible References for Input Conditions</i> in the section <i>Input Condition References</i> .
Activation Mode	A Input condition definition will always be activated (for job activations as well).

Field	Description	
	N	Default. Input condition definition will be activated for network activations only.
Usage section:		
Must Exist	Y	Specifies that the condition must exist (be true) as a prerequisite to job submission.
	N	Specifies that the condition must not exist (be false) as a prerequisite for job submission. Alternatively, this field also controls the setting of the condition according to the existence or non-existence of a file specified in the File Existence field (file or member in a file).
Exclusive	Y	Specifies that when this condition is in use, no other job can access this condition until it is free (job finished).
	N or blank	Any job can use the condition at any time. This feature is useful to prevent simultaneous execution of jobs with the same input conditions. Default.
Destroy after usage	Y	Specifies that the condition is automatically reset after the job is submitted.
	N or blank	Do not reset condition: later job runs can use this condition according to the Reference date. Default.
Activation Mode	A	Input condition definition will always be activated (for job activations as well).
	N	Input condition definition will be activated for network activations only. Default.
Depending on section:		
User Exit	<p>Enter any character and press ENTER if the condition is to be set by a user exit.</p> <p>A window opens in which you can enter the name of the user exit routine and the name of the Natural library in which the user exit resides.</p> <p>For further information and the fields/options available in the window, see <i>Input Condition: User Exit</i>.</p>	
File Existence	<p>Enter any character and press ENTER to define an input condition dependent on the existence or non-existence of a file.</p> <p>A window opens in which you can enter the name of the file.</p>	

Field	Description
	For further information and the fields/options available in the window, see Input Condition: File Existence .
User Sw. (BS2000)	<p>User switch (BS2000 only).</p> <p>Enter any character and press ENTER to define an input condition dependent on the existence or non-existence of a user switch.</p> <p>A window opens in which you can enter the name of the user switch.</p> <p>For further information and the fields/options available in the window, see Input Condition: User Switch - BS2000.</p>
Job Var. (BS2000)	<p>Job variable (BS2000 only).</p> <p>Enter any character and press ENTER to define an input condition dependent on a comparison with the contents of a BS2000 job variable.</p> <p>For further information and the fields/options available in the window, see Input Condition: Job Variable - BS2000.</p>
Multiple Suffixes	<p>Enter any character and press ENTER to define a symbol to be used for the active condition name.</p> <p>A window opens in which you can enter a symbol name.</p> <p>For further information and the fields/options available in the window, see Input Condition: Multiple Suffixes.</p>
Mailbox	<p>Enter any character and press ENTER to define a user prompt to a mailbox.</p> <p>A window opens in which you can enter the name of the mailbox.</p> <p>For further information and the fields/options available in the window, see Input Condition: Mailbox.</p>
Symbol Value	<p>Enter any character and press ENTER to define an input condition dependent on a comparison with the value of a symbol in a symbol table.</p> <p>A window opens in which you can enter the symbol name and other parameters.</p> <p>For further information and the fields/options available in the window, see Input Condition: Symbol Value.</p>

Restrictions for Condition Names

The name of a condition can contain numbers and letters as required. The maximum name length is 20 bytes.

The following restrictions apply:

- Umlauts are not allowed.
- The use of special characters is restricted to the following:

-+/\$#\$_&

- Names of **global conditions** begin with a plus sign (+).
- The **activation escape character**, the **submit escape character** and a period (.) symbol delimiter are still accepted if symbol replacement is allowed for the relevant name fields.
- The following condition names are reserved for special purposes and may not be used for common conditions:

Reserved Condition Name	Explanation
NET-BEGIN	Used for subnetwork control.
NET-END	These reserved conditions are described in detail in the section Link the Main Network .
NET-END-NOTOK	
NET-END-OK	
P-STOPCYC - <i>jobname</i>	<p>If this condition is set in the active symbol table of a job with the special type C, the cyclic execution will be stopped.</p> <p>For detailed information, see the field Special Type in the section <i>Fields: Job Definition</i>.</p>
<i>jobname</i> -MAX-RETRY	The special condition <i>jobname</i> -MAX-RETRY will be set by the Entire Operations Monitor when the message EOR5316 (Recovery Retry Maximum:1: exceeded) is issued during a Job Recovery .

Special PF Keys: Input Condition

You can perform the following functions from the **Master Input Condition window** using these PF keys:

PF Key	Name	Function
PF9	Xref	<p>Display the jobs for which the condition is defined as an input or output condition. The same window opens as if you had issued the W (Where used) line command for the condition.</p> <p>For further information, see Displaying Jobs Linked to a Condition.</p>
PF10	ScDep	<p>Define schedule dependency for an input condition. The same window opens as if you had issued the S (Sched.Dep.) line command for the condition.</p> <p>For further information, see Defining Schedule Dependencies for an Input Condition in the section <i>Schedule Maintenance</i>.</p>

Input Condition References

To check an input condition, you must know which reference is meant. References can result in time or run number intervals.

The simplest reference is `RUN`, which refers to conditions set in the current network run. However, if you define an external input condition (which is not produced by the current network), you should always remember that different networks usually have different run numbers, which implies that `RUN` makes no sense in this case.

Run numbers are not assigned sequentially in chronological order. For references to previous network runs use `LNR`.

With the exception of `RUN`, all references described in this section also apply to [global conditions](#).

- [Possible References for Input Conditions](#)
- [Relative Values](#)

Possible References for Input Conditions

The following table describes all references you can enter in the **Reference** field of the [Master Input Condition window](#).

Reference	Unit of Relative Value	Description
AAC		Job uses condition only if there is at least one entry in the active database for the owner, the network and the job.
ABS		Job uses condition only if it is absolute. Absolute conditions are independent of run numbers and can exist only once under the same name.
ANY		Job uses any occurrence of the condition, except <code>ABS</code> (absolute), which has a reserved run number.
ANT		Job uses condition only if there is no entry in the active database for this owner, network and job.
DAT	Days	Job uses the condition as set by the network run on the current date.
<i>date</i>		A date entered in the format <code>YYYYMMDD</code> . Job uses condition only if set on the explicit date. The job then uses the condition as set by the network run on this date (does not apply when job can run more than once daily).
DST		Job uses the condition as set during the network run on the date specified as the job start time.

Reference	Unit of Relative Value	Description
DUM		<p>If this condition is satisfied, the job is started as a temporary dummy job. If this condition is not satisfied, the job is started normally.</p> <p>If several conditions with the reference DUM are defined for a job, only one condition must be satisfied for the job to be executed as a dummy. The condition can have a special dependency (for example, on a file).</p> <p>The active condition will also be accepted if it has the reference ABS (absolute).</p> <p>If a job is started as a temporary dummy job on account of a condition, then this will be written to the log.</p>
HRC	Hours	<p>Job uses the condition only if it was set a defined number of hours previous to the check time of the condition.</p> <p>This reference can only be entered with a relative hour value. HRC - 24 is the default value if this field is left blank and if the condition is set in a different network. (RUN is the default in the same network.)</p>
LNR	Hours	<ul style="list-style-type: none"> ■ If the condition was set by another network: Job uses the condition if it was set by the most recent run in the previous <i>nnn</i> hours. ■ If the condition was set by an earlier run of the same network: Job uses the condition if it was set by an earlier run in the previous <i>nnn</i> hours. ■ The condition is not set if an error occurred during the most recent or earlier run. <p>This reference is recommended for constructing chains of networks and must be followed by a relative value (see <i>Relative Values</i>).</p>
LNT	Hours	<p>This reference is used like LNR.</p> <p>Additionally, the condition is set to true if the creating job network did not have an active occurrence in the time frame to be observed.</p>
MON	Months	Job uses the condition as set by the network run of the current month.
NSD		Job uses the condition as set during the network run on the date specified as the network start time.
PDA	Days	<p>Job uses the condition only if set on the same production date.</p> <p>The production date end time can be defined in the Entire Operations default settings: see <i>Default Setting (2)</i> in the <i>Administration</i> documentation.</p> <p>Note: This reference does not evaluate schedules or calendars.</p>

Reference	Unit of Relative Value	Description
PDS		<p>Job uses the condition only if set on the same production date.</p> <p>The production date end time can be defined in the Entire Operations default settings: see <i>Default Setting (2)</i> in the <i>Administration</i> documentation.</p> <p>Note: This reference does not evaluate schedules or calendars.</p>
RCA		<p>A job with multiple active subnetworks uses the same input conditions defined for a predecessor job that runs in the primary subnetwork.</p> <p>If Multiple suffixes are used as an input condition (see <i>Input Condition: Multiple Suffixes</i>) for the predecessor, the multiple suffixes are appended to the job.</p> <p>If RCA is specified, the output condition of the corresponding predecessor job must be referenced with RCM: see <i>RCM</i> in the section <i>Field Descriptions: Output Conditions</i>.</p>
RUN	Run numbers	<p>Job uses the condition as set by the current network run.</p> <p>This is the default value if this field is left blank and if the condition is set in the same network.</p> <p>(HRC - 24 is the default in a different network.)</p>
WEK	Weeks	Job uses the condition as set by the network run of the current week.
WCC	Days	Day in real time, relative to the current day.
WCW	Days	Day in calendar (work day), relative to the current day.
WCS	Days	Day in schedule, relative to the current day.

Relative Values

Some references can be followed by a minus (-) or plus (+) sign and a numeric offset. This is called a relative value. For example:

DAT - 1	Refers to yesterday.
HRC - 2	Refers to the previous 2 hours.
WEK - 1	Refers to the previous week.

Global Conditions

Logical conditions are either set for a single job network or independently of any networks. Independent conditions are referred to as global conditions.

A global condition is not restricted to a particular owner, network or job but reflects the current value of a condition set for the given environment. It is defined once and can be used in several networks and job environments.

The following applies to a global condition:

- A global condition has the prefix + (plus sign).
- A global condition is assigned to the owner SYSDBA and to the network SYSDBA.
- A global condition gets the reference `abs` (absolute). The reference `RUN` is accepted but is converted to `abs` at runtime.

This document covers the following topics:

- [Restrictions for Global Conditions](#)

Related Topic:

- [Maintaining Active Job Conditions](#) in the section *Active Job Networks*.

Restrictions for Global Conditions

For global conditions, only the following [references](#) are allowed:

With the definition of an active condition	ABS, ANY, RUN
If used as input condition	HRC, DAT, PDA, WEK, MON, ABS, ANT, DUM, RUN, ANY
If used as output condition	ABS, RUN

Input Condition: User Exit

Input conditions can depend on the result of a user exit (`P-CALL-PLACE` set to `IC0`; see the section *User Exits*). If a user exit is defined for an input condition, Entire Operations automatically executes the exit when checking the status of input conditions during the prerequisite check. The user exit can perform any database or Entire System Server call to obtain the necessary information. This allows Entire Operations to react to complex or user-specific dependencies.

User exits are Natural subprograms and can be edited with the Entire Operations editor. See also the section [User Exits](#).

When defining a user exit as an input condition, consider the following:

- The Entire Operations Monitor sets the parameter field P-RC (return code) to 0 (zero) before the user exit is called.
- You can also specify an input condition user exit for an input condition with the reference [DUM](#) (dummy job).

In this case, the job executes as [dummy due to condition](#) if the exits returns 0 for P-RC.

The job executes normally if the exit returns a non-zero value for P-RC.

- [Defining and Editing an Input Condition of the Type User Exit](#)
- [Fields: Input Condition User Exit](#)
- [Special PF Keys: Input Condition User Exit](#)
- [Example of a User Exit](#)

Defining and Editing an Input Condition of the Type User Exit

➤ To define a user exit which sets the condition

- 1 In the [Master Input Condition window](#), enter any character in the **Exit** field under the **Depending on** heading.
- 2 Press ENTER.

An **Input Condition Exit** window similar to the example below opens:

```

16-09-20          ***** Entire Operations *****          15:42:21
+-----+-----+
Owner E !
!
!           Master Input Condition Addition
!
----- !
C Condi !      Owner   ==> EXAMPLE____ ! it
_ E60-J !      Network ==> B60-FLOW__   Version ==>
!      Condition ==> COND-1_____
!      Reference  ==> _____
!
! Usage ( +-----+
! Must !
! Activat !           Input Condition Exit
!
! Dependi ! Condition ==> COND-1
! Exit !           Run ==>
***** ! File !
D Delet ! User ! Value will be determined by
!      Job V !           Natural Library ==> _____
!
!                               Exit ==> _____
Command ! -PF1---P !
! Help A ! PF1 Help PF3 End PF5 Save PF9 Delete ! u
+-----+-----+
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help Add End ACond Save Up Down Menu

```

In this window you can enter the name of the user exit and the name of the Natural library in which the user exit resides.

The input fields available are described in *Fields: Input Condition User Exit*.

- When finished, press PF3 (End) to save data and return to the **Master Input Condition** window.

➤ **To edit a user exit**

- On the **Input Conditions Maintenance screen**, type E (Edit Exit) in the line command input field next to an input condition of the type **Exit**.
- Press ENTER.

The first page of the Natural subprogram appears in the editor screen. See also *Example of a User Exit*.

(If no user exit is specified for the selected input condition, an appropriate message occurs instead.)

- If you are defining a new user exit, the parameter section appears in the editor screen. You can only modify parameters by field redefinition.

You can now create or modify the user exit using editor commands and PF keys. For a full description of the editor, see *Software AG Editor* in the *Natural* documentation.

For detailed information on handling user exits, see the section [User Exits](#).

Fields: Input Condition User Exit

The input fields provided for a user exit in the [Input Condition Exit](#) window are described in the following table:

Field	Description
Natural Library	Name of the Natural library where the user exit resides. This library must be different from the Entire Operations system library.
Exit	Name of the user exit which sets the condition. The user exit coding must start with DEFINE DATA PARAMETER USING NOPXPL-A. For further information, see Common User Exit Parameter Area NOPXPL-A .

Special PF Keys: Input Condition User Exit

You can perform the following function from the Input Condition Exit window using this PF key:

PF Key	Name	Function
PF9	Delete	Delete input condition dependent on user exit.

Example of a User Exit

Below is an example of a user exit which sets an input condition:

```

*
Entire Operations

* USER EXIT TO SET AN INPUT CONDITION
*
* THIS ROUTINE CHECKS THE EXISTENCE OF A FILE, DEPENDING ON
* GIVEN PARAMETERS
*
DEFINE DATA PARAMETER USING NOPXPL-A
LOCAL                               /* LOCAL VARIABLES START HERE
1 CATALOG VIEW OF CATALOG           /* An Entire System Server VIEW
  2 NODE
  2 DSNAME
  2 ERROR-CODE

```

```
2 ERROR-TEXT
*
1 #DSNAME          (A54)
END-DEFINE
* -----
RESET P-RC          /* ASSUME GOOD RETURN -> SET CONDITION
COMPRESS P-OWNER '.SYSF.SRCE' INTO #DSNAME LEAVING NO SPACE
CAT. FIND CATALOG WITH NODE = P-EXECUTION-NODE
      AND DSNAME = #DSNAME
IF CAT.ERROR-CODE NE 0
      MOVE CAT.ERROR-CODE TO P-RC          /* BAD RETURN
      MOVE CAT.ERROR-TEXT TO P-RT
      ESCAPE ROUTINE
END-IF
END-FIND           /* (CAT.)
END
```

The user exit must set a return code in P-RC.

If P-RC is not equal to 0, the condition is reset (false) and the user is notified with a message. In the example above, the returned condition code (ERROR-CODE) sets (fulfills) the input condition for which the user exit is defined if the routine finds a file with the string *owner.SYSF.SRCE*.

Input Condition: Multiple Suffixes

If you define a symbol for multiple suffixes, its contents are separated and the single fields are concatenated to the active condition name. These multiple conditions are used to wait until all parallel executing predecessors are finished.

The active conditions are created during activation of the job network. For example, if the condition name is COND and if the specified symbol contains 001003012, the active conditions CONDO01, CONDO03 and CONDO12 are created.

➤ To define an input condition dependent on multiple suffixes

- 1 In the **Master Input Condition** window, enter any character in the **Multiple Suffixes** field under the **Depending on** heading.
- 2 Press ENTER.

An **Input Condition: Multiple Suffixes** window similar to the following opens:

```

16-09-20          ***** Entire Operations *****          14:24:59
+-----+-----+
Owner E !
!
!           Master Input Condition Modification           !
----- ! ----- !
C Condi !      Owner   ==> EXAMPLE____          ! it
m E60-J !      Network  ==> B60-FLOW____   Version ==>   !
!      Condition  ==> E60-JOB1-0_____          !
!      Reference  ==> RUN_____          !
+-----+-----+
!
!           Input Condition: Multiple Suffixes           !
!
!      Condition ==> E60-JOB1-0          !
!      Run ==>          !
!
! ***** ! Suffixes will be taken from          ! *****
D Delet ! Symbol Table ==> DEMO_____   always Job Table ==> N   !
! SymTab Version ==> v1_____          !
! Symbol ==> FILE-1_____          !
Command !
! PF1 Help PF3 End PF5 Save PF7 Symbols PF9 Delete          !
+-----+-----+
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Add  End  ACond Save          Up    Down          Menu

```

Enter a symbol name and a symbol table/version, if required. See also [Fields: Input Condition Multiple Suffixes](#).

- When finished, press PF3 (End) to save data and return to the **Master Input Condition** window.

This section covers the following topics:

- [Fields: Input Condition Multiple Suffixes](#)
- [Special PF Keys: Input Condition Multiple Suffixes](#)

Fields: Input Condition Multiple Suffixes

The input fields provided for multiple suffixes in the [Input Condition: Multiple Suffixes window](#) are described in the following table:

Field	Description	
Always Job Table	Specifies whether the symbol table defined for the job is used. Possible values:	
	Y	The multiple suffix is always taken from the job symbol table. A local definition will be ignored (default).
	N	Use the symbol table defined here.
	Note: In the case of a job or network copy, it is recommended to set this field to Y. By this you make sure that always the defined symbol table of the job will be used, even if it was changed in the job definition.	
Symbol Table	Name of the symbol table with the symbol that contains the suffix(es) to be used for the condition when the job network is activated. You must specify the same symbol table in the predecessor job definition.	
SymTab Version	Version of the symbol table (if defined).	
Symbol	Name of the symbol that contains the suffix(es) to be used for the condition when the job network is activated. You must specify the same symbol in the predecessor job definition.	
Always Job Table	Possible values:	
	Y	The multiple suffix is always taken from the job symbol table. A local definition will be ignored (default).
	N	Use the symbol table defined here.
	Note: In the case of a job or network copy, it is recommended to set this field to Y. By this you make sure that always the defined symbol table of the job will be used, even if it was changed in the job definition.	

Special PF Keys: Input Condition Multiple Suffixes

You can perform the following functions from the **Input Condition: Multiple Suffixes window** using these PF keys:

PF Key	Name	Function
PF7	Symbols	List symbols in symbol table.
PF9	Delete	Delete input condition dependent on multiple suffixes.

Input Condition: File Existence

An input condition value can be dependent on the existence or non-existence of a file or of one of its members. The Monitor checks for the file or member on the job's **execution node** until the condition is satisfied.

➤ **To define an input condition that requires a file**

- 1 In the **Master Input Condition window**, enter any character in the **File Existence** field under the **Depending on** heading.
- 2 Press ENTER.

An **Input Condition depending on File** window similar to the example below opens:

```

16-09-20          ***** Entire Operations *****          15:42:21
+-----+
Owner E !
!
!           Master Input Condition Addition           !
----- !
C Condi !      Owner   ==> EXAMPLE__          ! it
_ E60-J !      Network ==> B60-FLOW__   Version ==> !
!      Condition ==> COND-1_____          !
!      Reference  ==> RUN_____          !
+-----+
!
!           Input Condition depending on File           !
!
!      Condition ==> COND-1                       !
!      Run ==>                                     !
!
** !      Condition will be set to true, if on node   31      ! **
D !      File ==> _____                      !
!      Member ==> _____                      !
!      exists ==> Y           (Y/N)              !
Co !
!      PF1 Help   PF3 End   PF5 Save           PF9 Delete   !
+-----+

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Add   End   ACond Save           Up   Down           Menu
  
```

- 3 Enter a file and a member name. The input fields are described in *Fields: Input Condition File Existence*.

When finished, press PF3 (End) to save data and return to the **Master Input Condition** window.

This section covers the following topics:

- [Fields: Input Condition File Existence](#)
- [Rules for File Names and File Checking](#)
- [Special PF Keys](#)

Fields: Input Condition File Existence

The input fields provided for a file existence check in the [Input Condition depending on File window](#) are described in the following table:

Field	Description				
Condition will be set to true, if on node	<p>This is the operating system server on which the file's existence is to be checked. The node used for the file check is always the job's execution node .</p> <p>Tip: If you must check a file on a different node, first create an additional dummy job, whose execution node is the same as the node with which the file is to be checked. This dummy job must be a predecessor of the main job.</p>				
File	<p>Name of the file that must or must not exist. If the file is not cataloged, specify the volume serial number in the format <i>file/volume</i>.</p> <p>Note: When entering a file name, remember to observe the rules for upper and lower case which are specific to some operating systems.</p> <p>Using a Wildcard in the File Name: A wildcard (*) can be used in all operating systems. For mainframe files, the wildcard may be set at the string end only. The condition will be set (or not set), if at least one file is found.</p> <p>See also Rules for File Check.</p>				
Member (optional)	<p>If the input condition is dependent on the existence or non-existence of a member in the file specified in the File field, enter the member name.</p> <p>Using a Wildcard in the Member Name: A wildcard (*) can be used at the string end. The condition will be set (or not set) if at least one member is found.</p> <p>Note: Only specify a member where necessary and possible. If this field is left blank, the existence of the whole file is checked.</p> <p>See also Rules for File Check.</p>				
exists	<p>Possible values:</p> <table border="1"> <tbody> <tr> <td>Y</td> <td>The file (or member) must exist as a prerequisite to job submission.</td> </tr> <tr> <td>N</td> <td>The file (or member) must not exist as a prerequisite.</td> </tr> </tbody> </table>	Y	The file (or member) must exist as a prerequisite to job submission.	N	The file (or member) must not exist as a prerequisite.
Y	The file (or member) must exist as a prerequisite to job submission.				
N	The file (or member) must not exist as a prerequisite.				

Rules for File Names and File Checking

The following rules apply when specifying a file as an input condition check:

BS2000 Files

The condition is satisfied only if the file is closed. For opened BS2000 files, the condition is not satisfied.

Migrated (archived) Files

Migrated (archived) files are recognized like standard existing files. If a member is included in the file existence check, the active job will be set to a permanent error, with the error text `Prerequisite File Check - Library containing member is archived.`

Entire System Server Node used for File Check

The node used for the file check is always the **execution node** of the job. The file is being checked with the access rights of the **Submit User ID** (on UNIX and Windows: `submit` and `submit group`).

If you must check a file on a different node, use a predecessor dummy job with a different **execution node** and/or **Submit User ID** for this purpose.

Variable File Name: Using Escape Characters

The fields **File** and/or **Member** can contain symbols preceded by an **activation escape character**.

If the activation escape character is used, symbol replacement is performed during the first existence check.

Symbol replacement can be used, for example, for:

- file generation groups;
- changing input files;

After a successful symbol replacement, these fields will contain the replaced value in the active job. This reduces the effort with symbol replacements.

The symbols are taken from the active symbol table assigned to the job. The symbol replacement in the file name is performed only once and the result is written back to the active input condition definition for further check. A missing symbol causes a permanent error.

It is also possible to use the **submission escape character**. In the case of an unsatisfied condition, the symbol replacement in the file name will be performed before each file check. The result is not written back. This allows more flexible use of symbols, but may cause more system overhead.

File in Use

The case `file in use` is handled as a temporary error. The file check is repeated as long as the file is in use. The waiting job will not be sent to passive wait.

z/OS: HSM Migrated Libraries

The following applies only if the operating system of an Entire System Server node is z/OS, and if the Entire System Server version is greater than or equal to Version 3.2.1. The Entire Operations Monitor performs the initialization of a file recall.

The file member check will be repeated in intervals of two minutes, until the file is reloaded. A reload will not be initiated if the file check is on file level only.

Special PF Keys

You can perform the following function from the [Input Condition depending on File window](#) using this PF key:

PF Key	Name	Function
PF9	Delete	Delete input condition depending on file.

Input Condition: Mailbox

Mailboxes are defined to the system and assigned to user IDs by using the **Mailbox Definition** function of the **System Administrator Services**. For more information on defining mailboxes, see *Mailbox Definition* in the *Administration* documentation.

For more information on how mailboxes can be used, see [Working with Mailboxes](#).

- [Using Mailboxes with Input Conditions](#)
- [Defining an Input Condition of the Type Mailbox](#)
- [Special PF Keys: Input Condition Mailbox](#)

Using Mailboxes with Input Conditions

Each logical condition can be assigned to a mailbox.

- If the condition is the only one pending (unfulfilled) and is therefore delaying the start of the subsequent job, a message is automatically sent to the mailbox.
- If an input condition is dependent on manual action(s), a message is sent to a mailbox that prompts a user to confirm completion of the action(s).

Each user linked to this mailbox sees this prompt whenever the **Main Menu** or the **Active Jobs** screen is invoked. A user can confirm the prompt by setting or resetting the requested condition. The condition will be set only if a user confirms the condition setting in the mailbox list.

A user can be associated with up to ten mailboxes.

Defining an Input Condition of the Type Mailbox

➤ To send a user prompt to a mailbox for an input condition that is not satisfied during network execution

- 1 In the **Master Input Condition** window, enter any character in the **Mailbox** field under the **Depending on** heading.
- 2 Press ENTER.

An **Input Condition: User Interaction** window similar to the example below opens:

```

16-09-20          ***** Entire Operations *****          15:42:21
+-----+-----+-----+-----+-----+-----+-----+-----+
Owner E !
!
!           Master Input Condition Addition
!
-----+-----+-----+-----+-----+-----+-----+-----+
C Condi !      Owner  ==> EXAMPLE__
_ E60-J !      Network ==> B60-FLOW__   Version ==>
!      Condi +-----+-----+-----+-----+-----+-----+
!      Referen !
+--- !      !      Input Condition: User Interaction
!      !      Usage ( !
!      !      Must !      Condition ==> COND-1
!      !      Activat !      Run ==>
!      C !
!      !      Dependi !      If the condition is not satisfied
!      !      Exit !      during the network execution, it
** !      C !      File !      is to be prompted at a mailbox.
D !      !      User !
!      !      Job V !
!      !      !      Mailbox ==> _____
Co !      !      -PF1---P !      F12--
!      P !      Help A !      PF1 Help PF3 End PF5 Save PF9 Delete
+---+-----+-----+-----+-----+-----+-----+-----+
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Add End ACond Save Up Down Menu
  
```

- 3 In the **Mailbox** field, enter the name of the mailbox to which the prompt is to be sent and press ENTER.

Or:

Enter an asterisk (*) to select a name from a list of available mailboxes.

A **Mailbox Selection** window opens with a list of mailboxes defined to Entire Operations.

Type any character in the input field next to the required mailbox and press ENTER.

The selected mailbox name appears in the **Mailbox** field.

- 4 Press PF5 (Save) to save your entry.
- 5 Press PF3 (End) to return to the **Master Input Condition** window.

Special PF Keys: Input Condition Mailbox

You can perform the following function from the **Input Condition: User Interaction window** using this PF key:

PF Key	Name	Function
PF9	Delete	Delete input condition depending on mailbox.

Input Condition: Symbol Value

An input condition can be dependent on a comparison with the contents of a symbol in a symbol table. The Monitor checks the value of the symbol on the job's **execution node** until the condition is satisfied.

This section covers the following topics:

- [Defining an Input Condition of the Type Symbol Value](#)
- [Fields: Input Condition Symbol Value](#)
- [Nested Symbol Evaluation](#)
- [Special PF Keys: Input Condition Symbol Value](#)

Defining an Input Condition of the Type Symbol Value

➤ To define an input condition that depends on a symbol value

- 1 In the **Master Input Condition window**, enter any character in the **Symbol Value** field under the **Depending on** heading.
- 2 Press ENTER.

An **Input Condition depending on Symbol Value** window similar to the example below opens:

```

16-09-20          ***** Entire Operations *****          18:07:07
+-----+-----+
Owner E !                                     !
!                                     Master Input Condition Addition !
----- !                                     ! -----
C Condi !      Owner   ==> EXAMPLE____ ! it
_ E60-J !      Network ==> B60-FLOW__  Version ==> !
_ COND- +-----+-----+
!                                     ! !
+--- !      Input Condition depending on Symbol Value ! !
! !                                     ! !
! !      Condition ==> COND-2 ! !
! !                                     ! !
! C !      Condition will be set to true, if Symbol ! !
! !      _____ ! !
! !      in Symbol Table _____ Version _____ ! !
** ! C !      at Position __ in Length __ with Format _ ! !
D ! !      is __ ! !
! !      _____ ! !
! !      _____ ! !
Co ! !                                     ! !
! P !      PF1 Help PF3 End PF5 Save          PF9 Delete ! !
+---+-----+-----+-----+-----+-----+-----+-----+-----+
!                                     defined ==> no !
! PF1 Help PF3 End PF5 Save PF9 Delete !
+-----+-----+-----+-----+-----+-----+-----+-----+
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Add End ACond Save          Up Down Menu

```

In this window you can enter the symbol to be compared and further parameters to specify the symbol.

- When finished, press PF3 (End) to save data and return to the **Master Input Condition** window.

Fields: Input Condition Symbol Value

The input fields of the **Input Condition depending on Symbol Value window** are described in the following table:

Field	Description
Condition will be set to true, if Symbol	Is a valid symbol name.
	Predefined symbols may be used too here.
	The value of this symbol, or a part of it, is to be compared with the given value.

Field	Description										
in Symbol Table	<p>Enter the name of a valid symbol table.</p> <p>If you leave this field blank, symbol search will start with the active symbol table of the job. Otherwise, the active symbol table with this name will be searched instead of the symbol table of the job. If the symbol is not found there or in the calling symbol tables, the owner's symbol master table with this name will be searched too.</p> <p>See also Symbol Table Types and Symbol Search Order in the section <i>Symbol Table and Symbol Maintenance</i>.</p>										
Version	<p>Symbol Table Version.</p> <p>By using a wildcard, you can select a version.</p> <p>Possible version names:</p> <table border="1"> <tr> <td>(current)</td> <td>Current version at determination date.</td> </tr> <tr> <td>(nv)</td> <td>Same version as network version.</td> </tr> <tr> <td>(svn)</td> <td>Symbol table version of network.</td> </tr> <tr> <td>(svj)</td> <td>Symbol table version of job.</td> </tr> <tr> <td>(unnamed)</td> <td>Unnamed version (without name).</td> </tr> </table>	(current)	Current version at determination date.	(nv)	Same version as network version.	(svn)	Symbol table version of network.	(svj)	Symbol table version of job.	(unnamed)	Unnamed version (without name).
(current)	Current version at determination date.										
(nv)	Same version as network version.										
(svn)	Symbol table version of network.										
(svj)	Symbol table version of job.										
(unnamed)	Unnamed version (without name).										
at Position	<p>Enter the position of symbol substring to be checked.</p> <p>Possible values: 1 through 80. (Checked only if Format is set to A.)</p>										
in Length	<p>Enter length of symbol substring to be checked.</p> <p>Possible values: 1 through 80. (Checked only if Format is set to A.)</p>										
with Format	<p>Specify the format in which the symbol substring is to be checked against the comparison string.</p> <p>Possible values:</p> <table border="1"> <tr> <td>A</td> <td>Alphanumeric.</td> </tr> <tr> <td>D</td> <td>Date in the format YYYYMMDD.</td> </tr> <tr> <td></td> <td>See also Date and Time Formats.</td> </tr> <tr> <td>N</td> <td>Numeric (zoned).</td> </tr> </table>	A	Alphanumeric.	D	Date in the format YYYYMMDD.		See also Date and Time Formats .	N	Numeric (zoned).		
A	Alphanumeric.										
D	Date in the format YYYYMMDD.										
	See also Date and Time Formats .										
N	Numeric (zoned).										
is	<p>This is the comparison operator.</p> <p>Specify a logical operator for the comparison of the defined symbol against the comparison string (see below).</p> <p>Possible values:</p> <table border="1"> <tr> <td>= or EQ</td> <td>Code is equal to specified value.</td> </tr> <tr> <td>>= or GE</td> <td>Code is greater than or equal to specified value.</td> </tr> <tr> <td>> or GT</td> <td>Code is greater than specified value.</td> </tr> <tr> <td><= or LE</td> <td>Code is lower than or equal to specified value.</td> </tr> </table>	= or EQ	Code is equal to specified value.	>= or GE	Code is greater than or equal to specified value.	> or GT	Code is greater than specified value.	<= or LE	Code is lower than or equal to specified value.		
= or EQ	Code is equal to specified value.										
>= or GE	Code is greater than or equal to specified value.										
> or GT	Code is greater than specified value.										
<= or LE	Code is lower than or equal to specified value.										

Field	Description
	< or LT Code is lower than specified value.
	<> or NE Code is different from specified value.
<i>comparison string</i>	<p>Comparison string.</p> <p>In the input lines below is, enter the string or field to be compared with the symbol (substring). The strings are compared in the defined Format.</p> <p>Symbol replacement is possible in this field.</p> <ul style="list-style-type: none"> ■ If an activation escape character is used, the replacement is performed once during activation. A symbol replacement error will be treated as permanent error in this case. ■ If a submission escape character is used, the replacement is performed directly before each prerequisite check. This causes more system overhead. A symbol replacement error will be treated as temporary error in this case.

Nested Symbol Evaluation

The symbol value may contain other (nested) symbols, prefixed by both activation escape character and submit escape character.

Symbols prefixed by the activation escape character are evaluated only once, at job activation.

Symbols prefixed by the submit escape character are evaluated at each prerequisite check of the active job.

Special PF Keys: Input Condition Symbol Value

You can perform the following function from the [Input Condition depending on Symbol Value window](#) using this PF key:

PF Key	Name	Function
PF9	Delete	Delete input condition dependent on symbol value.

Input Condition: User Switch - BS2000

An input condition value can be dependent on the existence or non-existence of a user switch. The Monitor checks for the user switch on the job's **execution node** until the condition is satisfied.

This section covers the following topics:

- [Defining an Input Condition of the Type User Switch](#)
- [Fields: Input Condition User Switch](#)

- [Special PF Keys: Input Condition User Switch](#)

Defining an Input Condition of the Type User Switch

➤ To define an input condition that depends on a user switch

- 1 In the [Master Input Condition window](#), enter any character in the **User Switch** field under the **Depending on** heading.
- 2 Press ENTER.

An **Input Condition depending on User Switch** window similar to the following opens:

```

16-09-20          ***** Entire Operations *****          18:07:07
+-----+-----+
Owner E ! EOR0037 - Please enter the Condition Name          !
          !           Master Input Condition Addition          !
----- !           ! -----
C Condi !      Owner   ==> EXAMPLE__          ! it
_ E60-J !      Network ==> B60-FLOW__   Version ==>          !
_ COND- !      Condition ==> COND-2_____          !
          !      Reference ==> _____          !
          !      +-----+-----+-----+-----+          !
          !      Usag !           !           !           !
          !      Mu ! Input Condition depending on User Switch ! : _ !
          !      Acti !           !           !           !
          !      ! Condition ==> COND-2          !           !
          !      Depe !           Run ==>          !           !
          !      Ex !           !           !           !
          !      Fi ! Condition will be set to true, if          ! > _ ! *****
D Delet !      Us !           User Switch ==> _0          ! > _ !
          !      Jo ! of BS2000 User ID ==> BSU_____          !           !
          !      !           exists ==> Y           (Y/N)          !           !
Command ! -PF1- !           !           !           !           ! 2-- !
          !      Help ! PF1 Help PF3 End PF5 Save PF9 Delete          ! u !
          +-----+-----+-----+-----+
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Add  End  ACond Save          Up  Down          Menu
  
```

In this window you can enter a user switch and BS2000 user ID.

The input fields available are described in [Fields: Input Condition User Switch](#).

- 3 When finished, press PF3 (End) to save data and return to the **Master Input Condition** window.

Fields: Input Condition User Switch

The input fields in the [Input Condition depending on User Switch window](#) are described in the following table:

Field	Description	
User Switch	Number of a user switch.	
of BS2000 User ID	BS2000 user ID to which the specified user switch belongs.	
exists	Possible values:	
	Y	The user switch must exist as a prerequisite to job submission.
	N	The user switch must not exist as a prerequisite.

Special PF Keys: Input Condition User Switch

You can perform the following function from the [Input Condition depending on User Switch window](#) using this PF key:

PF Key	Name	Function
PF9	Delete	Delete input condition depending on user switch.

Input Condition: Job Variable - BS2000

An input condition can be dependent on a comparison with the contents of a BS2000 job variable. The Monitor checks for the job variable on the job's **execution node** until the condition is satisfied.

This section covers the following topics:

- [Defining an Input Condition of the Type Job Variable](#)
- [Fields: Input Condition Job Variable \(BS2000\)](#)
- [Special PF Keys: Input Condition Job Variable \(BS2000\)](#)

- Using Symbols

Defining an Input Condition of the Type Job Variable

➤ To define an input condition that depends on a job variable

- 1 In the **Master Input Condition window**, enter any character in the **Job Variable** field under the **Depending on** heading.
- 2 Press ENTER.

An **Input Condition depending on Job Variable Contents** window similar to the following opens:

```

16-09-20          ***** Entire Operations *****          18:07:07
+-----+-----+
Owner E !
!
!           Master Input Condition Addition
-----!
C Condi !      Owner   ==> EXAMPLE___
_ E60-J !      Network ==> B60-FLOW__   Version ==>
_ COND- +-----+
!
+--- !      Input Condition depending on Job Variable Contents
!
!      !      Condition ==> COND-2
!      !      Run ==>
! C !
!      !      Condition will be set to true, if Job Variable
!      !      _____
** ! C !      at Position ___ in Length ___ with Format _
D !      !      is ___
!
!      !      _____
!      !      _____
Co !      !      _____
! P !      _____
+--- !
!
!           (Optional) Read Password ==>
!
!           defined ==> no
! PF1 Help PF3 End   PF5 Save   PF9 Delete
+-----+-----+

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Add   End   ACond Save           Up   Down                               Menu
    
```

In this window you can enter a job variable and further parameters to specify the input condition.

The input fields are described in *Fields: Input Condition Job Variable (BS2000)*.

3 When finished, press PF3 (End) to save data and return to the **Master Input Condition** window.

Fields: Input Condition Job Variable (BS2000)

The input fields provided in the **Input Condition depending on Job Variable Contents window** are described in the following table:

Field	Description												
Condition will be set to true, if Job Variable	<p>Enter the name of a valid BS2000 job variable.</p> <ul style="list-style-type: none"> ■ If the specified job variable does not exist, a content comparison will be done by Entire Operations, when the Monitor detects that the job variable has been created. ■ If a job variable does not exist, a job cannot be declared “dummy due to condition”. Instead, the Entire Operations Monitor waits until the job variable exists and then performs the check. ■ If the job variable is specified without an explicit user ID, the job default BS2000 user ID is used as prefix. ■ Symbol replacement: see Using Symbols. 												
at Position	Enter position of job variable substring to be checked. Possible values: 1 through 253.												
in Length	Enter the length of job variable substring to be checked. Possible values: 1 through 253.												
with Format	<p>Specify the format in which the job variable substring is to be checked against the comparison string.</p> <p>Possible values:</p> <table border="1"> <tr> <td>A</td> <td>Alphanumeric.</td> </tr> <tr> <td>N</td> <td>Numeric (zoned).</td> </tr> </table>	A	Alphanumeric.	N	Numeric (zoned).								
A	Alphanumeric.												
N	Numeric (zoned).												
is	<p>This is the comparison operator.</p> <p>Specify a logical operator for the comparison of the defined job variable substring against the comparison string (see below).</p> <p>Possible values:</p> <table border="1"> <tr> <td>= or EQ</td> <td>Code is equal to specified value.</td> </tr> <tr> <td>>= or GE</td> <td>Code is greater than or equal to specified value.</td> </tr> <tr> <td>> or GT</td> <td>Code is greater than specified value.</td> </tr> <tr> <td><= or LE</td> <td>Code is lower than or equal to specified value.</td> </tr> <tr> <td>< or LT</td> <td>Code is lower than specified value.</td> </tr> <tr> <td><> or NE</td> <td>Code is different from specified value.</td> </tr> </table>	= or EQ	Code is equal to specified value.	>= or GE	Code is greater than or equal to specified value.	> or GT	Code is greater than specified value.	<= or LE	Code is lower than or equal to specified value.	< or LT	Code is lower than specified value.	<> or NE	Code is different from specified value.
= or EQ	Code is equal to specified value.												
>= or GE	Code is greater than or equal to specified value.												
> or GT	Code is greater than specified value.												
<= or LE	Code is lower than or equal to specified value.												
< or LT	Code is lower than specified value.												
<> or NE	Code is different from specified value.												
<i>comparison string</i>	<p>In the input lines below is, enter the string or field to be compared with the job variable substring.</p> <p>The strings are compared in the defined Format. The content of this field is compared with the substring of the job variable, or it is inserted into the substring of the job variable.</p>												

Field	Description
	The content is treated as blank if ' ' (2 single quotes, no space) or ' ' (single quote, space, single quote) is defined. The comparison is made in the defined format. Symbol replacement: see Using Symbols .
(Optional) Read Password	(Optional) If the job variable is read password-protected, specify the password here.

Special PF Keys: Input Condition Job Variable (BS2000)

You can perform the following function from the [Input Condition depending on Job Variable Contents window](#) using this PF key:

PF Key	Name	Function
PF9	Delete	Delete input condition dependent on job variable contents.

Using Symbols

Resolving symbols in the job variable name produces the same behavior as resolving symbols in the job variable value:

- If the activation escape character is used:
 - The symbol is replaced once during the job activation.
 - The active job variable name is the resolved string.
- If the submission escape character is used:
 - The symbol is resolved during each performed prerequisite check.
 - This allows symbol setting shortly before the usage.

 **Note:** The submission escape character option causes more system overhead.

Listing Jobs Linked to an Input Condition

You can list jobs that also use a selected input condition as an input condition, or that use this input condition as an output condition.

➤ To list jobs linked to an input condition

- 1 On the [Input Conditions Maintenance screen](#), type W in the line command input field of the required condition.

2 Press ENTER.

A **Condition** window opens similar to the example below opens:

```

17-04-14          ***** Entire Operations *****          11:53:35
                    Input Conditions Maintenance
Owner EXAMPLE    Network E60-FLOW    Version          Job JOB-013
-----
C Condition      Reference Type      Sched.Dep.  U ex. Library  Exit
w E60-J012-0    RUN          true          Y
+-----+
!
!                      Condition: E60-J012-0
! Output Condition of          Input Condition for
! -----+-----
! Owner      Network  Job      ! Owner      Network  Job      !
! EXAMPLE    E60-FLOW  JOB-012 ! EXAMPLE    E60-FLOW  JOB-013 !
! EXAMPLE    E60-FLOW  JOB-013 !
!
!
!
! Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10-----
!                      End                      Up      Down
+-----+
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Add  End  ACond Save          Up      Down          Menu
    
```

The window displays the name of the selected condition and two lists of jobs:

- one list shows the jobs which use the condition as input condition;
- one list shows the jobs which use the condition as output condition.

The jobs are listed according to owner, network and job name.

3 Press PF3 (End) to close the window.



Note: For an active input condition, a similar window opens: see the **Condition** window described in the section *Active Job Networks*.

Deleting an Input Condition Definition

➤ To delete an input condition definition

- 1 On the **Input Conditions Maintenance screen**, type D in the line command input field next to the condition you want to delete.
- 2 Press ENTER.

A window prompts you to confirm the deletion.
- 3 Enter Y and press ENTER to delete the input condition definition.

Displaying Currently Active Conditions

➤ To display currently active conditions

- Press PF4 (ACond) on the **Input Conditions Maintenance** screen.

An **Active Conditions** screen appears described in the section *Listing Active Conditions* appears.

For further information, refer to *Listing Active Conditions* in the section *Active Job Networks*.

28

Handling Prerequisite Resources for a Job

- Listing Prerequisite Resources Defined for a Job 308
- Viewing the Usage of a Prerequisite Resource 309
- Displaying, Modifying and Adding a Prerequisite Resource Definition 309
- Deleting a Prerequisite Resource Definition 310
- Columns and Fields: Prerequisite Resource Definitions 311

The functions described in this section are used to list and define single or multiple resources for a job that are a prerequisite for submitting the job.

Before you can define a resource as a prerequisite for a job, it must be defined as resource master to Entire Operations. You define resources to Entire Operations by using the **Resources** maintenance functions described in the *Administration* documentation.

Related Topics:

- Maintenance of a resource master: *Resources* in the *Administration* documentation
- Basic use and summary of resource features: [Resources](#) in the section *System Overview*

Listing Prerequisite Resources Defined for a Job

➤ **To list prerequisite resources specified for a job**

- 1 On the **Job Maintenance screen**, enter L in the line command input field next to the required job.
- 2 Press ENTER.

A **Prerequisite Resources (Master)** screen similar to the example below opens:

```

17-04-05          ***** Entire Operations *****          17:50:36
                   Prerequisite Resources (Master)
-----
Owner EXAMPLE   Network B60-FLOW   Version                Job JOB-06
Run from        to                required
Cmd  Run Job    Resource          T  Quantity  D  DNO   allocated
-    -          HUGO              U    5.00    J  Y
-    -          PARA-1           R   10.00   N  Y
-    -          WILLI            U   20.00   K  Y

***** Bottom of Data *****
D Delete    M Modify   R Master Definition  W Where Used

Enter--PF1----PF2----PF3----PF5-----PF7----PF8-----
      Help   Add     End     Save     Up     Down
    
```

The screen lists all resources specified for the job as prerequisites. The columns contained in the window are described in *Columns and Fields: Prerequisite Resource Definitions*.

Viewing the Usage of a Prerequisite Resource

➤ To find out where a predefined resource is used

- 1 In the **Cmd** column of the **Prerequisite Resources screen**, enter the line command `W` (Where Used) next to the required resource.
- 2 Press `ENTER`.

A **Resource defined in Jobs** window opens with a list of all owners, networks and jobs that use the selected resource.

If invoked for a master job definition: Shows the usage of this resource as a prerequisite in all master job definitions.

If invoked for an active job definition: Shows the current usage of this resource by active jobs.

The fields and columns contained in the window correspond are described in *Field and Columns: Resource Defined in Jobs* in the section *Resources* in the *Administration* documentation.

Displaying, Modifying and Adding a Prerequisite Resource Definition

➤ To display or modify the prerequisite resource definition of a job

- 1 In the **Prerequisite Resources screen**, type `M` (Modify) in the line command input field next to the resource you want to view or modify.
- 2 Press `ENTER`.

A **Prerequisite Resource Definition (Master)** window similar to the example below opens:

```

17-04-06          ***** Entire Operations *****          17:55:11
                   Prerequisite Resources (Master)
-----
Owner EXAMPLE    Network B60-FLOW   Version                               Job JOB-06
Run from         t +-----+-----+-----+-----+-----+-----+
Cmd  Run Job     !                               !
m                                         Prerequisite Resource Definition (Master) !
-                                         !
-                                         Resource ==> HUG0_____ !
-                                         Required Quantity ==> 5.00_____ !
!                                         !
!                                         Deallocation Mode ==> J !
!                                         Deallocate if Job not ok ==> Y !
!                                         Allocated at ==> !
!                                         !
! Enter-PF1-----PF3-----PF5----- !
! Help           End           Save           !
+-----+-----+-----+-----+-----+
***** Bottom of Data *****
D Delete      M Modify      R Master Definition      W Where Used

Enter--PF1----PF2----PF3----PF5-----PF7----PF8-----
      Help   Add    End    Save      Up    Down

```

The fields are described in *Columns and Fields: Prerequisite Resource Definitions*.

➤ To add a prerequisite resource definition for a job

- In the **Prerequisite Resources list screen**, press PF2 (Add).

A **Prerequisite Resource Definition window** opens.

The input fields and options available in the window are described in *Columns and Fields: Prerequisite Resource Definitions*.

Deleting a Prerequisite Resource Definition

➤ To delete a prerequisite resource

- 1 In the **Prerequisite Resources screen**, type D (Delete) in the line command input field next to the resource you want to remove.
- 2 Press ENTER.

The selected resource is deleted for this job master.

Columns and Fields: Prerequisite Resource Definitions

The following table explains the columns contained in the [Prerequisite Resources list screen](#) and the corresponding input fields (if available) of the [Prerequisite Resource Definition window](#).

Column/Field/Option	Description
Cmd	One-character line command input field. Possible options:
	D Delete a prerequisite resource definition for this job.
	M Modify a prerequisite resource definition for this job.
	R Display the master resource definition.
W View where a prerequisite resource definition is used.	
Resource (name)	Name of the resource as defined in Entire Operations and listed on the Resources screen (see <i>Resources</i> in the <i>Administration</i> documentation). Input field: You can use a wildcard (*) to select a name from a list of available resources.
T	Resource type. For details, see the description of the Type field in <i>Fields: Resource Definition</i> in the section <i>Resources</i> (<i>Administration</i> documentation).
(Required) Quantity	Resource quantity required for job execution. Input field: Enter the resource quantity required for job execution. Entire Operations does not submit the job until this amount of resource is available.
D	Deallocation mode of this prerequisite resource: see Resource Deallocation Modes .
Deallocation Mode	
DNO	Applies to deallocation mode J only. Deallocate if job is not ok:
	Y The resource will always be deallocated after job termination, even if the job ended not ok.
Deallocate if Job not ok	N The resource will not be deallocated if the job ended not ok. (However, the job will still be deallocated at network termination.)
	Input option: Enter Y or N as required.
Allocated (at)	Applies to active prerequisite resources only.
	Date and time when the resource was allocated for the current job.

Column/Field/Option	Description
	<p>If the resource has not yet been allocated or if it has already been deallocated, this field is empty.</p> <p>See also Date and Time Formats.</p>

Resource Deallocation Modes



Note: Resource allocations will be deallocated if the retention period for an active condition is reached.

Mode/Input Value	Explanation
J	<p>Release resource at job termination (default).</p> <p>The resource is released when the allocating job terminates.</p>
N	<p>Release resource at network termination.</p> <p>The resource will be released when the Entire Operations Monitor detects that all jobs of a job network are terminated.</p> <p>The resource will be released also if a currently executing network is being deactivated.</p> <p>Note: To override the automatic detection of network ended ok, you must set the reserved condition NET-END-OK at least once in your network.</p>
E	<p>Release resource at network termination, but release earlier after an erroneous job.</p> <p>The resource will be kept until network termination. If any job ends not ok, it will be released immediately after this job's termination.</p>
K	<p>Keep resource until manual release.</p> <p>The resource will not be released automatically at job or network termination.</p> <p>If the retention period for active conditions is reached, the resource will be released automatically.</p>

The following additional rules apply:

- If an active network or job is deactivated, all the resources allocated by it will be released too. This will be performed regardless of the defined deallocation mode. This means that even resources with deallocation mode K (keep) will be released in such cases.
- Master resource amounts can be modified by API calls. Note that this is possible only if the amount is not determined by a resource master determination exit.
- Resources can be allocated for a job by an [API call](#). These additionally allocated resources will be handled in the same way as defined prerequisite resources.
- Resources can be deallocated by [API calls](#). Preliminary releases of resources are allowed for deallocation modes.

- All resource allocations and deallocations will be logged.

For more information, see [Handling of Resource Allocations \(NOPURE1N\)](#) in the section *List of Available Entire Operations API Routines*.

VII

■ 29 End-of-Job Checking and Actions	317
■ 30 End-of-Job Checking and Actions - Overview	319
■ 31 End-of-Job Checking and Actions Screen	323
■ 32 Adding Event Definition for z/OS or z/VSE Job	329
■ 33 Modifying Event Definition for z/OS or z/VSE Job	335
■ 34 Adding Event Definition for BS2000 Job	341
■ 35 Modifying Event Definition for BS2000 Job	345
■ 36 Adding Event Definition for UNIX or Windows Job	355
■ 37 Modifying Event Definition for UNIX or Windows Job	359
■ 38 Deleting an Event Definition	365
■ 39 EOJ Checking Defaults for Various Operating Systems	367
■ 40 Creating Online Documentation for Events	371
■ 41 Adding Output Condition Definitions	373
■ 42 Modifying Output Condition Definitions	379
■ 43 Define Symbol Modification or Setting	381
■ 44 Displaying Output Condition Use	385
■ 45 Editing End-of-Job Checking or Action User Exits	387
■ 46 Defining Activation of Jobs or Job Networks	391
■ 47 Defining Recovery Action	395
■ 48 SYSOUT Actions	401
■ 49 Message Sending	405
■ 50 Defining End-of-Job Action User Exit	413
■ 51 End-of-Job Action: Release Resource	417
■ 52 Defining Other Actions	419

29

End-of-Job Checking and Actions

[End-of-Job Checking and Actions - Overview](#)
[End-of-Job Checking and Actions Screen](#)
[Adding Event Definition for z/OS or z/VSE Job](#)
[Modifying Event Definition for z/OS or z/VSE Job](#)
[Adding Event Definition for BS2000 Job](#)
[Modifying Event Definition for BS2000 Job](#)
[Adding Event Definition for UNIX or Windows Job](#)
[Modifying Event Definition for UNIX or Windows Job](#)
[Deleting an Event Definition](#)
[EOJ Checking Defaults for Various Operating Systems](#)
[Creating Online Documentation for Events](#)
[Adding Output Condition Definitions](#)
[Modifying Output Condition Definitions](#)
[Define Symbol Modification or Setting](#)
[Displaying Output Condition Use](#)
[Editing End-of-Job Checking or Action User Exits](#)
[Defining Activation of Jobs or Job Networks](#)
[Defining Recovery Action](#)
[SYSOUT Actions](#)
[Message Sending](#)
[Defining End-of-Job Action User Exit](#)
[End-of-Job Action: Release Resource](#)
[Defining Other Actions](#)

Related Topics:

- *End-of-Job Checking and End-of-Job Actions in Concepts and Facilities.*

■ *Passing Files to Entire Output Management*

30

End-of-Job Checking and Actions - Overview

- List of End-of-Job Checking and Actions 320
- Kinds of End-of-Job Actions 321
- End-of-Job Checks for Various Operating Systems 322

List of End-of-Job Checking and Actions

- End-of-Job Actions with Execution as a Temporary Dummy Job
- End-of-Job Checking and Actions Facility
- Adding Event Definition for z/OS or z/VSE Job
- Modifying Event Definition for z/OS or z/VSE Job
- Adding Event Definition for BS2000 Job
- Modifying Event Definition for BS2000 Job
- Adding Event Definition for UNIX or Windows Job
- Modifying Event Definition for UNIX or Windows Job
- Deleting an Event Definition
- End-of-Job Checking Defaults for various Operating Systems
- Creating Online Documentation for Events
- Adding Output Condition Definitions
- Modifying Output Condition Definitions
- Modifying or Setting Master or Active Symbols
- Displaying Output Condition Use
- Editing End-of-Job Checking or Action User Exits
- Defining Activation of Jobs or Job Networks
- Defining Recovery Action
- SYSOUT Actions
- Message Sending
- Defining End-of-Job Action User Exit
- Release Resource
- **Defining Other Actions**

See also: [Passing Files to Entire Output Management](#)

Defining End-of-Job checking and actions means instructing Entire Operations what actions to trigger after a job has terminated. The triggered action is referred to as an *End-of-Job action*.

The End-of-Job action depends on the status of the whole job or any other event which can be checked by Entire Operations on termination. The status of the job on termination is determined by the occurrence of certain events during job execution which Entire Operations can check.

Before Entire Operations can check for an event, it must be defined to the system, together with instructions as to the action to be triggered. If no events are specified, Entire Operations performs End-of-Job checking automatically using system defaults.

Events are checked on a *check ok/check not ok* basis. End-of-Job actions can be triggered by:

- the result of a single event check (for example: job step level in z/OS);
- the overall result of the End-of-Job check, which always causes either the event *job ok* or *job not ok* to occur.

Both of these event definitions are always predefined for every job.

In the latter case, you can define two sets of actions:

- one set is performed if all checks finished *ok* (i.e. *job ok*);
- the other is performed if at least one check finished *not ok* (i.e. *job not ok*).

All actions are performed after the job has terminated.

Kinds of End-of-Job Actions

Possible End-of-Job actions are:

- **Set/reset output conditions;**
- **Set/reset/modify a job variable (BS2000 only);**
- **Execute an End-of-Job action user exit;**
- **Define activation of jobs or job networks;**
- **Start a recovery;**
- **Handle job SYSOUT;**
- **Pass output files to Entire Output Management;**
- **Send user-defined messages;**

Only one action of the same type can be defined for an event (except in the case of output conditions). If you need several actions of the same type for the event **job ok** or **job not ok**, you can define these by adding **A type events**.

End-of-Job Checks for Various Operating Systems

The following table provides an overview of the availability of various End-of-Job checks for the supported operating systems and environments:

End-of-Job Check	z/OS, MSP	z/VSE	BS2000	UNIX	Windows	SAP
Termination Codes	*	*				
Job Variables			*			
User Exit	*	*	*	*	*	
String Search	*	*	*	*	*	*
User Switch			*			
Exit Code				*	*	

An asterisk * indicates that the End-of-Job check is available for the operating system. These checks are described later in this section.

31

End-of-Job Checking and Actions Screen

■ Invoking End-of-Job Checking and Actions	324
■ Columns: End-of-Job Checking and Actions	325
■ Line Commands	326

- [Invoking End-of-Job Checking and Actions](#)
- [Columns: End-of-Job Checking and Actions](#)
- [Line Commands](#)

Invoking End-of-Job Checking and Actions

➤ To invoke the End-of-Job Checking and Actions facility, display the Job Maintenance screen.

- 1 Type 0 in the line command field of the selected job.
- 2 Press ENTER. The **End-of-Job Checking + Actions** screen appears:

```

24.10.08          ***** Entire Operations *****          13:04:16
Owner   SN          End-of-Job Checking + Actions   MVS/ESA   Job IEFBR14 ←
Network A-1                                     Run      Date          ←
-----
C Action      Step      will be checked for          means OA
-            ANYSTEP   Condition Code = C0004      --      ←
-                                                    ←
-                                                    ←
-                                                    ←
-                                                    ←
-                                                    ←
-                                                    ←
-                                                    ←
-                                                    ←
-                                                    ←
***** Bottom of Data *****
-          ST X    All Checks ok                               ←
-          S      Any Check not ok                          ←
-----
A Activat. C Cond.  D Delete  E Ed Usr Rtn  J JV  L Re1 Res  M Modify  O Other
P Descr.  R Recov.  S SYSOUT T Output Mgmt  U User Msg X Action Rtn Y Symbol Set
Command => _____ ←
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Add   End       Save      Up    Down          Menu

```

Columns: End-of-Job Checking and Actions

The columns on the [End-of-Job Checking and Actions screen](#) are described in the following table:

Column	Description																		
C	One-character line command input field. For an explanation of possible line commands, see Line Commands: End-of-Job Checking and Actions .																		
Action	Summary of defined actions for the event. The character(s) here correspond to the line commands at the bottom of the screen.																		
Step	Step name in JCL (ANYSTEP means all steps) (z/OS only).																		
will be checked for	Event to be checked for (condition code, system code, user abend code, System message, etc.)																		
means	Event status after check (All checks ok / Any check not ok).																		
OA	<p>This column only contains entries if an event occurred during job execution.</p> <p>The entries under O indicate the status of the event check, the entries under A indicate whether the action defined for the event was performed:</p> <table border="1"> <thead> <tr> <th>Entry</th> <th>O</th> <th>A</th> </tr> </thead> <tbody> <tr> <td><i>blank</i></td> <td>Event was not checked.</td> <td>No action performed.</td> </tr> <tr> <td>-</td> <td>Event did not occur.</td> <td>n/a</td> </tr> <tr> <td>+</td> <td>Event occurred.</td> <td>Action performed.</td> </tr> <tr> <td>E</td> <td>Error occurred during event check.</td> <td>One or more actions performed erroneously.</td> </tr> <tr> <td>M</td> <td>Event definition was modified.</td> <td>Action definition was modified.</td> </tr> </tbody> </table>	Entry	O	A	<i>blank</i>	Event was not checked.	No action performed.	-	Event did not occur.	n/a	+	Event occurred.	Action performed.	E	Error occurred during event check.	One or more actions performed erroneously.	M	Event definition was modified.	Action definition was modified.
Entry	O	A																	
<i>blank</i>	Event was not checked.	No action performed.																	
-	Event did not occur.	n/a																	
+	Event occurred.	Action performed.																	
E	Error occurred during event check.	One or more actions performed erroneously.																	
M	Event definition was modified.	Action definition was modified.																	

Line Commands

You can use the line commands to maintain the events and the actions to be triggered. All actions can be defined for an event (job step) or at the job level.

> To define an action at the job level

- Enter the required line command in the appropriate input field preceding the All checks ok or Any check not ok field.

The following line commands are available:

Cmd	Description
A	Activate a job or network if selected event occurs.
C	Define or modify the condition to be set or reset when the selected event occurs.
D	Delete an event definition.
E	Edit the user exit to be executed if the selected event occurs.
J	Add or modify job variable (BS2000 only).

Cmd	Description
L	Release resource.
M	Modify an event definition. z/OS/, z/VSE BS2000 UNIX, Windows
O	Other actions.
P	Description. Write a text description of an event.
R	Define a recovery action to be executed if the selected event occurs (normally job failure).
S	Define job SYSOUT handling on job completion, if the selected event occurs.
T	Pass files to Entire Output Management.
U	Define message to be sent if the selected event occurs.
X	Define a Natural user exit to be executed if the selected event occurs. This is called an <i>End-of-Job Action User Exit</i> .
Y	Define symbol modification or setting.

Note that the D (DELETE), E (EDIT) and M (MODIFY) line commands are event maintenance commands, and therefore available at the event level only. All other line commands are also available at the job level according to All checks ok and Any check not ok.

Use the following PF keys to maintain event definitions and define actions to be triggered at job termination.

- Press PF2 (Add) to save the definition and clear the window to add another event. You can define any number of events for any one job.
- Press PF5 (Save) to save the event definition(s).
- Press PF3 (End) to close the window and return to the *End-of-Job Checking and Actions Screen*. The new events are listed on this screen.

32

Adding Event Definition for z/OS or z/VSE Job

- Add/Modify Event Definition 330
- Field Descriptions: Add/Modify Event Definition 330

See also *z/OS Defaults for Event Checking* and *z/VSE Defaults for Event Checking*.

Add/Modify Event Definition

> To add an event to be checked at job termination

- Press PF2 (Add) on the End-of-Job Checking and Actions screen. A window opens in which you can enter the event definition:

```

12.02.10          ***** Entire Operations *****          13:09:39
Owner   SN          End-of-Job Checking + Actions   MVS/ESA   Job A-1
Network +-----+-----+-----+-----+-----+-----+
----- !
C Action !          Add Event Definition          ! s OA
_       ! Event Type ==> _ C Termination Code          MVS/ESA !
_       !          R Exit          !
_       !          S String          A Add. Job-ok,-not-ok !
_       !          !          !
_       ! Stepname ==> _____ !
_       ! Operator ==> ___ Event ==> _____ !
_       !          !          !
_       ! Exit ==> _____ in NAT Library ==> _____ !
***** !          Exit Mode ==> _          ! ****
_ C     ! String ==> _____ missing ==> _ !
_ A     ! Select ==> _____ !
----- !          ! -----
A Activa ! Occurrence of event means OK or NOT OK ==> ___ ! ther
P Descr. !          !          ! Set
! Enter-PF1---PF2---PF3-----PF5----- !
Command !          Help Add End Save          !
Enter-PF1 +-----+-----+-----+-----+ 2---
          Help Add End          Save          Up Down          Menu
    
```

Field Descriptions: Add/Modify Event Definition

(z/OS, z/VSE)

Meaning of the input fields:

Field	Description	
Event Type	Type of event to be checked. Possible values:	
	A	Additional definition for job ok or job not ok. See Example of an A-type Event .
	C	Return code received during job execution (job step). See Example of a C-type Event .
	R	Job SYSOUT is to be checked by a user exit. See Example of an R-type Event .
	S	<p>Occurrence of a specific string in job SYSOUT . See Example of an S-type Event.</p> <p>Note:</p> <ol style="list-style-type: none"> 1. All checks of the SYSOUT file and actions, which refer to the SYSOUT file, are not executed, if no SYSOUT file exists. This case will apply, if the job is submitted as a temporary dummy job. 2. A string search in a non-existent file is handled like "string not found" in an existing file.
Step Name	(for a <i>C-type</i> event) The JCL step to be checked for the return code specified in the Operator and Event Type fields.	
	Special step names:	
	/ANYPSTEP	Event occurs if the definition is true for any step of the jobs.
	/MAXCC	Event occurs if it is true for the maximum termination code (condition code) of the job.
	Note: In z/VSE you can define STEPNAMEs for Entire Operations with the LABEL statement: /.label, where label stands for STEPNAME. The STEPNAME defined in this way is valid for the following EXEC statement.	
Operator	(for a <i>C-type</i> event) Specify a logical operator to compare the received return code with the value entered in the Event Type field. Possible values:	
	= or EQ	Code is equal to specified value.
	>= or GE	Code is greater than or equal to specified value.
	>> or GT	Code is greater than specified value.
	<= or LE	Code is lower than or equal to specified value.
	< or LT	Code is lower than specified value.
	<> or NE	Code is different from specified value.
Event	(for a <i>C-type</i> event) Specify a condition code to be compared with the return code. Possible values:	
	Cxxxx	Condition code xxxx (z/OS) Return code xxxx

Field	Description
	Sxxx System abend code xxx (z/OS only)
	Uxxxx User abend code xxxx (z/OS only)
	JNR Job not run, JCL error. Note: If this event occurs, and a JNR event definition with OK exists, this can be treated as ok. See also z/OS Defaults for Event Checking .
	JFL Job failed, JCL error. Note: If this event occurs, and a JFL event definition with OK exists, this can be treated as ok. See also z/OS Defaults for Event Checking .
	JDL Job deleted by operator.
	JIR Job execution interrupted (for example: system crash)
User Exit	(for an R-type event) Name of the end-of-job check user exit which is to run on job termination. Entire Operations will perform an Entire System Server logon to the job's execution node with the job's submit user ID prior to the execution of the exit. See also the section Editing End-of-Job User Exits .
in Natural Library	(for an R-type event) The Natural library in which the user exit resides. This library should be different from the Entire Operations system library.
Exit Mode	' ' (blank) Exit will be executed synchronously. (Default)
	'A' Exit will be executed asynchronously (in a Natural task).
String	(for an S-type event) Specify the actual string in the job SYSOUT for which Entire Operations is to check. The search string may contain symbols. Symbols with activation escape characters will be replaced at job activation time. Symbols with submit escape characters will be replaced at end-of-job checking time. Missing symbols will cause a permanent activation error or permanent end-of-job checking error.
missing	String missing: If "Y" is defined here: If the string is not found, the event will be satisfied.
Select (z/OS)	You can limit the search for strings by specifying file numbers in this field according to file type (SM, SO, JL). Enter file type followed by number, range of numbers or an asterisk * as wildcard. For example:
	SM 1 Search system messages 1.
	SM 1 5 Search system messages 1 to 5.
	SO* Search all SYSOUT files.
Select (z/VSE)	You can use these file types: LST, PUN, RDR.
Occurrence of event	Specify the event check status, if defined event occurs. Possible values:
	OK Check ok.

Field	Description
means OK or NOT OK	NO Check not ok.
	-- No effect on the job result.
	Usually actions are defined at the job level with the events All checks ok or Any check not ok. If you want to define an action of the same type two or more times for one of these events, you need additional event entries for job ok or job not ok. See Example of an A-type Event .
	Note: You can use these additional event definitions to set conditions, but it is not recommended, because they are not evaluated for the display of job dependencies (see Displaying Job Dependencies and Connecting or Disconnecting Jobs in the section Job Maintenance).

When you have finished defining the event, you can proceed in one of the following ways:

- If you defined an R-type event, you can enter E in the line command field of the event on the [End-of-Job Checking and Actions Screen](#) and press Enter to define the user exit using the Entire Operations Editor. (See the subsection [Editing End-of-Job User Exits](#).)
- Press PF2 (Add) to save the definition and clear the window to add another event. You can define any number of events for any one job.
- Press PF5 (Save) to save the event definition(s).
- Press PF3 (End) to close the window and return to the [End-of-Job Checking and Actions Screen](#). The new events are listed on this screen.

33

Modifying Event Definition for z/OS or z/VSE Job

- Modifying an Existing Event Definition 336
- Example of an A-type Event 337
- Example of a C-type Event 338
- Example of an R-type Event 339
- Example of an S-type Event 340

Modifying an Existing Event Definition

> To modify an existing event definition

- 1 Type M in the line command field of the selected event on the *End-of-Job Checking and Actions Screen*.
- 2 Press Enter.
- 3 The *Modify Event Definition* window opens with the current values for the event. You can modify the values by overtyping them.

When you are finished defining the event, you can proceed as follows:

- 4 Press PF5 (Save) to save the modification.
- 5 Press PF3 (End) to close the window and return to the *End-of-Job Checking and Actions Screen*.



Note: The modify option (M line command) is available at the event level only.

Below are examples of each type of event definition:

Example of an A-type Event

(z/OS)

```

24.10.08          ***** Entire Operations *****                      13:17:08
Owner   SN          End-of-Job Checking + Actions   MVS/ESA   Job JOB1
Network +-----+
-----+
C Action !          Add Event Definition          !      OA
-       !  Event Type ==> a  C Termination Code          MVS/ESA !
-       !  +-----+
-       !  !
-       !  !      Event Definition: Additional 'Job ok/not ok'          !  !
-       !  !
-       !  !      This screen can be used to define an additional          !  !
-       !  !      Job-ok or Job-not-ok event, to define multiple          !  !
-       !  !      End-of-Job actions.          !  !
***** !  !
-  C    !  !      The event is for OK or NOT OK ==> __          !  !
-       !  !
----- !  !      Enter-PF1-----PF3-----PF5-----          !  ! ----
A Activa !  !      Help      End      Save          !  ! ther
P Descr. !  !  +-----+          !  ! Rtn
        !  !      Enter-PF1---PF2---PF3---PF5-----          !  !
Command !  !      Help  Add  End  Save          !  !
Enter-PF1 +-----+          !  ! 2---
          Help  Add  End      Save      Up  Down          Menu

```

Explanation:

This screen can be used to define an additional *Job ok* or *Job not ok* event, to execute multiple End-of-Job actions of the same type.

Usually actions are defined at the job level with the events *All checks ok* or *Any check not ok*. If you want to define an action of the same type two or more times for one of these events, you need additional event entries for *Job ok* or *Job not ok*.

 **Note:** You can use these additional event definitions to set conditions, but it is not recommended, because they are not evaluated for the display of job dependencies (see [Displaying Job Dependencies and Connecting or Disconnecting Jobs](#) in the section *Job Maintenance*).

Example of a C-type Event

(z/OS)

```

24.10.08          ***** Entire Operations *****          13:35:00
Owner   SN          End-of-Job Checking + Actions   MVS/ESA   Job JOB4
Network +-----+-----+-----+-----+-----+-----+
----- !
C Action !           Add Event Definition                !      OA
_        !   Event Type ==> C   C Termination Code      MVS/ESA !
_        !           R   User Exit                          !
_        !           S   String          A Add. Job-ok,-not-ok !
_        !
_        !   Stepname ==> ANYSTEP_____                    !
_        !   Operator ==> EQ   Event ==> C0003                !
_        !
_        !   User Exit ==> _____ in NATURAL Library ==> _____ !
***** !                                           ! ****
_        !   String ==> _____                            !
_        !   Select ==> _____                            !
----- !
A Activa ! Occurrence of event means OK or NOT OK ==> NO    ! ther
P Descr. !
_        ! Enter-PF1---PF2---PF3-----PF5-----PF12--    !
Command !           Help Add   End   Save                Menu   !
Enter-PF1 +-----+-----+-----+-----+-----+ 2---
          Help Add   End       Save           Up   Down                Menu

```

Explanation:

If return code C0003 is received at any step during job execution, the event result will be **not ok**.

Example of an R-type Event

(z/OS)

```

24.10.08          ***** Entire Operations *****          13:40:07
Owner   SN              End-of-Job Checking + Actions   MVS/ESA   Job JOB4
Network +-----+-----+-----+-----+-----+-----+
----- !
C Action !           Add Event Definition                       !   OA
_       !   Event Type ==> R   C Termination Code           MVS/ESA !
_       !           R user exit                                   !
_       !           S String           A Add. Job-ok,-not-ok     !
_       !
_       !   Stepname ==> _____                               !
_       !   Operator ==> __   Event ==> _____                 !
_       !
_       !   User Exit ==> UR1_____ in NATURAL Library ==> USER__ !
***** !
_       !   String ==> _____                                   !
_       !   Select ==> _____                                   !
----- !
A Activa ! Occurrence of event means OK or NOT OK ==> NO         ! ther
P Descr. !
_       ! Enter-PF1---PF2---PF3-----PF5-----PF12--          !
Command !   Help Add End Save                               Menu !
Enter-PF1 +-----+-----+-----+-----+-----+ 2---
         Help Add End Save Up Down Menu

```

Explanation:

User exit UR1 checks the job SYSOUT at job completion. This exit resides in Natural library SYSEORU.

➤ **To edit the user exit**

- Enter E in the line command field of the event on the *End-of-Job Checking and Actions Screen* (see the subsection [Editing End-of-Job User Exits](#)).

Example of an S-type Event

(z/OS)

```

24.10.08          ***** Entire Operations *****          13:44:16
Owner   SN          End-of-Job Checking + Actions   MVS/ESA   Job JOB4
Network +-----+-----+-----+-----+-----+-----+
----- !
C Action !           Add Event Definition                !      OA
_       !   Event Type ==> S   C Termination Code          MVS/ESA !
_       !           R   User Exit                          !
_       !           S   String          A Add. Job-ok,-not-ok !
_       !
_       !   Stepname ==> _____                        !
_       !   Operator ==> ___   Event ==> _____          !
_       !
_       !   User Exit ==> _____ in NATURAL Library ==> _____ !
***** !                                           ! ****
_       !   String ==> EXCEEDS_____                        !
_       !   Select ==> SM*_____                          !
----- !
A Activa ! Occurrence of event means OK or NOT OK ==> NO      ! ther
P Descr. !
_       ! Enter-PF1---PF2---PF3-----PF5-----PF12--      !
Command !           Help Add End Save                          Menu !
Enter-PF1 +-----+-----+-----+-----+-----+ 2---
          Help Add End Save Up Down Menu
    
```

Explanation:

If the string EXCEEDS appears in any system message, the event result is *not ok*.

If you enter NE or <> in the Operator field, the event occurs if the string is *not found*.

If the string to be searched for contains at least one blank at the beginning, in the middle or at the end, it must be enclosed in single quotes. For example:

```
' leading blanks '
```

Notes:

1. All checks of the SYSOUT file and actions which refer to the SYSOUT file, are not executed, if *no* SYSOUT file is available. This case only applies, if the job is submitted as a temporary dummy job.
2. A string search in a non-existent file is handled like *string not found* in an existing file.

34 Adding Event Definition for BS2000 Job

- Adding an Event Definition 342
- Field Descriptions 342

See also BS2000 Defaults for [Event Checking](#).

Adding an Event Definition

> To add an event to be checked at job termination

- 1 Press PF2 (Add) in the End-of-Job Checking and Actions screen.
- 2 A window opens in which you can enter the event definition:

```

12.02.10          ***** Entire Operations *****          13:13:52
Ow +-----+-----+-----+-----+-----+-----+-----+-----+
Ne !                                                    !
-- !                Modify Event Definition                ! ---
C !                                                    ! BS2000 ! OA
_ !   Event Type ==> R  R  Exit          J  Job special event !
_ !                               S  String          A  Add. Job-ok, Job-not-ok !
m !                               U  User Switch      !
_ !                               V  Job Variable      !
_ !                                                    !
_ !   Exit ==> BS2EJC01 in Natural Library ==> EOR531U_ Exit Mode ==> _ !
_ !                                                    !
_ !   String ==> _____ missing ==> _ !
** !   in File ==> _____ ! ***
_ !                               (Optional) Read Password ==> !
_ !                               defined ==> no !
-- !   User Switch ==> 0_ of BS2000 User ID ==> _____ ! ---
A !                                                    ! her
P !   Occurrence of event means OK or NOT OK ==> OK ! Set
_ !                                                    !
Co ! Enter-PF1---PF2---PF3-----PF5----- !
Ent !      Help  Add  End          Save          ! ---
+-----+-----+-----+-----+-----+-----+

```

Field Descriptions

(BS2000)

Meaning of the input fields:

Field	Description	
Event Type	Type of event to be checked. Possible values:	
	A	Additional definition for <i>job ok</i> or <i>job not ok</i> . See Example of an A-type Event .
	J	Special event during job execution. See Example of a J-type Event .
	R	Job SYSOUT is to be checked by a user exit. See Example of an R-type Event .
	S	Occurrence of a specific string in job SYSOUT. See Example of an S-type Event .
	U	Event depends on a user switch. See Example of a U-type Event .
	V	Event depends on contents of a job variable. If you enter V, press Enter immediately to open a special window for job variable definition. See Example of a V-type Event .
User Exit	(for an R-type event) Name of the end-of-job check user exit which is to run on job termination. Entire Operations will perform an Entire System Server logon to the job's execution node with the job's submit user ID prior to the execution of the exit. See also Editing End-of-Job User Exits .	
in Natural Library	(for an R-type event) Natural library in which the user exit resides. This library should be different from the Entire Operations system library.	
Exit Mode	<i>blank</i>	Exit will be executed synchronously. (Default)
	A	Exit will be executed asynchronously (in a Natural task).
String	(for an S-type event) Specify the actual string in the job SYSOUT for which Entire Operations is to check. The search string may contain symbols. Symbols with activation escape characters will be replaced at job activation time. Symbols with submit escape characters will be replaced at end-of-job checking time. Missing symbols will cause a permanent activation error or permanent end-of-job checking error.	
(String) missing	(for an S-type event) If you enter Y here, the event occurs if the string is not found.	
in File	(for an S-type event) If you leave this field blank, Entire Operations searches for the String in the SYSOUT collection file created by the Entire Operations Monitor. You can enter another file here to be searched instead. If the file name contains the activation escape character, a symbol replacement is performed (from the active symbol table). The updated file name is stored in the active database.	
User Switch	(for a U-type event) A valid BS2000 user switch must be defined. (Range: 0 through 31).	
BS2000 User ID	(for a U-type event) The BS2000 user ID of the user switch.	

Field	Description	
Occurrence of event means OK or NOT OK	Specify event check status if the defined event occurs. Possible values:	
	OK	Check ok.
	NO	Check not ok.
	--	No effect on the job result.

When you are finished defining the event, you can proceed in one of the following ways:

- If you defined an R-type event, you can enter E in the line command field of the event on the *End-of-Job Checking and Actions Screen* and press Enter to define the user exit using the Entire Operations Editor. (See the subsection *Editing End-of-Job User Exits.*)
- Press PF2 (Add) to save the definition and clear the window to add another event. You can define any number of events for any one job.
- Press PF5 (Save) to save the event definition(s).
- Press PF3 (End) to close the window and return to the *End-of-Job Checking and Actions Screen*. The new events are listed on this screen.

35

Modifying Event Definition for BS2000 Job

▪ Modifying an Event Definition	346
▪ Example of an S-type Event	347
▪ Example of a U-type Event	348
▪ Example of a V-type Event	349
▪ Adding or Modifying a Job Variable	351
▪ Example of a J-type Event	354

Modifying an Event Definition

> To modify an existing event definition

- 1 Type M in the line command field of the selected event on the *End-of-Job Checking and ActionsScreen*.
- 2 Press Enter.
- 3 The Modify Event Definition window opens with the current values for the event. You can modify the values by overtyping them.
- 4 When you are finished defining the event, you can proceed as follows:
 - Press PF5 (Save) to save the modification.
 - Press PF3 (End) to close the window and return to the *End-of-Job Checking and Actions Screen*.



Note: The modify option (M line command) is available at the event level (job step) only.

See the following examples of **U-type** (user switch), **V-type** (job variable) and **J-type** (special event) event definitions and for examples of **R-type** (user exit) and **S-type** (string search) event definitions.

Example of an S-type Event

- Field Descriptions

(BS2000)

```

24.06.08          ***** Entire Operations *****          13:55:13
Ow +-----+
Ne !
-- !              Modify Event Definition                      ! ---
C !                                     BS2000                !  OA
m !   Event Type ==> S  R  User Exit          J  Job special event !
_ !                                     S  String          A  Add. Job-ok, Job-not-ok !
_ !                                     U  User Switch      !
_ !                                     V  Job Variable     !
_ !
_ !   User Exit ==> _____ in Natural Library ==> _____ !
_ !
_ !   String ==> ABEND_____ missing ==> _                !
** !   in File ==> _____                               ! ***
_ !                                     (Optional) Read Password ==> !
_ !                                     defined ==> no      !
-- !   User Switch ==> 0_ of BS2000 User ID ==> _____ ! ---
A !
P !   Occurrence of event means OK or NOT OK ==> OK        ! Set
!
Co ! Enter-PF1---PF2---PF3-----PF5-----PF12-- !
Ent !   Help Add  End          Save          Menu    ! ---
+-----+
    
```

Field Descriptions

String Search - BS2000

Meaning of the input fields:

Field	Description
String	<p>This string will be searched for in the SYSOUT or any defined file. If the string is found, the event is satisfied. If the search string contains at least one blank at the beginning, in the middle or at the end, it must be enclosed in quotes. For example:</p> <pre>' leading blanks'</pre> <p>The search string may contain symbols.</p> <p>Symbols with activation escape characters will be replaced at job activation time.</p>

Field	Description
	Symbols with submit escape characters will be replaced at end-of-job checking time. Missing symbols will cause a permanent activation error or permanent end-of-job checking error.
missing	You can negate the string search, if you specify "Y" here.
in File	If a BS2000 file is specified here, it will be scanned for the search string. Symbol replacement is possible for the file name or any part of it: <ul style="list-style-type: none"> ■ with activation escape - once at activation time ■ with submission escape - during EOJ checking <p>If this field is left blank, the SYSOUT will be scanned. Symbol replacement is possible. The submission escape character is to be used.</p>

Example of a U-type Event

(BS2000)

```

24.10.08          ***** Entire Operations *****          14:04:07
Ow +-----+-----+-----+-----+-----+-----+-----+
Ne !
-- !              Add Event Definition                          ! ---
C !                                     BS2000                  ! 0A
_ !      Event Type ==> U  R  User Exit          J  Job special event !
_ !                                     S  String          A  Add. Job-ok, Job-not-ok !
_ !                                     U  User Switch                          !
_ !                                     V  Job Variable                          !
_ !
_ !      User Exit ==> _____ in NATURAL Library ==> _____ !
_ !
_ !      String ==> _____ missing ==> _ !
** !      in File ==> _____ ! ***
_ !                                     (Optional) Read Password ==> !
_ !
-- !      User Switch ==> 11 of BS2000 User ID ==> DC1_____ ! ---
A !
P !      Occurrence of event means OK or NOT OK ==> OK ! Rtn
!
Co ! Enter-PF1---PF2---PF3-----PF5-----PF12-- !
Ent !      Help Add End Save Menu ! ---
+-----+-----+-----+-----+-----+-----+-----+

```

Explanation:

The event is ok if user switch 11 of the BS2000 user ID DC1 is *on* at job termination time.

Adding or Modifying a Job Variable

- [Field Descriptions: BS2000 Job Variable](#)
- [Special PF Keys](#)

(BS2000)

➤ **To add or modify a job variable in BS2000**

- 1 Type J in the line command field of the selected event on the *End-of-Job Checking and Actions Screen*. Press Enter.
- 2 The Job Variable Modification window opens with the current values. You can modify the values by overtyping them:



Note: Symbol replacement is performed in the name, if the name contains the activation escape character at least once.

Field Descriptions: BS2000 Job Variable

Field	Description	
Job Variable	Enter the name of a valid BS2000 job variable. If the job variable is specified without an explicit user ID, the job default BS2000 user ID is used as prefix. Symbol replacement is performed in the name, if the name contains the activation escape character at least once.	
at Position	Enter position of job variable substring to be checked. Possible values: 1 through 253.	
in Length	Enter length of job variable substring to be checked. Possible values: 1 through 253.	
with Format	Enter format in which the job variable substring is to be checked against the comparison string. Possible values:	
	A	alphanumeric
	N	numeric
to	Enter the string or field to be set as the job variable or as a substring of the job variable. Symbol replacement will be performed, if the field contains the activation escape character at least once.	
Write Password	If the job variable is write password-protected, specify the password here.	
defined	This field is for information only. It indicates whether a Write Password is defined or not.	
Execute if temporary Dummy	If "Y" or "N" is specified here, the End-of-Job action will be executed according to this definition, if the job was executed as a temporary dummy job. If the field is left empty, the defaults for the action apply.	

Special PF Keys

You can perform the following function from the Job Variable Modification window using this PF key:

Key	Name	Function
PF9	Delete	Delete the complete job variable definition.

Example of a J-type Event

(BS2000)

This function enables you to define special events which could occur during job execution. When such an event occurs this always means *Job not ok*.

```

24.10.08          ***** Entire Operations *****                      14:25:28
Ow +-----+
Ne !
-- !              Add Event Definition                                  ! ---
C !              BS2000                                             ! 0A
_ !      Event Type ==> J  R  User Exit          J  Job special event  !
_ !      +-----+-----+-----+-----+-----+-----+-----+-----+
_ !      !                                     !                               !
_ !      !              Event Definition: Special Event                !
_ !      !              BS2000                                         !
_ !      User !                                     !                               !
_ !      !      Event ==> ____                                         !
_ !      Strin !                                     ! ==> _                               !
** !      in Fi !                                     ! _____ ! ***
_ !      !      The occurrence of this event always has the           ! ==>                               !
_ !      !      meaning 'Job not ok'.                                   !                               !
-- !      User !                                     !                               ! ---
A !      !      Enter-PF1-----PF3-----PF5-----                !
P !      Occur !      Help      End      Save                        !
!      +-----+-----+-----+-----+-----+-----+
Co !      Enter-PF1---PF2---PF3-----PF5-----PF12--
Ent !      Help  Add  End      Save                        Menu  ! ---
+-----+

```

Explanation:

JIR = Job execution interrupted.

This event occurs if an operating system abend or hardware failure occurs during job execution. This is always used when a job did not run on LOGOFF (job ended normally) or ABEND (job interrupted with error). The defined actions are performed after the start of the Monitor.

36 Adding Event Definition for UNIX or Windows Job

- Adding an Event Definition 356
- Field Descriptions 356
- Exit Code Check Error Handling 358

See also [UNIX and Windows Defaults for Event Checking](#).

Adding an Event Definition

➤ **To add an event to be checked at job termination**

- 1 Press PF2 (Add) in the End-of-Job Checking and Actions screen.
- 2 A window opens in which you can enter the event definition:

```

12.02.10          ***** Entire Operations *****          13:18:22
Ow +-----+
Ne !
-- !           Add Event Definition          ! ---
C !                                     Linux ! 0A
_ !   Event Type ==> _ X Exit Code         J Job special event !
_ !                                     R Exit           A Add. Job-ok, Job-not-ok !
_ !                                     S String          !
_ !
_ !
_ !   Exit Code is ==> __ than ==> 0__
_ !
_ !   Exit ==> _____ in NAT Library ==> _____ Exit Mode ==> _
** !
_ !   String ==> _____ missing ==> _
_ !   in File ==> _____
-- !
A !
P !   Occurrence of event means OK or NOT OK ==> __
!
Co ! Enter-PF1---PF2---PF3-----PF5-----
Ent !   Help  Add  End          Save          ! ---
+-----+
  
```

Field Descriptions

Meaning of the input fields:

Field	Description	
Event Type	Type of event to be checked. Possible values:	
	A	Additional definition for job ok or job not ok. See Example of an A-type Event .
	J	Special event during job execution. See Example of a J-type Event .
	R	Job SYSOUT is to be checked by a user exit. See Example of an R-type Event .
	S	Occurrence of a specific string in job SYSOUT. See Example of an S-type Event . Note: A string search in a non-existent file is handled like <code>string not found</code> in an existing file.
X	Exit Code Check. This function allows you to check the exit code of a UNIX or Windows job. The result of the event comes from the exit code of a script under UNIX or Windows or from the exit code of a program executable under Windows. Note: A standard check for the exit code can be defined in the Entire Operations defaults. See also Exit Code Check Error Handling .	
User Exit	(for an R-type event) Name of the end-of-job check user exit which is to run on job termination. Entire Operations will perform an Entire System Server logon to the job's execution node with the job's submit user ID prior to the execution of the exit. See also the section Editing End-of-Job User Exits .	
in Natural Library	(for an R-type event) Natural library in which the user exit resides. This library should be different from the Entire Operations system library.	
Exit Mode	' ' (blank)	Exit will be executed synchronously. (Default)
	'A'	Exit will be executed asynchronously (in a Natural task).
String	(for an S-type event) Specify the actual string in the job SYSOUT for which Entire Operations is to check. The search string may contain symbols. Symbols with activation escape characters will be replaced at job activation time. Symbols with submit escape characters will be replaced at end-of-job checking time. Missing symbols will cause a permanent activation error or permanent end-of-job checking error.	
(String) missing	(for an S-type event) If you enter "Y" here, the event occurs if the string is not found.	
in File	(for an S-type event) If you leave this field blank, Entire Operations searches for the String in the SYSOUT collection file created by the Entire Operations Monitor. You can enter another file here to be searched instead. If the file name contains the activation escape character, a symbol replacement is performed (from the active symbol table). The updated file name is stored in the active database.	

Field	Description
Occurrence of event means OK or NOT OK	Specify event check status if the defined event occurs. Possible values:
	OK Check ok .
	NO Check not ok .
	-- No effect on the job result.

When you are finished defining the event, you can proceed in one of the following ways:

- If you defined an R-type event, you can enter E in the line command field of the event on the *End-of-Job Checking and Actions Screen* and press Enter to define the user exit using the Entire Operations Editor. (See the subsection *Editing End-of-Job User Exits*.)
- Press PF2 (Add) to save the definition and clear the window to add another event. You can define any number of events for any one job.
- Press PF5 (Save) to save the event definition(s).
- Press PF3 (End) to close the window and return to the *End-of-Job Checking and Actions Screen*. The new events are listed on this screen.

Exit Code Check Error Handling

If a SYSOUT file read error occurs while the Monitor is trying to read the exit code from the job SYSOUT, this error will be treated as a temporary error. The Monitor will attempt to read the exit code up to 10 times. The time between the attempts is equal to the Monitor wait time. If there is still no success after 10 attempts, the job will be set to the *status permanent error*.

37

Modifying Event Definition for UNIX or Windows Job

- Modifying an Event Definition 360
- Example of an S-type Event 360

Modifying an Event Definition

> To modify an existing event definition

- 1 Type **M** in the line command field of the selected event on the *End-of-Job Checking and Actions Screen*. Press **Enter**.
- 2 The Modify Event Definition window opens with the current values for the event. You can modify the values by overtyping them. When you are finished defining the event, you can proceed as follows:
- 3 Press **PF5** (Save) to save the modification.
- 4 Press **PF3** (End) to close the window and return to the *End-of-Job Checking and Actions Screen*.



Note: The modify option (**M** line command) is available at the event level only.

Example of an S-type Event

- [Field Descriptions](#)

(UNIX and Windows)

```

+-----+
!
!           Modify Event Definition
!
!                                     Linux
!   Event Type ==> X  X  Exit Code      J  Job special event
!                                     R  Exit      A  Add. Job-ok, Job-not-ok
!                                     S  String
!
!
!   Exit Code is ==> >_ than ==> 0__
!
!   Exit ==> _____ in NAT Library ==> _____ Exit Mode ==> _
!
!   String ==> _____ missing ==> _
!   in File ==> _____
!
!
!   Occurrence of event means OK or NOT OK ==> NO
!
!   Enter-PF1---PF2---PF3-----PF5-----
!           Help  Add   End           Save
!
+-----+

```

Field Descriptions

Field	Description
Event Type	Type of event to be checked. Possible values:
	A Additional definition for job ok or job not ok. See Example of an A-type Event .
	J Job special event.
	R Job SYSOUT is to be checked by a user exit. See Example of an R-type Event .
	S Occurrence of a specific string in job SYSOUT . See Example of an S-type Event . Note: 1. All checks of the SYSOUT file and actions, which refer to the SYSOUT file, are not executed, if no SYSOUT file exists. This case will apply, if the job is submitted as a temporary dummy job. 2. A string search in a non-existent file is handled like "string not found" in an existing file.
	X Exit code.

Field	Description	
Operator	(for a X-type event) Specify a logical operator to compare the received return code with the value entered in the Event Type field. Possible values:	
	= or EQ	Exit code is equal to specified value.
	>= or GE	Exit code is greater than or equal to specified value.
	>> or GT	Exit code is greater than specified value.
	<= or LE	Exit code is lower than or equal to specified value.
	< or LT	Exit code is lower than specified value.
	<> or NE	Exit code is different from specified value.
User Exit	<p>(for an R-type event) Name of the end-of-job check user exit which is to run on job termination.</p> <p>Entire Operations will perform an Entire System Server logon to the job's execution node with the job's submit user ID prior to the execution of the exit.</p> <p>See also the section Editing End-of-Job User Exits.</p>	
in Natural Library	(for an R-type event) The Natural library in which the user exit resides. This library should be different from the Entire Operations system library.	
Exit Mode	' ' (blank)	Exit will be executed synchronously. (Default)
	'A'	Exit will be executed asynchronously (in a Natural task).
String	<p>This string will be searched for in the SYSOUT or any defined file. If the string is found, the event is satisfied. If the search string contains at least one blank at the beginning, in the middle or at the end, it must be enclosed in single quotes. For example:</p> <pre>' leading blanks'</pre> <p>The search string may contain symbols.</p> <p>Symbols with activation escape characters will be replaced at job activation time.</p> <p>Symbols with submit escape characters will be replaced at end-of-job checking time.</p> <p>Missing symbols will cause a permanent activation error or permanent end-of-job checking error.</p>	
missing	<p>String missing:</p> <p>If "Y" is defined here:</p> <p>If the string is not found, the event will be satisfied.</p>	
in File	<p>If a UNIX or Windows file is specified here, it will be scanned for the search string. Symbol replacement is possible for the file name or any part of it:</p> <ul style="list-style-type: none"> ■ with activation escape - once at activation time ■ with submission escape - during EOJ checking 	

Field	Description	
	If this field is left blank, the SYSOUT will be scanned. Symbol replacement is possible. The submission escape character is to be used.	
Occurrence of event means OK or NOT OK	Specify the event check status, if defined event occurs. Possible values:	
	OK	Check ok.
	NO	Check not ok.
	--	No effect on the job result.
	<p>Usually actions are defined at the job level with the events All checks ok or Any check not ok. If you want to define an action of the same type two or more times for one of these events, you need additional event entries for job ok or job not ok. See Example of an A-type Event.</p> <p>Note: You can use these additional event definitions to set conditions, but it is not recommended, because they are not evaluated for the display of job dependencies (see Displaying Job Dependencies and Connecting or Disconnecting Jobs in the section Job Maintenance).</p>	

38

Deleting an Event Definition

➤ **To delete an event definition**

- 1 Type **D** in the line command field of the selected event on the *End-of-Job Checking and Actions Screen*. Press **Enter**. A window opens in which Entire Operations asks you to confirm the deletion by entering the event name.
- 2 Press **Enter** to perform the deletion and close the window.



Note: The **D** line command is available only at the event level. Deletion of an event also deletes all associated definitions made via the *End-of-Job Checking and Actions Screen* (action for the event, etc.).

39

EOJ Checking Defaults for Various Operating Systems

▪ Common Defaults for Event Checking	368
▪ z/OS Defaults for Event Checking	368
▪ z/VSE Defaults for Event Checking	369
▪ BS2000 Defaults for Event Checking	369
▪ UNIX and Windows Defaults for Event Checking	370

Common Defaults for Event Checking

- [Symbol Replacement Error during Job Submission](#)

Symbol Replacement Error during Job Submission

If a symbol replacement error occurs at submission time, the event **JNR** “Job not run - JCL error” will be set to “occurred”.

If the event **JNR** is not defined, it will be added to the active job.

z/OS Defaults for Event Checking

- [Precedence of Event Checks](#)

If no events are defined for a job, Entire Operations provides a default check of the return code type (for a C-type event):

- If a system code unequal to 0 (zero) has occurred at any job step, the job is evaluated as *not ok* if this event is not explicitly defined. This also applies, if an event has occurred which has been defined using ANYSTEP. The occurrence of such an event has the priority over definitions, which apply for any steps.
- It can be defined throughout the system, how the occurrence of the message *IEF201I job terminated* because of condition codes is treated (see *Entire Operations Administration Documentation*). If it was defined there that this message shall not default to *not ok* as a consequence, then this is written to the log, the job set to the status *Job interrupted*, and the job is treated as *not ok*. If this was not defined, then the occurrence of IEF201I in SYSOUT has no consequences.
- If the message *IEF287I data set not cataloged* is received, it is written to the log and the job is treated as *not ok*.
- If a condition code or user code, which is not explicitly defined, is received at any job step, it is compared with the highest value accepted as ok. The job is not ok if the code is higher than this default. The job is also not ok, if no explicit check has returned an ok. For more information, see the subsection Entire Operations Defaults in the section *System Administrator Services* of the *Entire Operations Administration Documentation*.
- If a string defined in the Global Message Code Table is found anywhere in the SYSOUT, its severity equivalent is compared with the highest condition code value accepted as ok. The job is not ok if the severity equivalent is higher than this default.
- Events like JCL error, Job not run, Job interrupted cause the status “not ok” by default. Dedicated event definitions can override this.
- In all other cases, the job is accepted as ok.

Precedence of Event Checks

- If a `/MAXCC` or `/ANYSTEP` check has been defined, and this event occurs, no check is done on the highest value accepted as still OK
- `/MAXCC` precedes `/ANYSTEP`.

z/VSE Defaults for Event Checking

Date Formats in z/VSE SYSOUT

The same rules as for z/OS defaults apply to z/VSE.



Note: No system codes are available in the z/VSE operating system.

The following messages lead to a *not ok* status of the z/VSE job:

- PROGRAM ABEND
- ENDED ABNORMALLY
- OPERATOR CANCEL

Date Formats in z/VSE SYSOUT

In z/VSE SYSOUT the following date formats are recognized and handled:

American Format	MM/DD/YY, MM/DD/YYYY
European Format	DD/MM/YY, DD/MM/YYYY

If a date contains a 2-digit year specification in SYSOUT, then Entire Operations adds the century for further processing. In this case, a sliding window of + / - 50 years is applied.

BS2000 Defaults for Event Checking

Entire Operations either uses a defined Monitor job variable for a job or creates its own job variable, if job variables are available.

- If the Monitor job variable displays an abnormal end, the job is *not ok*.
- By default, if defined message keys appear, the job is regarded as faulty and is set to *not ok*: These message codes (see *Defaults for BS2000*) in the *Administration* documentation) are default settings which are in effect after the installation of Entire Operations. However, they can be deliberately adapted. It is possible to completely delete this table.

Please note each time this table is modified that possibly faulty jobs may not be set to *not ok* any longer. This list can be changed.

- If messages like *program dump*, etc. appear in the SYSOUT, the job is *not ok*.
- The Global Message Code Table is checked as described for z/OS, above.
- In all other cases, the job is accepted as *ok*.

UNIX and Windows Defaults for Event Checking

Entire Operations inserts start and end messages (EOR0301, EOR0302) in the SYSOUT.

- If the end message EOR0302 is missing, the job is treated like an interrupted job. Because it is possible that the SYSOUT of an asynchronous child processes is written behind the end message, the message EOR0302 will be searched in the whole SYSOUT file. If the End-of-Job checking detects that there is unexpected SYSOUT behind the end message, the process times for accounting cannot be obtained. A note is being written to the log:

Process times for accounting not found ... SYSOUT of background processes may exist.

- The Global Message Code Table is checked, as described for **z/OS**.
- If no special event checking was defined for a job (**UNIX and Windows**), then the event is compared to a system-wide default value for UNIX or Windows. Depending upon this check, the job result can be set to *not ok*.
- In all other cases, the job is accepted as *ok*.

40

Creating Online Documentation for Events

You can browse text description of active event and write or modify a text description of a master event.

➤ **To do that**

- 1 Type P in the line command input field of the appropriate event on the *End-of-Job Checking and Actions Screen*. Press Enter. The Entire Operations Editor screen appears.
- 2 You can write a text using Editor commands. A full description of the Entire Operations Editor is contained in the Software AG Editor Documentation .

41 Adding Output Condition Definitions

- Adding Output Conditions for an Event 374
- Adding Output Conditions for a Job Event 374
- Fields and Columns: Output Conditions 375

You can define output conditions for an event or a job.

Adding Output Conditions for an Event

➤ **To add an output condition definition for a defined event**

- 1 Type C in the line command field of the selected event on the *End-of-Job Checking and Actions Screen*. Press Enter.

The Output Conditions window opens with a list of existing **output conditions**).

- 2 Proceed as described after **Column Headings: Output Conditions** in the subsection **Adding Output Conditions for a Job Event**.

Adding Output Conditions for a Job Event

You can define a condition at the job level according to End-of-Job status (job ok = All checks ok, job not ok = Any check not ok).

➤ **To do that**

- 1 Type the C line command in the *All checks ok* or *Any check not ok* line command field on the *End-of-Job Checking and Actions Screen*.
- 2 Press Enter.

The Output Conditions opens with a list of existing output conditions:

```

24.10.08          ***** Entire Operations *****          15:07:09
Owner   EXAMPLE          End-of-Job Checking + Actions      MVS/ESA   Job JOB-02
Network E60-FLOW          Run                               Date
-----
C Action      +-----+ OA
-   P         !                                     !
-         !   Job ended ok                          !
-         !           Output Conditions              !
-         !   Cmd State  Condition Name      Reference   Run        !
-         !   -   Set    E60-JOB2-01        RUN          !
-         !   -   Set    E60-JOB2-02        RUN          !
-         !   -   Reset  E60-J015-0        RUN          !
-         !   -   Reset  E60-J019-0        RUN          !
*****         !   -   Reset  E60-JOB1-0        RUN          ! *****
c  C  P       !                                     !
-   PR       !                                     !
-----         !                                     !
A Activation !                                     ! ther
P Descr. R Re !                                     ! Rtn
              !   D Delete  M Modify  W Where used   !
Command => ___ ! Enter-PF1---PF2---PF3---PF5-----PF7---PF8---PF12-   !
Enter-PF1---PF !   Help  Add  End  Save  Up  Down  Menu  ! 2---
              Help Ad +-----+ u
↵

```

Fields and Columns: Output Conditions

- [Field Descriptions: Output Conditions](#)
- [Column Headings: Output Conditions](#)

Field Descriptions: Output Conditions

Field	Description
Condition	<p>Condition name. Symbol replacement is possible. The submission escape character is to be used.</p> <hr/> <p>A condition that can be used across networks is called a global condition. If you want to add or modify a global condition:</p> <ul style="list-style-type: none"> ■ The prefix is the plus sign (+). ■ A global condition is assigned to the SYSDBA owner and network.
Reference	<p>An output condition can be set with different references (usually the current network run number). Possible values:</p>

Field	Description
	<p>RUN Current run number of the job network is assigned. This is the default.</p> <p>ABS Absolute condition. Exists only once, because it is independent of run numbers.</p> <p>RCM Sets an active condition with owner, network, run number of calling network and appends multiple suffixes.</p> <p>RCM can be used to determine that the conditions defined for a predecessor job apply to all jobs that reference this predecessor in multiple activated subnetworks. In this case, the input condition of the corresponding successor job must be referenced with RCA: see RCA in the section Possible References for Input Conditions.</p> <p>The reference is evaluated and set when the active condition is created by the Monitor according to the End-of-Job definition.</p>
Run	Run number of job to which this condition is to apply when modifying condition for an active job. Default is a blank character.
State	Determines whether the defined output conditions must be met if the associated event occurs. Possible values:
	<p>SET The specified conditions must be met.</p> <p>This is the default.</p> <p>RESET The specified conditions are ignored.</p>

- Press **PF3** (End) to save the definition and return to the list of conditions in the Output Conditions window.

You can define up to 50 output conditions for any one event at the job step level, or at the job level according to *All checks ok* and *Any checks not ok*.

Column Headings: Output Conditions

Column	Description
Cmd	One-character line command input field. Possible values:
	D Deletes condition as output condition for the job.
	M Modifies the output condition definition.
	W Displays a table of jobs for which this condition is defined as input or output condition. For more information, see the subsection Displaying Output Condition Use .
State	A condition can be set (to TRUE) or reset (to FALSE) on the occurrence of an event.

Column	Description
Condition Name	User-defined name for condition. Use this name if you wish to define this output condition as an input condition for another job.
Reference	Notation used to distinguish between the same condition in different job runs. Use this reference date if you wish to define this output condition as input condition for another job. If left blank, the current run is assumed.
Run	Run number of job to which this condition applies when modifying condition for an active job.

➤ To add an output condition definition

- Press PF2 (Add).

The Output Condition Addition window opens:

```

24.10.08          ***** Entire Operations *****          15:11:33
Owner   EXAMPLE          End-of-Job Checking + Actions      MVS/ESA   Job JOB-02
Network E60-FLOW                               Run       Date
-----
C Action  +-----+ OA
c   P      !                                     !
-       !   Step STEP1 returns Condition Code = C0000      !
-       +-----+                                     +       !
-       !                                     ! n       !
-       !   Step STEP1 returns Condition Code = C0000      !       !
-       !   Output Condition Addition                       !       !
-       !   Condition ==> _____                       !       !
***** !   Reference ==> _____                       !       ! *****
_   C P !   Run ==> _____                             !       !
-   PR !   Set/Reset ==> _____                        !       !
----- !                                     !       !
A Activa !   PF1 Help  PF3 End  PF5 Save                   !       ! ther
P Descr. +-----+                                     !       !
          !   D Delete  M Modify  W Where used           !
Command => __ ! Enter-PF1---PF2---PF3---PF5-----PF7---PF8---PF12- !
Enter-PF1---PF !   Help  Add  End  Save  Up  Down  Menu  ! 2---
          Help Ad +-----+                                     +   u
↵
    
```


42

Modifying Output Condition Definitions

➤ To modify an existing output condition

- 1 Type M in the line command input field of the selected condition in the Output Conditions window.

The Output Condition Modification window opens with the current values of the conditions.

The Output Condition Modification and Addition windows have the identical **layout**. The input fields are explained above under **Field Descriptions: Output Condition Addition/Modification**.

- 2 Modify the values.
- 3 Press PF3 (End) to save the new definition and return to the Output Conditions window.



Note: After you add or modify an output condition definition, a loop check is performed on the affected network. The same conditions apply as described in the subsection **Checking for a Loop in a Job Network** in the section **Network Maintenance** with one exception: if a loop is detected in the job flow, no corresponding message appears.

43 Define Symbol Modification or Setting

- Modifying a Symbol Modification Definition 382
- Symbol Modification Definition Screen 383
- Field Descriptions: Symbol Modification 383
- Special PF Keys: Symbol Modification 384

You can modify master or active symbols as End-of-Job action. Analogous to BS2000 job variables, it is possible to set or modify substrings of symbols.

Modifying a Symbol Modification Definition

➤ To modify an existing event definition:

- 1 Type Y in the line command field of the selected event on the *End-of-Job Checking and Actions Screen*.
- 2 Press Enter.
- 3 The Modify Symbol Definition window opens with the current values. You can modify the values by overtyping them.
- 4 When you have finished defining the event, you can proceed as follows:
 - Press PF5 (Save) to save the modification.
 - Press PF3 (End) to close the window and return to the *End-of-Job Checking and Actions Screen*.

Symbol Modification Definition Screen

```

24.10.08          ***** Entire Operations *****          15:18:58
Owner   SN          End-Of-Job Checking + Actions   MVS/ESA   Job A-1
Network A-1  +-----+
-----+-----+-----+-----+-----+-----+-----+-----+
C Action    ! Step ANYSTEP returns Condition Code = C0004      !
y          ! Symbol Modification                                !
-          !                                                    !
-          !          Owner   (SAME)_____                        !
-          !          Network A-1                                     !
-          !          Run                                           !
-          !          Symbol Table (SAME)_____                       !
-          !          Set Contents of A A Active M Master B Both      !
-          !          Symbol _____                               !
*****      !          at Position ___ in Length ___ Format A    !
_ C        STUX ! to Value                                       !
_ A        S Y ! _____                                       !
-----+-----+-----+-----+-----+-----+-----+-----+
A Activat. C C !
P Descr. R Rec ! Execute if temp. Dummy ==> _                          !
!                                                    !
Command => ___ ! Enter--PF1-----PF3-----PF5-----PF9-----      !
Enter-PF1---PF2 !          Help      End      Save      Delete          !
Help Add +-----+-----+-----+-----+-----+-----+-----+

```

Field Descriptions: Symbol Modification

Meaning of the input fields:

Field	Description
Set Contents of ...	Symbol Modification Mode. In this field you can specify where the symbol setting should be performed.
	A Perform modification in the active table.
	M Perform modification in the Master table.
	B Perform modification in both tables.
Symbol	A symbol or a symbol substring can be set or modified depending on the occurrence of any event.
in Symbol Table	Symbol table to be used.
Run	Run number to be used. If empty (zero), a master symbol will be modified.

44

Displaying Output Condition Use

➤ **To display a list of jobs, for which the output condition is defined as an input or output condition**

- 1 Enter `W` in the line command input field of the selected condition in the Output Conditions window. Press `Enter`.

The following window opens:

```

24.10.08          ***** Entire Operations *****          15:21:01
Owner   EXAMPLE   End-of-Job Checking + Actions   MVS/ESA   Job JOB-02
Network E60-FLOW                                     Run       Date
-----
C Action   S +-----+s 0A
-         P     S !
-         !     Job ended ok
+-----+
!
!
!           Condition: E60-JOB2-01
!
! Output Condition of           Input Condition for
!
! -----+----- !
! Owner      Network   Job      ! Owner      Network   Job      !
! EXAMPLE    E60-FLOW  JOB-02  ! EXAMPLE    E60-FLOW  JOB-03  !
! EXAMPLE    E60-FLOW  JOB-03  !
!
!
!
!
!
!
! Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10-----
!
!           End           Up     Down
!
E +-----+
   Help Ad +-----+nu
↵
    
```

- 2 Press PF3 (End) to return to the Output Conditions window.

45

Editing End-of-Job Checking or Action User Exits

▪ Creating or Editing User Exits	388
▪ Example	388

Creating or Editing User Exits

For End-of-Job checking user exits, the common exit parameter list **NOPXPL-A** must be used. The parameter P-CALL-PLACE will contain EJC for end-of-job checking or EJA for end-of-job action.

Note on parameter list usage:

The field P-JOB-NUMBER (N5) should no longer be used. It is too short for some operating systems. Use the field P-JOB-ID (A10) instead.

➤ To create or edit the user exit for End-of-Job checking or end-of-job action

- 1 Type E in the line command field of the associated event in the *End-of-Job Checking and Actions Screen*. Press Enter.

The Entire Operations Editor screen appears. If a user exit has already been defined, it appears on the screen.



Note: The edit option - E line command - is available only at the event level, and only if the event is an R-type.

- 2 You can modify the existing user exit or create a new one using Editor commands and PF keys.

A user check routine must be a Natural subprogram. The subprogram returns a code zero to signal *job ok* status. Any other code signals *job not ok*.

Example

An End-of-Job checking user exit is shown in the example below. This routine summarizes the contents of a numeric field in the SYSOUT and compares it with a given value. If both are equal, the event is treated as `ok`, otherwise as `not ok`.

For a full description of the Editor, see the Software AG Editor Documentation.

```

DEFINE DATA PARAMETER USING NOPXPL-A
LOCAL
1 READ-SPOOL VIEW OF READ-SPOOL      /* AN ENTIRE SYSTEM SERVER VIEW
  2 RECORD
  2 REDEFINE RECORD
    3 RECORD-HEADER      (A05)
*
1 #NF          (I02) CONST <10>    /* MAX. NUMBER FOR SEPARATION
1 #SUM         (N08) INIT  <0>
1 #SUM-EXPECTED (N08) CONST <2000>
1 #FIELD      (A10/1:#NF)
1 #USERID     (A08)
1 #DBENV      (A10)
1 #NETWORK-VERSION (A10)

1 #LOG-STATUS (N04)
1 #LOG-MESSAGE (A60)
1 #LOG-TARGET (A03) CONST <'NOP'>
1 #USE-NAT-MSG-CACHE (L) CONST <FALSE>
END-DEFINE

```

```

* -----
* This EOJ User Exit will read the Job SYSOUT (data set 2)
* and will summarize a defined field.
* If the sum is equal to a given value, this routine returns
* an 'OK' (P-RC = 0), otherwise 'NOT OK'.
*
* A log record will be written to document the result.
*
RESET #SUM
RS. FIND READ-SPOOL
  WITH NODE = P-EXECUTION-NODE AND JOB-NUMBER = P-JOB-NUMBER
  AND TYPE = 'SO' AND DATA-SET = 2 AND RECORD-NUMBER >= 1
  ACCEPT IF RS.RECORD-HEADER = 'REC01'      /* interesting only
  SEPARATE RS.RECORD LEFT JUSTIFIED INTO #FIELD(*)
  ADD VAL(#FIELD(3)) TO #SUM
END-FIND
*
IF #SUM = #SUM-EXPECTED
  RESET P-RC
  MOVE 'Sum check is ok' TO #LOG-MESSAGE
ELSE
  MOVE 1 TO P-RC
  COMPRESS 'Sum:' #SUM 'Expected:' #SUM-EXPECTED INTO #LOG-MESSAGE
END-IF
#USERID := *USER
MOVE 9999 TO #LOG-STATUS
CALLNAT 'NOPULW93N'      /* WRITE LOG RECORD
  #LOG-STATUS #LOG-MESSAGE #DBENV
  P-OWNER P-NETWORK #NETWORK-VERSION P-RUN P-JOB

```

```
P-JOB-ID #LOG-TARGET # USE-NAT-MSG-CACHE  
END
```

46

Defining Activation of Jobs or Job Networks

The occurrence of a defined event during job execution can trigger the activation of a specified single job or a whole job network. You can define activation of a job or network for specific events or at the job level for *All checks ok/Any check not ok*.

➤ **To specify job or job network activation as End-of-Job checking and actions**

- 1 Type A in the line command input field of the selected event on the *End-of-Job Checking and Actions Screen*. Press Enter.

The following window opens when defining job or network activation at the job level for All checks ok:

```

17-03-03          ***** Entire Operations *****          18:25:46
Owner EXAMPLE    Network B60-FLOW  Version                      Job JOB-SUBNET
End-of-Job Ch +-----+-----+-----+-----+-----+-----+
-----+-----+-----+-----+-----+-----+-----+
C Action        ! Job ended ok                      ! eans OA
                !           Network / Job Activation    !
                !                               !
                ! Owner      ==> _____      !
                ! Network   ==> _____      !
                ! Version   ==> _____      !
                ! Job       ==> _____      ! (empty: whole network)
                !                               !
                ! use Time Frame ==> _         !
*****          ! Schedule Usage ==> _         ! *****
a              ! Schedule Owner ==> _____ !
_             ! Schedule      ==> _____ !
-----+-----+-----+-----+-----+-----+-----+
A Activat. C    ! Execute if temp. Dummy ==> _     ! 0 Other
P Descr. R Re  !                               ! bol Set
              ! ---PF1---PF3-----PF5-----PF9-----PF12-- !
Command => ___ ! Help End      Save      Delete      Menu      !
              +-----+-----+-----+-----+-----+-----+

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Add  End      Save      Up      Down      Menu

```

- 2 Specify the owner, network name and, if a single job is to be activated, the job name in the appropriate fields in the window. If the activation is to be performed depending on a time frame and / or schedule, the necessary fields must be defined too.
- 3 Press PF5 (Save) to save the definition.

Field Descriptions: Job/Network Activation

Field	Description
Owner	Owner of the network to be activated.
Network	Network to be activated.
Job	Job to be activated. If empty, the whole network will be activated.
use Time Frame	T - Use time frame of called network for activation. empty - Activate immediately. (default)
Schedule Usage	C Activate only if the current day is defined in the network's resp. explicitly defined schedule. empty Activate always. (default)
Schedule Owner	If an explicit schedule is to be checked, you may specify the schedule's owner here. If empty, the owner of the network to be activated will be used.

Field	Description
Schedule	If an explicit schedule is to be checked, you may define it here. If empty, the schedule of the network to be activated will be used.
Execute if temporary Dummy	If Y or N is specified here, the End-of-Job action will be executed according to this definition, if the job was executed as a temporary dummy job. If the field is left empty, the defaults for the action apply.

You can display a selection list of available Networks, Jobs and Owners by entering an asterisk * in the appropriate field and pressing `Enter`.



Notes:

- Wildcard Usage**
- You can select Owner, Network and Job names from selection windows invoked by entering search criteria with an asterisk * as wildcard in the input fields and pressing `Enter`.
- For example, enter ABC* in the Network field and press `Enter` to list all network names beginning with ABC and belonging to the specified owner. Enter an asterisk * in the Job field and press `Enter`, for example, to list all jobs in the specified network.
- If you leave the Job field blank, the whole specified network is activated.
- Authorization**
- Activation is only possible if the network belongs to the defining user or if the defining user is authorized to activate the network of another user. The user defining the network to be activated must at least have authorization to activate this network (see the subsection [Granting Definition: Authorizing Other Users or Owners to Access a Network](#) in the section *Network Maintenance*).

47 Defining Recovery Action

- Recovery Action Definition 396
- System Symbols for Recovery Actions 398

You can define a recovery process to be started as the result of a defined event during job execution. You can also define recovery action at the job level according to *Any check not ok*. Usually, recovery is used in case of job failure.

The conventional way to trigger a recovery network would be to define an event which activates the recovery network. Recovery is then started only if the event occurs or if any check is *not ok*.

Recovery Action Definition

- [Field Descriptions: Recovery Definition](#)
- [Special PF Keys: Recovery Definition](#)

However, Entire Operations also provides a line command on the *End-of-Job Checking and Actions Screen* to define recovery action.

➤ To define recovery action

- 1 Type R in the line command input field of the selected event on the *End-of-Job Checking and Actions Screen*. Press Enter.
- 2 The Recovery Definition window opens, in which you can specify the recovery network to be started. (The recovery network must be defined in the Network Maintenance facility.):

```

3.  +-----+
4.  !                                           !
5.  ! Job ended not ok                         !
6.  !           Recovery Definition           !
7.  ! Owner      ==> (same)_____          !
8.  ! Network    ==> (same)_____ Version ==> (same)_____ !
9.  ! Job        ==> _____ (blank: whole network) !
10. !                                           !
11. ! Same Run   ==> _ (Y/N)                  !
12. ! Reschedule ==> _ (Y/N/S)              !
13. ! Repeat     ==> _____              !
14. ! Wait Time  ==> _____ Minutes      !
15. !                                           !
16. ! ---PF1---PF3-----PF5-----PF9-----PF12--- !
17. ! Help End      Save      Delete      Menu      !
18. +-----+

```

For example, a recovery job can be a compression job started after the message "Library full" is received. The string "Library full" should be defined as an S-type event. Entire Operations starts the recovery job automatically when the event occurs.

- [Field Descriptions: Recovery Definition](#)

- Special PF Keys: Recovery Definition

Field Descriptions: Recovery Definition

Field	Description
Owner	Name of the owner of the recovery network. Default: (same). If you do not change the default, (same) means the owner of the network to be recovered. The default value is replaced at execution time.
Network	Name of network containing the recovery job(s). Default: (same). If you do not change the default, (same) means the same network as the one containing the job(s) to be recovered. The default value is replaced at execution time. When you copy a recovery job to another network, the default values (same) for owner, network, and network version remain unchanged, so that the recovery job can be used unchanged in the network to which it was copied.
Version	Network version. Default: (same). If you do not change the default, (same) means the same network version as the one containing the job(s) to be recovered. The default value is replaced at execution time. When you copy a recovery job to another network, the default values (same) for owner, network, and network version remain unchanged, so that the recovery job can be used unchanged in the network to which it was copied.
Job	Name of the last job in the recovery network. This job must terminate successfully to start rescheduling.
Same Run	Single recovery jobs can be defined in the same network and can be executed under the same run number as the network to be recovered. All active objects of the network are then accessible for the recovery job with this run number. Possible values:
	Y Use same run number (default for single jobs).
	N Use a separate run number for the recovery (default for whole network).
Reschedule	Specifies whether the job is to be rescheduled for submission after recovery. Possible values:
	Y Resubmit the job.
	N Do not resubmit the job. (default)
	S Stop original network after recovery.
Repeat	Maximum number of times the original job is to be rescheduled after a recovery. Default is 1. The Entire Operations Monitor sets the reserved condition <jobname>-MAX-RETRY if the retry limit was reached with unsuccessful rescheduling attempts. "Repeat" is meaningful only if "Reschedule" is set to "Y".
Wait Time	The time to wait in minutes until the recovery is started, and between the recovery attempts.

You can display a selection list of available Networks, Jobs and Owners by entering an asterisk * in the appropriate field and pressing Enter.

Special PF Keys: Recovery Definition

You can perform the following function from the Recovery Definition window using this PF key:

Key	Name	Function
PF9	Delete	Delete current recovery definition.

System Symbols for Recovery Actions

When defining a recovery action, the following Entire Operations system symbols can be prefixed with the escape symbol for symbol prompting and used, for example, in a comment line in the JCL of the recovery job:

Field Name	Description
P-C-OWNER	Owner of the job making the call.
P-C-NETWORK	Network of the job making the call.
P-C-JOB	Name of job making the call.
P-C-RUN	Run number of the job making the call.
P-C-SUFFIX	Value of the suffix symbol of the job making the call.

They are substituted at JCL load with their current values. In the SYSOUT of the recovery job, you can then find the substitution of the parameters:

```

*** Symbol   : P-C-OWNER
*** Value    : GFR
*** Symbol   : P-C-NETWORK
*** Value    : NET-1
*** Symbol   : P-C-JOB
*** Value    : JOB-6
*** Symbol   : P-C-RUN
*** Value    : 208
*** =====
2 //STEP1    EXEC PGM=NOPCONTI
3 //STEPLIB DD  DISP=SHR,DSN=NOP.DEV.LOAD
*** CALLING OWNER    GFR
*** CALLING NETWORK  NET-1
*** CALLING JOB      JOB-6
*** CALLING RUN      208
***

```

In this way you can find out the job making the call, that is, the job to be recovered by the recovery job.

48

SYSOUT Actions

- Defining Job SYSOUT Actions 402
- Special PF Keys: SYSOUT Actions 403

You can define the job SYSOUT to be cancelled or printed as the result of a defined event or the End-of-Job status (*All checks ok* and *Any check not ok*).

Defining Job SYSOUT Actions

➤ To define job SYSOUT actions

- 1 Type the S line command in the appropriate line command field in the End-of-Job Checking and Actions screen.
- 2 Press Enter. The SYSOUT Actions window opens:

```

+-----+
!                                     !
!   Job ended ok                       !
!                                     !
!           SYSOUT Actions              !
!                                     !
!   Delete SYSOUT after Job Completion  ===> D   !
!   Print  SYSOUT                       ===> P   !
!   Log SYSOUT in Entire Operations Log  ===> L   !
!   Log SYSOUT, then Delete it          ===> N   !
!   Log SYSOUT, then Print it           ===> M   !
!   Pass SYSOUT to Entire Output Mgmt   ===> O   !
!                                     !
!           Please Select ===> 0               !
!                                     !
!   Set Spool Class after Completion ===> _____ !
!                                     !
!   PF1 Help   PF3 End                   PF9 ToMF   !
+-----+

```

End-of-Job user exits are available for more complex actions, for example for obtaining selected information from the job SYSOUT. For information on the input fields, see [Field Descriptions: SYSOUT Actions](#).

- 3 Press PF5 (Save) to save the definition.
- 4 Press PF3 (End) to close the window and return to the [End-of-Job Checking and Actions Screen](#). SYSOUT actions are performed at job completion if the associated event occurs.

Field Descriptions: SYSOUT Actions

Field	Description
Please Select	Enter one of the following letters in this field to perform the function:
	D Delete SYSOUT on job completion.
	P Release SYSOUT for printing.
	L Log SYSOUT in Entire Operations log file.
	N Log SYSOUT and then cancel it.
	M Log SYSOUT and then print it.
	O Pass files to Entire Output Management (NOM).
Spool Class to be set after Job Completion	<p>You can specify that the spool class of a job is to be modified after completion. The definition created here overrides the default setting.</p> <p>Note:</p> <ol style="list-style-type: none"> 1. This field is protected unless the executing operating system is z/OS or VSE. 2. The class defined here has priority over other definitions. 3. If no definition is made on the job level, the default for the node level will be used. 4. If no other definition is found, the system-wide default for the operating system will be used (see <i>Defaults for z/OS and z/VSE</i> in the <i>Administration</i> documentation). 5. You should only define SYSOUT class changes under <i>All checks ok</i> or <i>Any check not ok</i>. 6. The spool class change will be allowed only if the job is in the OUT queue.

Special PF Keys: SYSOUT Actions

■ Copy SYSOUT to Mainframe

Key	Name	Function
PF9	ToMF	Copy the SYSOUT from UNIX or Windows to a mainframe node.

Copy SYSOUT to Mainframe

If this function is used, the following window opens:

```
+-----+
!
!  SYSOUT Copy from UNIX / Windows to Mainframe  !
!
!  Owner      SN          SYSOUT Node          !
!  Network    X517-S1     SYSOUT Cat ID          !
!  Run        SYSOUT User ID          !
!  Job        JOB3
!  Exec.Node  517
!
!  Note: Please use Job Definition function 'Specials' !
!          (PF6) to modify the values                !
!
!  ---PF1---PF3-----
!  Help  End
!
+-----+
```

This window is for information only. The definition of SYSOUT copy target can be made in the operating system dependent job definitions for UNIX and Windows. See [Defining Operating System Specific Execution Features](#) in the section *Job Maintenance*.

49

Message Sending

■ Introduction	406
■ Defining a Message	406
■ Fields and Columns: Message and Message Recipients	407
■ Message Destination Types	408
■ Message Recipients	409
■ Special PF Keys: Message and Message Recipients	410
■ E-Mail on Mainframes	410
■ E-Mail on UNIX and Windows Systems	411

See also:

- *Global Messages for Events* in the *Administration* documentation.

Introduction

You can define or modify a message to be sent as the result of a defined event, or if the job terminated *ok* or *not ok*. This is especially useful to notify appropriate personnel of job failure.

The Entire Operations Monitor continues to send a message until it has been successfully transmitted.

Defining a Message

A message can be defined for any one event or at the job level according to All checks ok and Any check not ok.

> **To do so:**

- 1 Type U in the line command input field of the appropriate event on the *End-of-Job Checking and Actions Screen*. Press Enter.

A **Message and Message Recipients** window opens.

```

+-----+
!
! Owner SN          Network A-1          Version V00003      !
!   Job A-1        !
! Job ended ok     !
!           Message and Message Recipients !
!
! Text ==> hugo_____ !
!
! to ==>  Destination Type      Node      !
!           $ESN_____ =EMAIL__ N0517    !
!           _____ _____ _____ !
!           _____ _____ _____ !
!           _____ _____ _____ !
!           _____ _____ _____ !
!           _____ _____ _____ !
!
! Execute if temp. Dummy (E0J only) ==> Y  !
! Enter-PF1---PF3-----PF5-----PF9----- !
!           Help End       Save       Delete  !
+-----+

```

Add your definitions in the corresponding fields.

For further information, see the following sections:

- [Fields and Columns: Message and Message Recipients.](#)
- [Special PF Keys: Message and Message Recipients](#)

- 2 Press PF5 (Save), to save your definitions.
- 3 Press PF3 (End), to close the **Message and Message Recipients** window and return to the **End of Job Checking and Actions** screen.

Please refer to the common description of the [message receiver table definition](#).

Fields and Columns: Message and Message Recipients

The fields and columns provided to define a message and message recipients are described in the following table:

Field	Description
Text	<p>Text of message to be sent if the associated event occurs.</p> <p>If the text is not modifiable on the map, the effective message text will be generated for the event automatically by Entire Operations.</p> <p>If the message contains an activation escape character, text replacement is performed from the active symbol table.</p>
Destination	<p>Enter CONSOLE to send the message to the operator console (mainframes only).</p> <p>For Con-nect: enter a valid Con-nect user ID.</p> <p>In z/OS and z/VSE: enter the user ID of a TP system user.</p> <p>In BS2000: enter a terminal name.</p> <p>In UNIX: enter a UNIX user ID valid in the UNIX environment of the addressed server. The message is sent with the UNIX mail function.</p> <p>This field can also contain a symbol. This allows you to define a longer destination as well, for example an e-mail address. If e-mail addresses are to be defined (in an EBCDIC character set) on the mainframe, the string "(a)" must be used instead of "@".</p>
Type	See <i>Message Destination Types</i> for a detailed description.
Node	<p>Enter the Entire System Server node, by which the message is to be sent. This is not necessary for messages to Con-nect.</p> <p>Default: the execution node of the job.</p>
Execute if temporary Dummy	<p>This field has a meaning for End-of-Job actions only.</p> <p>If Y or N is specified here, the End-of-Job action will be executed according to this definition, if the job was executed as a temporary dummy job.</p> <p>If the field is left empty, the defaults for the action apply.</p>

Message Destination Types

Type	Description
=EMAIL	<p>Sends the message by email to any receiver in the Intranet or Internet.</p> <p>The email will be sent in plain text mode.</p> <p>Since the receiver field is usually too short for an email address, it is recommended to put a symbol into the destination field, preceded by the activation escape character. The effective email address must be defined as symbol value in the symbol table used.</p> <p>The "at" sign (@) must be entered as (a) on mainframes (in the EBCDIC character set), for example:</p>

Type	Description
	<p>user(a)any.host</p> <p>For UNIX, see also Email on UNIX and Windows Systems.</p> <p>As Windows does not have a built-in <code>sendmail</code> command, a command line tool must be specified in the node definition.</p> <p>Only for Entire System Server with version lower than 3.2.1:</p> <p>For z/OS, some special definitions, which are used for message sending, must be made made for Entire System Server nodes with version lower than 3.2.1. See <i>Other Definitions for a Node (Mainframe)</i> in section <i>Definition of Nodes</i> of the <i>Administration</i> documentation.</p>
=EMH	<p>Sends the message by email to any receiver in the Intranet or Internet.</p> <p>The email will be sent in HTML format.</p> <p>Please refer to the =EMAIL description for more about email sending.</p>
=EXIT	<p>Use the global message sending exit.</p> <p>See <i>Global Message Sending Exit</i> in section <i>Global User Exits</i> of the <i>Administration</i> documentation.</p>
=MAILBOX	Sends the message to the mailbox specified in Destination .
=COMPLET	Sends the message explicitly to a Com-plete user (z/OS only).
=TSO	Sends the message explicitly to a TSO user (z/OS only).
=CONSOLE	Sends the message to the operator console (mainframes only).
Other	BS2000: The processor name related to the BS2000 terminal name to be specified in the Destination column.

Message Recipients

If Con-nect is installed in your system, the message is routed to the defined Con-nect user ID(s). The message includes the environment (Owner, Network, Job, Run) and the standard event text.

If sending to Con-nect is not possible, the message is routed via Entire System Server to the TP system user with the designated user ID. If no special node has been defined, the executing node is used.

If you enter =MAILBOX in the Processor field, the message is sent to the Entire Operations internal mailbox.



Note: If no message recipient is defined for a network, the messages are sent to the SYSDBA mailbox.

For further details on mailboxes, see the section [Mailboxes](#).

Special PF Keys: Message and Message Recipients

The PF key assignment in the lower area of the **Message and Message Recipients** window is as follows:

Key	Name	Function
PF9	Delete	Delete all message receiver definitions. Caution: Use this function with care.

E-Mail on Mainframes

- z/OS und z/VSE
- BS2000

z/OS und z/VSE

1. The sender name contains “Entire Operations” for all sending platforms.
2. An additional line with the Entire Operations event text is added to the mail.

All further changes can be done in Entire Operations, when the future Entire Operations Server view for e-mail sending will be available.

Usage Notes:

- Before using an z/OS Entire System Server node for sending e-mail, you must invoke the node definition in Entire Operations and enter the line command S (specials) for that node.

For further information, see the subsection Special Definitions for a Node (Mainframe) in the section *Definition of Nodes* of the *Administration* documentation.

- **Mail Destination** and **Mail SYSOUT Class** must be defined.
- The e-mail send routine for z/OS performs an Entire System Server logon, if necessary. The user ID for the logon is the active job's submit user ID. If no submit user ID is defined for the job, the default user ID for the node will be used.
- To use a z/OS (MVS) Entire System Server node with version 321 or higher for sending e-mail, make sure that all users who send e-mail, are correctly defined in the security systems (if applicable). It might be necessary to define user ID as USS users.

For more information, see the documentation for Entire System Server, for the operating system and for the security system.

BS2000

E-mail can be sent via BS2000 nodes, if Entire System Server nodes with version 321 or higher are used.

- Before using a BS2000 Entire System Server node for sending e-mail, you must invoke the node definition in Entire Operations and enter the line command `S` (specials) for that node.

For further information, see the subsection *Special Definitions for a Node (Mainframe)* in the section *Definition of Nodes* of the *Administration* documentation.

- The e-mail send routine for BS2000 performs an Entire System Server logon, if necessary. The user ID for the logon is the active job's submit user ID. If no submit user ID is defined for the job, the default user ID for the node will be used.

E-Mail on UNIX and Windows Systems

- [UNIX](#)
- [Windows](#)

UNIX

To be able to send e-mail on UNIX machines from Entire System Server, the following mail programs must be available there:

AIX, HP-UX, Linux, Sun Solaris	sendmail
others	rmail



Note: The *Return-Path:* of the e-mail is being set to the same value as *Reply-To:*.

Windows

- [Message Command](#)

- [Usage Notes](#)

Message Command

To be able to send e-mail on Windows machines from Entire System Server, you must define a message command. This is a (third-party) command line tool to be used for sending e-mail via Entire System Server / Windows.

Usage Notes

Before using a Windows node for sending e-mail, you must invoke the node definition in Entire Operations and enter the line command S (specials) for that node.

For further information, see the subsection *Special Definitions for a Node (UNIX and Windows)* in the section *Definition of Nodes* of the *Administration* documentation.

50

Defining End-of-Job Action User Exit

You can specify a Natural user exit to be triggered by a specified event.

Specifying a User Exit

The End-of-Job checking user exits must use the common exit parameter list **NOPXPL-A**. The parameter P-CALL-PLACE will contain `EJA`.

➤ To specify a user exit to be triggered

- 1 Type `X` in the line command input field of the appropriate event on the *End-of-Job Checking and Actions Screen*. Press Enter. The following window opens in which you can define a user exit:

```

12.02.10          ***** Entire Operations *****          13:21:21
Owner   SN          End-of-Job Checking + Actions   Linux   Job J001
Network X517-S1          Run           Date
-----+-----+-----+-----+-----+-----+-----+-----+-----+
C Action      !                                     ! eans OA
x             ! Exit Code is = 0                       ! ok
_             !           Execute EOJ Action Exit         !
_             !                                     !
_             !           Library   ===> _____         !
_             !           Exit       ===> _____         !
_             !           Exit Mode ===> _                 !
_             ! Execute if temp. Dummy ===> _             !
*****       !                                     ! *****
_             ! Enter-PF1---PF3---PF4---PF5-----PF9----- !
_             !           Help End Edit Save Delete           !
-----+-----+-----+-----+-----+-----+
A Activat. C Cond. D Delete E Edit Exit J JV L ReI Res M Modify O Other
P Descr. R Recov. S SYSOUT T Output Mgmt U User Msg X Action Exit Y Symbol Set

Command => _____
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
          Help Add End Save Up Down Menu
    
```

- 2 Press PF5 (Save) to save the data.
- 3 Press PF3 (End) to close the window and return to the *End-of-Job Checking and Actions Screen*.

Field Descriptions: Execute EOJ Action User Exit

Meaning of the input fields:

Field	Description	
Library	Name of the Natural library containing the user exit.	
User Exit	Name of the Natural user exit. To display a selection list for user exits available in a given library, first enter the name of a Natural library in the Library field, then enter an asterisk * in the User Exit field and press Enter.	
Exit Mode	' ' (blank)	Exit will be executed synchronously. (Default)
	'A'	Exit will be executed asynchronously (in a Natural task).
Execute if temporary Dummy	If "Y" or "N" is specified here, the End-of-Job action will be executed according to this definition, if the job was executed as a temporary dummy job. If the field is left empty, the defaults for the action apply.	

Special PF Keys: Execute EOJ Action User Exit

You can perform the following functions from the Execute EOJ Action User Exit window using these PF keys:

Key	Name	Function
PF4	Edit	Create or edit an End-of-Job user exit.
PF9	Delete	Reset execution of a user exit.

Execution Notes

- Entire Operations will perform an Entire System Server logon to the job's execution node with the job's submit user ID prior to the execution of the exit.
- End-of-job action user exits will not be performed for temporary *dummy jobs*.

51 End-of-Job Action: Release Resource

You can specify to release an explicitly kept resource as End-of-Job action.

It is possible to release a resource held by any other job of this network with the 'Keep' option. The released amount is equal to the kept amount.

Map

```

24.10.08          ***** Entire Operations *****          16:01:57
Owner   SN        End-of-Job Checking + Actions   MVS/ESA   Job IEFBR14
Network A-1          Run                          Date
-----
C Action   +-----+ eans OA
-          !
-          ! Job ended ok
-          !           Release Kept Resource
-          !
-          ! Resource ==> _____
-          !
-          ! Execute if temp. Dummy ==> _
-          !
-          ! *****
***** ! ---PF1---PF3-----PF5-----PF9-----PF12-- ! *****
|          !   Help  End       Save       Delete       Menu       !
-          +-----+
-----
A Activat. C Cond.  D Delete  E Ed Usr Rtn  J JV  L ReI Res  M Modify  O Other
P Descr.  R Recov.  S SYSOUT T Output Mgmt  U User Msg X Action Rtn Y Symbol Set

Command => _____
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Add  End       Save       Up       Down                          Menu
↵

```

Fields

Field	Meaning
Resource	The resource to be released. The resource can be selected by usage of a wild card.
Execute if temporary Dummy	If Y or N is specified here, the End-of-Job action will be executed according to this definition, if the job was executed as a temporary dummy job. If the field is left empty, the defaults for the action apply.

52

Defining Other Actions

- Invoking the Other Actions Function 420
- Special PF Keys: Other Actions 421

Invoking the Other Actions Function

To define a job to be deactivated automatically, even if it ended not ok, you must use the Other Actions function.

➤ **To do so:**

- 1 Enter 0 in the line command field of the appropriate event on the *End-of-Job Checking and Actions Screen*. Press Enter.

The Other Actions window opens:

```

24.10.08          ***** Entire Operations *****                16:07:34
Owner   EXAMPLE   End-of-Job Checking + Actions   MVS/ESA   Job JOB-02
Network E60-FLOW          Run                   Date
-----
C Action          +-----+ ans OA
o   P             !                               !
-             ! Step STEP1 returns Condition Code = C0000   ! ok
-             !                               !
-             !                               !
-             ! Accept the job if not ok           ==> _ (Y/N) !
-             ! (The job can be deactivated, if executed)   !
-             !                               !
-             !                               !
*****          ! ---PF1---PF3-----PF5-----PF9----- ! *****
_   C P           !   Help  End    Save      Delete      !
-   PR           +-----+
-----
A Activation  C Cond.  D Delete  E Edit User Rtn  J Job Var  M Modify  O Other
P Descr.    R Recovery S SYSOUT T Output Mgmt  U User Messages  X Action User Rtn

Command => _____
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Add   End      Save      Up    Down      Menu
←
    
```

- 2 In this window you can define other actions. If you enter Y in the field Accept the job if not ok, the active job can be automatically deactivated, even if it ended *not ok*. Otherwise, the job remains active until a correction or manual deactivation.

Special PF Keys: Other Actions

You can perform the following function from the Other Actions window using this PF key:

Key	Name	Function
PF9	Delete	Delete data entered here.

VIII

Active Job Networks

When Entire Operations activates a job network, a copy of the network defined in the master database is made on the active database and a run number assigned to it. Several copies of the same network can be in the active database, each distinguished by its run number.

Entire Operations allows extensive modification of job networks and jobs after activation.

This section explains the maintenance functions you can perform on active jobs and networks using the **Active Job Networks** option on the Main Menu.

For example, you can modify active copies of a job definition including associated logical conditions, resources and JCL. All modifications to active jobs take place on the active database and are valid for the current job run only, so that all original definitions on the master database remain unchanged.



Note: Your user profile must contain the appropriate authorization to access active networks.

[Maintaining Active Job Networks](#)

[Maintaining Active Jobs](#)

[Maintaining Active Job Conditions](#)

[Maintaining Active JCL \(Job Control Language\)](#)

Related Topic:

- *Active Database* in the *Concepts and Facilities* documentation

53

Maintaining Active Job Networks

- Listing Active Job Networks 426
- Deactivating Active Job Networks 428
- Repeating Active Job Networks 431
- Displaying Operating System Information on Active Tasks 433
- Listing Next Network Activations 436
- Viewing the Execution History of a Network Active 436

Listing Active Job Networks

> To list all active job networks of an owner

- Select the **Active Job Networks** option from the Main Menu.

An **Active Job Networks** screen similar to the example below appears:

```

29.10.08          ***** Entire Operations *****          13:48:42
Owner REQUEST          Active Job Networks
Selection OR_____
-----
Cmd #Run Owner      Network      Node  Description
   *----- *-----
_      1 REQUEST    HZE-TEST    N0148
_      1 REQUEST    N02226-1    N0148 stop cyclic job
_      6 REQUEST    P211238A    N0053
_      1 REQUEST    P220056     N0409 NPR3329, Windows error text
_     10 REQUEST    P230426     N0148 Prevent multiple schedule activations
_     10 REQUEST    P230426B    N0148 Prevent multiple schedule activations
_      6 REQUEST    P240663B    N0148 Mult. activations
_      1 REQUEST    P249577     N0517 userid 'root' logon
_      5 REQUEST    P249637B3   N0148 time frame modification
_      1 REQUEST    P252323A    N0145 DAEFDB26 Copy subsystem          /Week1
_      2 REQUEST    P273387A    N0148 dummy due to ...
***** m o r e *****
A Active Jobs N Deactivate Network P Description R Repeat
S Act.Jobs Schedule T Accounting X History
Command => _____
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      End              Up      Down  ATask All  NxtSt Menu
    
```

- [Listing a Selected Range of Networks](#)
- [Columns: Active Job Networks](#)

- [Line Commands: Active Job Networks](#)

Listing a Selected Range of Networks

You can specify the range of networks to be listed by entering A, G, O or R in the **Selection** field on the **Active Job Networks** screen and pressing ENTER. For more information, see [Selecting a Range of Networks to be Listed](#) in the section *Network Maintenance*.

Columns: Active Job Networks

The following table explains the column headings for the data listed on the **Active Job Networks** screen:

Field	Description
Cmd	One-character line command input field. For possible input values, see Line Commands: Active Job Networks .
#Run	Number of currently active runs of the network.
Owner	Owner name for the network. For possible selection criteria, see Filtering Items in a Result List in Using Entire Operations .
Network	Network name. For possible selection criteria, see Filtering Items in a Result List in Using Entire Operations .
Node	Execution node specified for the network as the default for its jobs.
Description	Short description of network as defined in the job network definition on the master database.

Line Commands: Active Job Networks

The following line commands are available on the **Active Job Networks** screen:

Line Command	Description
A	List active jobs for the selected network. See Listing All Active Jobs for a Single Network Active .
N	Deactivate the selected network. See Deactivating an Active Job Network .
P	Display text description of selected network (browse mode only). See Viewing Long Descriptions of Active Jobs .
R	Repeat active network. See Repeating Active Job Networks .

Line Command	Description
S	Active jobs schedule. See Viewing and Modifying the Active Jobs Schedule .
T	Display job network accounting data. See Displaying Network Accounting Data .
X	Display network execution history. See Viewing the Execution History of a Network Active .

The following special PF keys are available on the **Active Job Networks screen**:

PF Key	Name	Function
PF7	Up	Scroll list backwards.
PF8	Down	Scroll list forwards.
PF9	ATask	List active tasks specific to the operating system. See Displaying Operating System Information on Active Tasks .
PF10	All	List all active jobs on the All Active Jobs screen. See Listing All Active Jobs for All Networks .
PF11	NxtSt	Next scheduled network activations. See Listing Next Network Activations and Displaying Next Network Starts (System-Wide) in the section Schedule Maintenance .

Deactivating Active Job Networks

There are several ways of preventing or interrupting the execution of an active job network. These are described in the following section.

- [Deactivation User Rights \(Profile Settings\)](#)
- [Deactivating Active Runs for a Network](#)
- [Deactivating all Networks](#)
- [Deactivation of planned Networks \(Activation Canceling\)](#)

- [Stopping a Running Network](#)

Deactivation User Rights (Profile Settings)

Deactivation requires active job deletion user rights and either network master deletion rights or network activation rights. See *User Definitions and Profile Settings* in the *Administration* documentation.

Deactivating Active Runs for a Network

If you wish to prevent Entire Operations from submitting active jobs for a network, you can deactivate a network before the first job run is performed.

➤ To deactivate run numbers for active jobs in a selected network

- 1 On the [Active Job Networks screen](#), type **N** (Deactivate Network) in the **Cmd** column next to the selected network, and press **ENTER**.

A selection window opens with a list of job run numbers for the selected network together with the activation date:

```

05.02.16          **** +-----+-----+
Owner EXAMPLE    | Owner EXAMPLE      Network B60-FLOW
Selection OR_____ |
-----+-----+-----+-----+
Cmd #Run Owner   Network | Please mark the run numbers, for which
      *----- *----- | active jobs are to be deactivated.
_      EXAMPLE   ADMIN   | Then continue with 'Enter'.
_      EXAMPLE   ADMIN2  |
n P    5 EXAMPLE   B60-FLOW | Cmd   Run   Start Time
_ P    EXAMPLE   B60-FLOW | _    2132  01.02.16 06:30
_ P    EXAMPLE   B60-FLOW | _    2133  02.02.16 06:30
_ P    EXAMPLE   B60-FX   | _    2134  03.02.16 06:30
_ P    EXAMPLE   B60FLOW1 | _    2135  04.02.16 06:30
_ P    EXAMPLE   B60FLOW2 | _    2136  05.02.16 06:30
_      EXAMPLE   DEMO-NET |
_      EXAMPLE   EXA-NET1 |
_ P    EXAMPLE   E01-CO-F |
*****
A Active C Copy D Delete F
P Descr R Activate S Schedu
Command => _____
| PF3 End PF7 Up PF8 Down PF9 Deact all
| PF9: Deactivate all visible runs.
+-----+-----+-----+-----+

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Add End Save Up Down NxtSt Menu
  
```

- 2 In the **Cmd** column, type any character next to the job run number(s) you want to deactivate, and press ENTER.

Or:

Press PF9 (Deact all) to deactivate all job run numbers listed in the window.

If the active runs do not fit on one screen, it might be necessary to repeat this function several times.



Note: Be careful with PF9, because no confirmation is requested.

- 3 Press PF3 (End) to close the window.

Command Processing

- The message in Progress in the **selection window** indicates that the command has been accepted.

The message planned in the **selection window** indicates that the active job network is not completely activated. No active jobs exist in this state.

- Deactivation is performed by the Entire Operations Monitor in the background. A slight delay can occur during this operation.

If the Monitor is not active, you can deactivate a job using the **Deactivation in Foreground** option of **Special Functions** of the **System Services Menu**.

Deactivating all Networks

the system administrator can stop the Entire Operations Monitor to halt the execution of all job networks. Jobs currently being executed continue until normal termination, unless they are explicitly cancelled from the operating system.

Deactivation of planned Networks (Activation Canceling)

If you try to deactivate planned network activation, you will be prompted for confirmation:

```
+-----+
!                                     !
! Please confirm                       !
! the Deactivation of NET01-517 (4737)  !
! by entering NET01-517                !
!      ==> _____                  !
! PF3 End                               !
+-----+
```

You must confirm the deactivation by entering the network name here.



Note: Confirmation of deactivation of planned activations can be switched off globally by the setting the **Confirm activation canceling** option on the **Default Setting (3)** screen of the Entire Operations defaults described in the *Administration* documentation.

Stopping a Running Network

To interrupt a running network, add an input condition, e.g. `WAIT-HERE`, for the job before which the network is to stop running, and ensure that it will not be set. For more information, see [Adding an Active Condition](#).

To restart the network at the point of interruption, you can manually set the input condition, or remove it from the active job definition (see [Displaying and Modifying an Active Job Definition](#)).

Repeating Active Job Networks

Authorized users can repeat single or multiple active job network runs.

Only active jobs which were already executed will be repeated.

Active output conditions of these jobs will be reset prior to repetition.

➤ To repeat or activate an active job network

- 1 On the **Active Job Networks** screen, type the line command `R` next to the network run you want to repeat or manually activate.

A run number window similar to the example below opens:

```

09.02.16          **** +-----+
Owner EXAMPLE    |           |
Selection OR_____ | Owner EXAMPLE      Network E01-CONTI
-----+-----+
Cmd #Run Owner   | Network |
  *-----* *-----+
_ P   22 EXAMPLE | B60-FLOW |
_ P    1 EXAMPLE | E01-C0-1 |
R     8 EXAMPLE | E01-CONT |
_ P   8 EXAMPLE | E52-LOGR |
_ P    6 EXAMPLE | E60-FLOW |
_ P    5 EXAMPLE | Z60-FLOW |
-----+-----+
Please mark the network runs
to be repeated

Cmd   Run   Start Time
_     1     05.02.16 17:56
_     598   05.02.16 05:00
_     599   05.02.16 14:45
_     600   05.02.16 16:52
_     601   08.02.16 05:00
_     602   09.02.16 05:00

*****
A Active Jobs N Deactivate
S Act.Jobs Schedule T Accou
Command => _____
PF3 End  PF7 Up  PF8 Down
-----+-----+

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      End      Up      Down  ATask All  NxtSt Menu

```

- 2 Select one or more network runs to be repeated with any character, and press ENTER.

For each run to be repeated, a confirmation window similar to the example below opens:

```

+-----+
| Owner EXAMPLE   Network B60-FLOW   Run 2138 |
| Adapt job time frames for the repetition ==> N (Y/N) |
| PF3 End |
+-----+

```

- 3 Adapt job time frames for the repetition. If Y (yes) is entered here, the active job's time frames will be adapted as follows:

Formula	Example
time delay = current time - old earliest start	13:00 - 9:00 = 4 hours
new earliest start = current time	13:00
new latest start = old latest start + time delay	17:00 + 4 hours = 21:00
new deadline = old deadline + time delay	18:00 + 4 hours = 22:00

Displaying Operating System Information on Active Tasks

This section covers the following topics:

- [Listing All Active Tasks](#)
- [Columns: All Active Tasks](#)
- [Listing Active Tasks according to Node](#)
- [Listing Active Tasks according to Type](#)

Listing All Active Tasks

➤ To display operating system information on active tasks

- Press PF9 (ATask) on the [Active Job Networks screen](#).

An **All Active Tasks** screen similar to the example below appears:

```

16-10-14          ***** Entire Operations *****          16:47:35
                        All Active Tasks
Node ==> 55522  MVS/ESA                                     Type ==> JOB_____
-----
  DP ID Type   JobId Name      Step      Proc      Status   Disp Region   CPU
236 30 JOB     24962 XGSLDDIE NATRPC
236 52 JOB     23836 PQATIMER PQAMIT    N          YES     3520    4.00
238 10 JOB     62099 SYSP00L  NAT          YES     2188    0.01
255 11 JOB     23765 SNRPC531 EOR##531   SWAP-OUT NO     1372    2.27
230 11 JOB     23766 SNRPC543 EOR##543   YES     1308    0.08
230 12 JOB     36186 ESI       EOR##543   YES     1516    0.32
255 12 JOB     59842 ETSWCPEF LYCANSRV   SWAP-OUT NO     1496    3.22

***** Bottom of Data *****
Command => _____

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

This screen displays information on all active tasks running on the operating system, including jobs not defined to Entire Operations (z/OS only).

Columns: All Active Tasks

The following table explains the column headings for the data listed on the [All Active Tasks screen](#):

Column	Description	
DP	Dispatching priority assigned by operating system.	
ID	JES2 batch initiator identifier (for JOB type tasks only).	
Type	Type of job. Corresponds to the value entered in the Type field (see Listing Active Tasks according to Type).	
JobId	Identifier of job as assigned by the operating system or by the job entry subsystem.	
Name	Name of job (job card name) as assigned in the job statement of the JCL.	
Step	Name of job step currently being executed.	
Proc	Name of procedure currently being executed.	
Status	Status of address space. Possible values:	
	V=R	Running in real memory.
	TERM	Address space is terminating.

Column	Description	
	NON - SWAP	Address space cannot be swapped.
	SWAP - OUT	Address space is swapped out.
	(blank)	Address space is swapped in.
Disp	Dispatch ability of address space:	
	YES	Address space is dispatchable.
	NO	Address space is not dispatchable.
Region	Amount of real storage used by address space (in KB).	
CPU	Amount of CPU consumed by address space (to one-hundredth of a second).	

Operating system information is offered as a view only and cannot be modified.

Listing Active Tasks according to Node

➤ **To select an Entire System Server node for which the tasks are to be listed**

- 1 Enter a node number in the **Node** input field of the **All Active Tasks screen**.

Or:

Enter an asterisk (*) and press ENTER to open a selection window for nodes and mark a listed node with any character.

- 2 Press ENTER.
- 3 The **All Active Tasks** screen lists tasks for the selected node only.

Listing Active Tasks according to Type

➤ **To list active tasks according to type**

- 1 In the **Type** input field of the **All Active Tasks screen**, enter one of the following values (shortest possible input underlined):

Type	Description
(blank)	Show all types.
*	Show all types.
<u>I</u> NI	JES2 initiator.
<u>J</u> OB	Batch jobs (default).
<u>S</u> TC	Started tasks.
<u>I</u> SU	TSO users.

The operating system information which appears immediately after the node number cannot be modified.

- 2 Press ENTER.

The **All Active Tasks screen** now lists tasks of the selected type only.

Listing Next Network Activations

➤ To list all next scheduled and manual network activations

- Press PF11 (NxtSt) on the **Active Job Networks screen**.

The **Next scheduled Network Starts screen** appears.

For further information, see *Displaying Next Network Starts (System-wide)* in the section *Schedule Maintenance*.

Viewing the Execution History of a Network Active

➤ To display execution history for a selected network

- On the **Active Job Networks screen**, type X in the line command input field next to the required network, and press ENTER.

A **History** screen appears for the selected network.

For further instructions and information, see *Displaying the Network Execution History* in the section *Schedule Maintenance*.

54

Maintaining Active Jobs

▪ Listing Active Jobs	438
▪ Listing Jobs of an Active Subnetwork	448
▪ Adding a New Job to the Active Network	448
▪ Viewing and Modifying the Active Jobs Schedule	450
▪ Canceling, Holding and Releasing Active Jobs	453
▪ Resubmitting an Active Job	456
▪ Deactivating a Job in an Active Network	459
▪ Reactivating an Active Job	461
▪ Displaying and Modifying an Active Job Definition	462
▪ Modifying EOJ Checking and Actions	464
▪ Viewing Long Descriptions of Active Jobs	464
▪ Displaying Prerequisites for Active Jobs	465
▪ Viewing and Modifying Resources Used by Active Jobs	467
▪ Modifying Scheduling Parameters	469
▪ Browsing Active Job SYSOUT	469

Listing Active Jobs

You can access active runs of a network to list and maintain active jobs.

User Restrictions for Active Jobs Lists

Users can be restricted to view jobs on the [Active Jobs](#) or [All Active Jobs](#) screen.

If the user is of type A (Administrator) or if the user has access to the owner SYSDBA, then all active jobs are displayed. In all other cases, the following active jobs are displayed:

- From networks of the current owner,
- From networks of owners to which the user has access,
- From networks to which the user has access.

See also [Granting Definition: Authorizing Other Users or Owners to Access a Network](#) in the section *Network Maintenance*.

This section covers the following topics:

- [Listing All Active Jobs for All Networks Active](#)
- [Listing All Active Jobs for a Single Network Active](#)
- [Listing Active Jobs by Process Status](#)
- [Listing Active Jobs According to Run Numbers](#)

Listing All Active Jobs for All Networks Active

➤ **To list all active jobs for all active networks**

- Press PF10 (All) on the **Active Job Networks** screen.

An **All Active Jobs** screen similar to the example below appears:

```

08.02.16          ***** Entire Operations *****          17:58:51
                        All Active Jobs
Selection AW_____ Submit User ID _____
-----
C Owner      Network  Job          Run    JobId  Time Message
EXAMPLE--- *----- *-----
_ EXAMPLE    B60-FLOW    TST-JIR      2159          17:26 E60-JOB4-0 - 2159 - RU
_ EXAMPLE    B60-FLOW    MILE1-TEST   2159          17:26 Job Type JOB requires
_ EXAMPLE    B60-FLOW    TST-J        2159          17:26 Node 0 undefined
_ EXAMPLE    B60-FLOW    ABC3A        2159          17:26 Job Type JOB requires
_ EXAMPLE    Z60-FLOW    JOB-06       568    14951 13:51 Ended ok
_ EXAMPLE    Z60-FLOW    JOB-05       568          13:50 Dummy Job terminated
_ EXAMPLE    Z60-FLOW    JOB-04       568    14911 13:49 Ended ok
_ EXAMPLE    E60-FLOW    JOB-06       4670   14918 13:45 Ended ok
_ EXAMPLE    E60-FLOW    JOB-05       4670          13:44 Dummy Job terminated
_ EXAMPLE    E60-FLOW    JOB-04       4670   14893 13:43 Ended ok
_ EXAMPLE    Z60-FLOW    JOB-03       568          13:43 JOB-03 ended normally
***** m o r e *****
A Sched.P C Canc D Deact E Edit G Gen.JCL H Hold I InCond J JCL K A.Res L Res.
M Mod. O EOJ P Prose R Resub. S SYSOUT T Stop U Rel. V RA W Wf Y SubC Z Subnet
Command => _____

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Add  End  ACond Save  ResUs Up    Down  ATask Left  Right Menu
  
```

The screen lists all jobs currently active in your Entire Operations environment for the selected owner(s) depending on your view authorizations (see also [User Restrictions for Active Jobs Lists](#)). Additionally, network runs are listed where activations are planned and in the process, but not yet completed.

A hyphen (-) instead of a job name in the **Job** column indicates that the network has been activated but not yet started. The hyphen (-) is replaced by the job name when the network starts to execute the job.

 **Note:** By contrast, the [Active Jobs screen](#) lists all active jobs for a single network selected from the **Active Job Networks** screen.

Sort Order

The sort order for this screen can be defined individually for each user in the user definition: see *Sort Orders in Lists* described in the *Administration* documentation.

This section covers the following topics:

- [Input Fields: Filtering Jobs on the All Networks Active Screen](#)
- [Columns: All Active Jobs](#)

- [Commands: All Active Jobs](#)

Input Fields: Filtering Jobs on the All Networks Active Screen

You can limit the number of jobs listed on the **All Active Jobs** screen by entering selection criteria in the following input fields:

Owner, Network and Job

For valid input values, see [Filtering Items in a Result List](#) in the section *Using Entire Operations*.

Submit User ID

Enter a user ID in the **Submit User ID** field and press ENTER to list only those active jobs with the specified submit user ID.

The user ID is case-sensitive.

Selection by Job Status

You can enter a job status in the **Selection** field to select jobs according to their status: see [Listing Active Jobs by Process Status](#).

Columns: All Active Jobs

The following table explains the column headings for the data listed on the **All Active Jobs** screen:

Column	Description
C	One-character line command input field. For possible values, see the line commands in the section <i>Commands: Active Jobs</i> .
Owner	Network owner. For possible selection criteria, see <i>Filtering Items in a Result List</i> in <i>Using Entire Operations</i> .
Network	Network to which the job belongs. For possible selection criteria, see <i>Filtering Items in a Result List</i> in <i>Using Entire Operations</i> .
Job	Job name as defined to Entire Operations. For possible selection criteria, see <i>Filtering Items in a Result List</i> in <i>Using Entire Operations</i> .
Run	Job run number automatically assigned at activation time. The job run number uniquely identifies an active copy of a job together with the job name.
JobId	Job identifier as assigned by the operating system or by the job entry subsystem.
Time	Last action or check time for the job.
Message	Last message issued for the job by Entire Operations. Press PF11 (Right) or PF10 (Left) to display the full message text.

Column	Description
	For a list of possible messages and their meaning, see <i>Messages in Active Jobs Display</i> in the <i>Messages and Codes</i> documentation.

Commands: All Active Jobs

Line Commands

For a description of the available line commands, see the [line commands](#) in the section *Commands: Active Jobs*.

 **Note:** Most of these line commands refer to a specific job and may therefore not be used in lines referring to network runs. For these lines, only the commands D and A are allowed.

Special PF Keys

For a description of available PF keys, see the [special PF keys](#) in the section *Commands: Active Jobs*.

Listing All Active Jobs for a Single Network Active

You can access active runs of a network to list and maintain active jobs.

> To list all active jobs running in an active network

- On the **Active Job Networks** screen, type A in the line command input field next to the required network, and press ENTER.

An **Active Jobs** screen similar to the example below appears:

```

19.12.13          ***** Entire Operations *****          15:39:12
Active Jobs      Owner SN          Network A-1          Version v20131217
Selection AW_____ Run from 6182_ to 6182_
-----
C  Job          Run Typ Loc      JobId  Node  Date  Time Message
*-----
_  BBBBB        6182 JOB          N0148 18.12 14:46 Invalid JCL Location
_  BS2-0001     6182 JOB BS2      17GJ  N0194 14:47 Ended ok
_  CLASS-T0     6182 JOB PDS      40353 N0146 14:47 Job not run - JCL Erro
_  CLASS-T1     6182 JOB PDS      40354 N0146 14:47 Job not run - JCL Erro
_  CYC-1        6182 DUM          N0148 19.12 15:22 Dummy Job terminated
_  DEADLINE-1  6182 JOB PDS      40355 N0148 18.12 15:07 Ended ok
_  D1           6182 DUM          N0148 14:47 Dummy Job terminated
_  EMPTLINE     6182 JOB NAT      40356 N0148 14:47 Ended ok
_  EMPTLINE2    6182 JOB NAT      N0148 14:46 Invalid Job Card
_  FNF          6182 JOB PDS      N0148 14:46 JCL Load - ESY5998 Ele
_  HUGO-1       6182 JOB PDS      N0148 14:46 Symbol CLASS-K not fou
***** m o r e *****
A Sched.P C Canc D Deact E Edit G Gen.JCL H Hold I InCond J JCL K A.Res L Res.
M Mod. O EOJ P Prose R Resub. S SYSOUT T Stop U Rel. V RA W Wf Y SubC Z Subnet
Command => _____
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Add  End  ACond Save  ResUs Up    Down  ATask Left  Right Menu

```

This screen lists all active jobs in the network depending on your view authorizations (see also [User Restrictions for Active Jobs Lists](#)).

The jobs are sorted in the expected execution order of the network, derived from the logical dependencies. Jobs with the same execution order are sorted alphabetically.

The fields and columns available on the screen are explained in [Input Fields: Active Jobs](#) and [Columns: Active Jobs](#).

You can use line commands and PF keys on this screen to modify active jobs: see [Commands: Active Jobs](#).

Modifications of active jobs are effective for a specific run of a job only and do not affect any definitions made on the master database. This also applies to changes made to active job JCL, input conditions and End-of-Job checking and actions.

This section covers the following topics:

- [Input Fields: Active Jobs](#)
- [Columns: Active Jobs](#)

- [Commands: Active Jobs](#)

Input Fields: Active Jobs

The input fields on the [Active Jobs screen](#) are described in the following table:

Input Field	Description
Selection	You can restrict the list to jobs with special status values. For further information, see Listing Active Jobs by Process Status .
Run from	Enter a run number or enter an asterisk (*) to select a run number from a list. Default: the latest active run (in chronological order). The latest active run is set depending on the Last Run Display option set for the user in the Network Maintenance function (see the <i>Administration</i> documentation).
Run to	Enter a run number or enter an asterisk (*) to select a run number from a list. If the entered or selected Run to is lower than Run from , it will be set to the value of Run from . Default: the latest active run (in chronological order). Maximum run number: 99999. The latest active run is set depending on the Last Run Display option set for the user in the Network Maintenance function (see the <i>Administration</i> documentation).

Columns: Active Jobs

The following table explains the column headings for the data listed on the [Active Jobs](#) screen:

Column	Description						
C	One-character line command input field. For possible values, see the line commands in the section <i>Commands: Active Jobs</i> .						
(no heading)	The column without heading between C and Job indicates the current job type or status, and whether a job description exists. Possible indicators: <table border="1" data-bbox="386 1585 1474 1890"> <tbody> <tr> <td>D</td> <td>This is a dummy job. For more information, see Job Types in the section <i>Job Maintenance</i>.</td> </tr> <tr> <td>R</td> <td>This is a recovery job. For more information, see Defining Recovery Action in the section <i>End-of-Job Checking and Actions</i>.</td> </tr> <tr> <td>K</td> <td>This is a job for which a request for deactivation is pending in the Monitor task queue after a user initiated the deactivation of the job or job network.</td> </tr> </tbody> </table>	D	This is a dummy job. For more information, see Job Types in the section <i>Job Maintenance</i> .	R	This is a recovery job. For more information, see Defining Recovery Action in the section <i>End-of-Job Checking and Actions</i> .	K	This is a job for which a request for deactivation is pending in the Monitor task queue after a user initiated the deactivation of the job or job network.
D	This is a dummy job. For more information, see Job Types in the section <i>Job Maintenance</i> .						
R	This is a recovery job. For more information, see Defining Recovery Action in the section <i>End-of-Job Checking and Actions</i> .						
K	This is a job for which a request for deactivation is pending in the Monitor task queue after a user initiated the deactivation of the job or job network.						

Column	Description
	P This is a job for which a long description exists.
Job	Job name as defined to Entire Operations. For possible selection criteria, see <i>Filtering Items in a Result List in Using Entire Operations</i> .
Run	Job run number automatically assigned at activation time. The job run number uniquely identifies an active copy of a job together with the job name.
Typ	Job type as defined to Entire Operations.
JobId	Job identifier as assigned by the operating system or by the job entry subsystem.
Node	Execution node of the machine designated for the job.
Date	Last action or check date for the job. See also <i>Date and Time Formats</i> .
Time	Last action or check time for the job. See also <i>Date and Time Formats</i> .
Message	Last message issued for the job by Entire Operations. Press PF11 (Right) or PF10 (Left) to display the full message text. For a list of possible messages and their meaning, see <i>Messages in Active Jobs Display</i> in the <i>Messages and Codes</i> documentation.

Commands: Active Jobs

Use the following line commands to perform the described functions on the active jobs listed on the [Active Jobs](#) or [All Active Jobs](#) screen:

Line Command	Description
A	For active networks: Modify scheduling parameters for the active job . For planned network runs: Modify planned network start time .
B	Browse for the selected active job. See Browsing Active JCL .
C	Cancel the active job from the operating system. See Canceling an Active Job .
D	For active networks: Deactivate the selected job from the active database, including any definitions made at job level (conditions, End-of-Job handling etc.). For planned network runs: <i>Cancel the planned job activation</i> .
E	Edit the activated JCL for the selected active job. See Editing Active JCL .
F	Release edit lock. This function can only be executed by administrators.

Line Command	Description
	Removes a lock from active JCL held on the source by a user who previously edited it. The unlock action is archived in the Entire Operations log. Important: This function is for emergency use only. Data loss can occur when you remove the lock from a source while it is being edited by several concurrent users. See also <i>Locking of Natural Sources</i> in the section <i>Job Maintenance</i> .
G	(Re)generate the dynamic JCL for this activation. See <i>Regenerating Active JCL</i> .
H	Set active job in hold status. See <i>Holding an Active Job</i> .
I	Modify the defined input conditions for the active job. See <i>Listing Active Conditions</i> .
J	Define JCL for the active job. See <i>Defining JCL for an Active Job</i> .
K	View the resources allocated for the active job. See <i>Viewing and Modifying Resources Used by Active Jobs</i> .
L	Maintain resources for the active job. See <i>Viewing and Modifying Resources Used by Active Jobs</i> .
M	Modify definition for the active job. See <i>Displaying and Modifying an Active Job Definition</i> .
O	Modify End-of-Job checking and actions for the active job. See <i>Modifying EOJ Checking and Actions</i> .
P	Text description of the job. See <i>Viewing Long Descriptions of Active Jobs</i> .
R	Resubmit active job. See <i>Resubmitting an Active Job</i> .
S	Browse job SYSOUT. See <i>Browsing Active Job SYSOUT</i> .
T	Stop cyclic job (special type C only). This sets the reserved condition P-STOPCYC- <i>jobname</i> to terminate the cyclic execution.
U	Release active job from hold status. See <i>Releasing an Active Job</i> .
V	Reactivation of an active job. See <i>Reactivating an Active Job</i> .
W	(Waiting for Activation) For active jobs: Display all pending prerequisites for the active job. For active networks awaiting symbol prompting: Invoke symbol prompting (see also <i>Symbol Prompting during Network or Job Activation</i>).
Y	Show the calling job (if this active job is part of an active subnetwork). See <i>Viewing Calling Jobs of Subnetworks</i> .
Z	List jobs in a subnetwork (for jobs of the type NET only). See <i>Listing Jobs of a Subnetwork</i> in the section <i>Job Maintenance</i> .

Use the following special PF keys to perform the described functions on the active jobs listed on the **Active Jobs** or **All Active Jobs** screen:

PF Key	Name	Function
PF2	Add	Active Jobs screen: Add an active job definition. All Active Jobs screen: Select an active node from a window.
PF4	ACond	Maintain active conditions.
PF9	ATask	Display operating system information on active tasks.
PF12	Menu	Return to Entire Operations Main Menu.

The following section describes in more detail the functions you can perform on active jobs using line commands and PF keys (explanations are not necessarily given in the same order as the list of associated line commands above).

Listing Active Jobs by Process Status

You can list active jobs according to their processing status by entering one or more of the following values in the **Selection** field on the **Active Jobs** or **All Active Jobs** screen and pressing ENTER:

Value	Description
A	All active jobs (default).
E	All active jobs waiting for at least one event.
H	All jobs in hold.
I	All jobs in the scheduling system input queue.
N	All jobs with job not ok terminating status.
O	All jobs with job ok terminating status.
S	All submitted jobs.
T	All terminated jobs.
W	Planned network runs waiting for activation.
X	All jobs in execution.

The default setting is AW (all active jobs and the planned network runs waiting for activation).

Alternatively, you can enter a question mark (?) in the **Selection** field and press ENTER to open the following selection window:

```

09.02.16          ***** Entire Operations *****          12:12:02
                    All Active Jobs
Selection ? _____ Submit User ID _____
-----+-----+-----+
C Owner          |          | obId  Time Message
  EXAMPLE       | Status |          |          |
_ EXAMPLE       | Preselection
_ EXAMPLE       |          |          |          |
_ EXAMPLE       | W      |          |          |
_ EXAMPLE       | A      |          |          |
_ EXAMPLE       | E      |          |          |
_ EXAMPLE       | H      |          |          |
_ EXAMPLE       | S      |          |          |
_ EXAMPLE       | I      |          |          |
_ EXAMPLE       | X      |          |          |
_ EXAMPLE       | T      |          |          |
_ EXAMPLE       | O      |          |          |
_ EXAMPLE       | N      |          |          |
*****         | *      |          |          |
A Sched.P       | All Jobs
M Mod. 0        |          |          |          |
Command =      |          |          |          |
-----+-----+-----+
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Add   End   ACond Save  ResUs Up    Down  ATask Left  Right Menu

```

Enter one or more values in the **Select ==>** field and press ENTER to list the active jobs with the selected status.

Listing Active Jobs According to Run Numbers

By default, only jobs of the last run are displayed on the **Active Jobs** screen.

> To list specific job runs

- Enter the required start and end numbers in the **Run from/to** input fields of the **Active Jobs** screen, and press ENTER.

For example, if you enter 7 in the **Run from** field, all jobs with run number equal to or greater than 7 are listed. If you enter 4 in the **Run from** field and 7 in the **to** field, all jobs with run numbers 4, 5, 6 and 7 are listed.

This setting is kept until you modify it or until you display the **Active Jobs** screen for another network.

Listing Jobs of an Active Subnetwork

➤ To list jobs of an active subnetwork

- Proceed as described in *Listing Jobs of a Subnetwork in the section Job Maintenance*.

Adding a New Job to the Active Network

➤ To add a job to the active database for the current run of an active job network

- 1 Press PF2 (Add) on the **Active Jobs screen**.

A **Job Definition (active)** window opens where you can define the new active job.

- 2 Enter a new job definition for the current network run.

The **fields** and **special PF keys** available in the **Job Definition (active)** are described in *Displaying and Modifying an Active Job Definition*.

- 3 Press PF5 (Save) to add the active job.
- 4 Press PF3 (End) to close the **Job Definition (active)** window.

The new active job is now listed on the **Active Jobs screen**.

Adding a job to an active job network involves temporarily changing the network structure and job flow. This is achieved with the minimum of effort, as the following **example** shows.

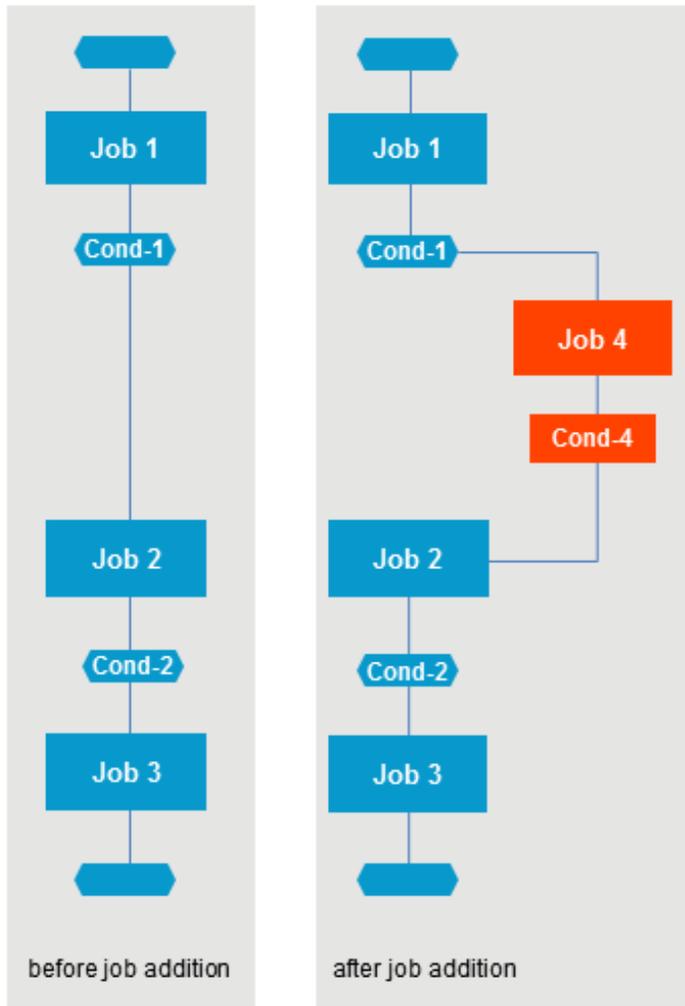
This section covers the following topics:

- [Example](#)
- [Achieving a Sequential Flow](#)

Example

You may wish to add a job to a network for one specific run.

This example illustrates the addition of a job to a network consisting of 3 sequential jobs. The new job (Job 4) is to be inserted between Job 1 and Job 2:



 condition

Achieving a Sequential Flow

➤ To achieve a sequential job flow

- 1 Add the new active job using the **Job Definition (active)** window;
- 2 Define Cond-1 as input condition for Job 4;
- 3 Define Cond-4 as output condition in End-of-Job handling for Job 4;
- 4 Change the input condition defined for Job 2 (Cond-1) to Cond-4.

Job 4 is executed for the current run of the job network only.

If you wish this change to be permanent, use this procedure on the master database.

Viewing and Modifying the Active Jobs Schedule

➤ To list active jobs schedules

- On the **Active Job Networks** screen, type S (Act.Jobs Schedule) in the line command input field next to the required network, and press ENTER.

An **Active Jobs Schedule** screen similar to the example below appears:

```

17-02-24          ***** Entire Operations *****          16:13:39
Owner  EXAMPLE          Active Jobs Schedule          Network E01-CONTI
Selection *          Run from 2____ to 2____
-----
C  Job              Run Type Latest Start Deadline      Message
*-----*-----*-----*-----*-----*-----*-----*
_  E01-J01          2 JOB   02-25 16:10  02-26 16:10  Ended ok
_  E01-J02          2 JOB   02-25 16:10  02-26 16:10  Ended not ok - STEP02 C001
_  E01-J05          2 JOB   02-25 16:10  02-26 16:10  JobId 57487 executing (sub

***** Bottom of Data *****
A Sched.P  M Modify Latest Start

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      End              Up      Down  ChLat Left  Right
    
```

This screen lists all active jobs in the network.

You can use [line commands and PF keys](#) to modify single and multiple jobs.

Modifications of jobs on the **Active Jobs Schedule** screen are effective for a specific run of a job only and do not affect any definitions made on the master database. This also applies to changes made to active job JCL, input conditions and End-of-Job checking and actions.

This section covers the following topics:

- [Columns: Active Jobs Schedule](#)
- [Commands: Active Jobs Schedule](#)
- [Modify Latest Start Time](#)

Columns: Active Jobs Schedule

The following table explains the columns for the data listed on the [Active Jobs Schedule screen](#):

Column	Description
C	One-character line command input field. For possible values, see the line commands in Commands: Active Jobs Schedule .
(no heading)	There is another column without a heading between C and Job . Possible values:
D	This is a dummy job. For more information, see Job Types in the section <i>Job Maintenance</i> .
R	This is a recovery job. For more information, see Defining Recovery Action in the section <i>End-of-Job Checking and Actions</i> .
Job	Job name as defined to Entire Operations. For possible selection criteria, see <i>Filtering Items in a Result List</i> in <i>Using Entire Operations</i> .
Run	Job run number automatically assigned at activation time. The job run number uniquely identifies an active copy of a job together with the job name.
Type	Job type as defined to Entire Operations.
Latest Start	Latest start time of the job.
Deadline	Deadline time of the job.
Message	Last message issued for the job by Entire Operations. Press PF10 (Left) or PF11 (Right) to display the full message text. For a list of possible messages and their meaning, see <i>Messages in Active Job Display</i> in the <i>Messages and Codes</i> documentation.

Commands: Active Jobs Schedule

The following line commands are available on the **Active Jobs Schedule** screen:

Line Command	Description
A	Active scheduling parameters. See also <i>Modifying Scheduling Parameters</i> .
M	Modify latest start time.

The following special PF key is available on the **Active Jobs Schedule** screen:

PF Key	Name	Function
PF9	ChLat	Change latest start (for all jobs).

Modify Latest Start Time

The latest start time may be changed for an active job that is not started or is already terminated.

You can use either of the following methods to modify the latest start time:

- **Set:** Enter the date and time of the new latest start;
- **Add:** Enter the days, hours and minutes to be added to the latest start time of the active job.

The deadline value can change during this modification to guarantee that the following rule is met: latest start time + estimated elapsed time <= Deadline.

All actions are logged in the Entire Operations log file.

> To modify the latest start time for active jobs

- 1 On the **Active Jobs Schedule screen**, enter the line command M next to the job(s) for which you want to change the start time.

Or:

On the **Active Jobs Schedule screen**, press PF9 (ChLat) to change the start times of all jobs scheduled for the network run.

```

2  29.10.08          ***** Entire Operations *****                               14:09:30
   Owner SN                Active Jobs Schedule                               Network A-2
   Selection * +-----+-----+-----+-----+-----+-----+-----+-----+
   ----- !----- !----- !----- !----- !----- !----- !----- !-----
   C Job      !      Modify Latest Start Time      !      Message
   *----- !      22.06      22:59:00              !      -----
   _ HUG0-1   !      for HUG0-2                      !      JobId 43825 - Symbol Repla
   m D HUG0-2 !      !                                  !      WORKDAY - 0 - DAT not foun
   _ D HUG0-3 !      Date      Time                  !      WORKDAY - 0 - DAT not foun
   _          !      Set  29.10.08 00:00:00          !      !
   _          !      !                                  !      !
   _          !      or                               !      !
   _          !      !                                  !      !
   _          !      Add  ___ days ___ hr ___ min      !      !
   _          !      !                                  !      !
   _          !      ---PF1---PF3-----                !      !
   _          !      Help  End                               !      !
   _          +-----+-----+-----+-----+-----+-----+-----+-----+
   ***** Bottom of Data *****
   A Sched.P  M Modify Latest Start

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      End                               Up      Down  ChLat Left  Right
  
```

Canceling, Holding and Releasing Active Jobs

- [Operating System Jobs Handling](#)
- [Canceling an Active Job](#)
- [Holding an Active Job](#)
- [Releasing an Active Job](#)

Operating System Jobs Handling

Operating system jobs in the Entire Operations active database can be handled directly from the [Active Jobs](#) or [All Active Jobs](#) screen.

When Entire Operations submits jobs to the operating system, they are assigned a job number which appears, for example, on the [Active Jobs](#) or [All Active Jobs](#) screen. In z/OS systems, JES operator commands are issued.



Note: In BS2000, console commands are issued.

Canceling an Active Job

You can cancel the planned activation of a job in a planned network run that is waiting for activation. The job is then not executed within the active network.

➤ **To cancel an active job from the operating system**

- 1 Enter C (Canc) in the line command input field of the appropriate job on the **Active Jobs** or **All Active Jobs** screen.
- 2 Press ENTER.
- 3 A window opens in which you can confirm the cancellation request:

```

10.02.16          ***** Entire Operations *****          18:04:27
Active Jobs      Owner EXAMPLE   Network E01-C0-148 Version
Selection AW_____ Run from 23__ to 23__
-----
C  Job          Run Typ Loc   JobId  Node  Date  Time Message
*-----*
_  E01-J01      23 JOB NAT   58452  N0148 05.02 14:46 Ended ok
_  E01-J02      23 JOB NAT   58453  N0148      14:46 Ended not ok - STEP02
c  E01-J03      23 JOB NAT   58454  N0148      14:46 Ended not ok - STEP03
_  E01- +-----+ EP03
_  E01- |
_  E01- | Please confirm
    |   | the Cancelling of E01-J03
    |   | by entering 'Y' ==> _
    |   |
    |   | PF3 End
    |   +-----+
***** Bottom of Data *****
A Sched.P C Canc D Deact E Edit G Gen.JCL H Hold I InCond J JCL K A.Res L Res.
M Mod. O EOJ P Prose R Resub. S SYSOUT T Stop U Rel. V RA W Wf Y SubC Z Subnet
Command => _____

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Add End ACond Save ResUs Up Down ATask Left Right Menu
    
```

- 4 Enter the name of the job to be cancelled and press ENTER.

The specified job is cancelled.

For a job listed when W (Waiting for Activation) is entered in the **Selection** field, the planned activation of the job in a planned network run is cancelled.

Notes:

1. A cancelled job cannot be released for further processing.

2. For BS2000 jobs, you can only use the cancel command if the following conditions are fulfilled:

- The BS2000 user ID in effect is allowed to issue a cancel command for the Entire System Server node(s) in use. If required, specify the following in the Entire System Server parameter file and restart Entire System Server:

```
SYSTEM-COMMAND-USER=TSOS, user-id [, ...]
```

For example:

```
SYSTEM-COMMAND-USER=TSOS, SAG1, SAG2, SAG3
```

- The active and passive Entire System Server console servers are available (see the Entire System Server *User's Guide*).

3. UNIX jobs are cancelled by the SIGTERM signal.

Holding an Active Job

You can stop an active job and place it in hold status.

➤ To stop a job and place it in hold status

- 1 Enter H in the line command input field of the appropriate job on the [Active Jobs](#) or [All Active Jobs](#) screen.
- 2 Press ENTER.

The message ==> H-Cmd executed <== appears in the **Message** column for the job.

- 3 Press ENTER again.

The message Job set to Hold appears in the **Message** column for the job.



Notes:

1. If the job has already been submitted to the operating system, it will be held in the operating system.
2. Operating system hold does not work for Windows jobs.
3. For UNIX jobs, the whole process group belonging to the job ID will be held.

Releasing an Active Job

You can release a job from hold status.

➤ To release a job from hold status and allow it to continue

- 1 Enter `U` in the line command input field of the appropriate job on the [Active Jobs](#) or [All Active Jobs](#) screen.
- 2 Press `ENTER`.

The message `==> U-Cmd executed <==` appears in the **Message** column for the job.

- 3 Press `ENTER` again.

The message `Job released from Hold` appears in the **Message** column.



Notes:

1. If the command is used for a job of the type [NET](#) (subnetwork): during a release, no new activation of the subnetwork will be performed.
2. The release from hold command is available for networks in the status `to be activated too`.

Resubmitting an Active Job

After a job has terminated, you can modify and resubmit it while it is still in the active database. This function is useful after a job has failed.

For example, if a JCL error has occurred, you can modify the active JCL (see [Editing Active JCL](#)) and resubmit the job. The resubmitted job uses the old input conditions for submission, but can set different output conditions, depending on the End-of-Job analysis.

Any output conditions set by the job's first run can be reset by the second run if this feature is defined in the job's original End-of-Job handling.

➤ To resubmit an active job

- 1 On the [Active Jobs](#) or [All Active Jobs](#) screen, type `R` in the line command input field next to the required job, and press `ENTER`.

The following window opens:

```

18-02-02          ***** Entire Operations *****          10:23:31
Active Jobs      Owner EXAMPLE      Network E20-DYN-01 Version
Selection A_____ Run from 4_____ to 4_____
-----
C  Job          Run Typ Loc   JobId  Node  Date  Time Message
*-----
R  E20-J01      4 JOB MAC          148 02-02 10:23 MAC Exit SYSEORU/E51-M

          +-----+
          !
          !           Resubmission of Job E20-J01           !
          !
          ! Please select:                                   !
          ! re-load active JCL ..... ==> _ !
          ! with submission symbol replacement ==> _ !
          ! unchanged JCL ..... ==> _ !
          !
          ! PF1 Help   PF3 End                               ! *****
*****
A Sc.P B Brw C +-----+ L K A.Res L Res.
M Mod. 0 EOJ P Prose R Resub. S SYSOUT T Stop U Rel. V RA W Wf Y SubC Z Subnet
Command => _____

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Add  End  ACond Save  ResUs Up    Down  ATask Left  Right Menu

```

- 2 Select either option with any character and press ENTER to confirm the resubmission.
- 3 The window closes and the resubmission time with the message Job is resubmitted appears in the message field of the **Active Jobs** or **All Active Jobs** screen.

This section covers the following topics:

- [Options for Resubmission](#)
- [Resubmitting an Active Job of NET type \(Subnetwork\)](#)
- [No Reloading after Editing](#)
- [Job Resubmission under BS2000](#)

Options for Resubmission

You have one of the following options to perform the resubmission:

re-load active JCL	The active JCL will be reloaded.
with submission symbol replacement	The active JCL will be used again, and the submission symbol replacement (using the submit escape character) will be repeated. Active symbols created at submission time will be deleted prior to a resubmission. This forces the recreation of these active symbols during the resubmission.
unchanged JCL	The job will be resubmitted exactly as previously submitted; without any symbol replacement.



Notes:

1. Jobs of types **NAT** (Natural Program) and **STC** (Started Task) are resubmitted unchanged anyway.
2. See also the Entire Operations default setting **Submit symbol/function recalculation at resubmit** on the **Default Setting (3)** screen described in the *Administration* documentation.

Resubmitting an Active Job of NET type (Subnetwork)

If the line command **R** (resubmit) is used for jobs with **NET** type (subnetwork), the subnetwork will be repeated with the same subnetwork run number. However, this is only allowed if the subnetwork has not started yet.

No Reloading after Editing

In case of a resubmission of the active job, the **pregenerated JCL** is not reloaded if the active JCL of the job has been edited in the meantime.

Job Resubmission under BS2000

The SYSOUT file copy attempts are limited if the SYSOUT file is locked. The limit is 10 times the task (Monitor) wait time. If the task wait time is smaller than 30 seconds, the limit is $10 * 30 = 300$ seconds.

If the SYSOUT file is still locked when the limit is reached, the active job will be set to a permanent error status, with the message text 10 SYSOUT Renaming Attempts failed.

For each unsuccessful attempt, the message text will be set to SYSOUT File Renaming - File in use. In addition, the event will be logged with the SYSOUT file name.

Deactivating a Job in an Active Network

➤ To prevent the execution of a certain job in an active network for the current run

- 1 On the **Active Jobs** or **All Active Jobs** screen, type D in the line command input field of the selected job, and press ENTER.

A window opens in which you must confirm deactivation by typing Y in the appropriate field:

```

22.02.16          ***** Entire Operations *****                               11:47:36
Active Jobs      Owner EXAMPLE      Network B60-FLOW      Version
Selection AW_____ Run from 2167_ to 2167_
-----
C  Job          Run Typ Loc   JobId  Node  Date  Time Message
*-----*-----*-----*-----*-----*-----*-----*-----*-----*
d  JOB-05       2167 DUM                N0031 22.02 10:21 E60-JOB4-0 - 2167 - RU
_  JOB-06       2167 JOB MAC          N0031      10:30 E60-JOB5-0 - 2167 - RU

      +-----+
      |           |
      | Please confirm |
      | the Deactivation of JOB-05 |
      | by entering 'Y' ==>> _ |
      |           |
      | PF3 End       |
      +-----+

***** Bottom of Data *****
A Sched.P C Canc D Deact E Edit G Gen.JCL H Hold I InCond J JCL K A.Res L Res.
M Mod. O EOJ P Prose R Resub. S SYSOUT T Stop U Rel. V RA W Wf Y SubC Z Subnet
Command => _____

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Add End ACond Save ResUs Up Down ATask Left Right Menu
  
```

- 2 Press ENTER to deactivate the job.

Deactivation is performed by the Entire Operations Monitor in the background. Deactivating a job from the active network involves deleting it from the active job network and restructuring the active network in order to ensure uninterrupted job flow. The following **example** illustrates job deactivation by bypassing a job in an active network consisting of three sequential jobs.

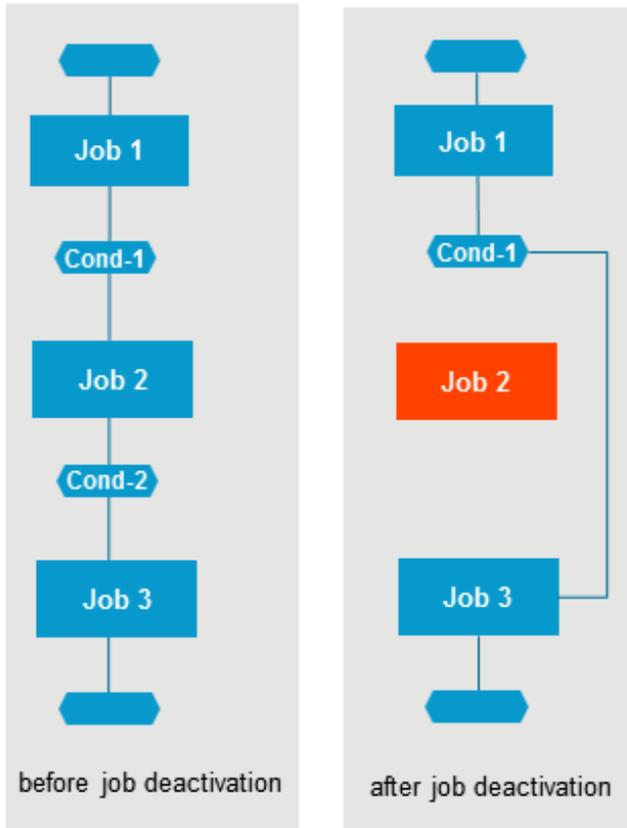
This section covers the following topics:

- [Example: Deactivation of a Job](#)

- Ensuring Uninterrupted Job Flow

Example: Deactivation of a Job

The network in the following figure is to run without Job 2:



Ensuring Uninterrupted Job Flow

› To ensure uninterrupted job flow

- 1 To prevent Job 2 from running, it is sufficient to delete its input condition Cond-1. If you deactivate Job 2 from the **Active Jobs** or **All Active Jobs** screen, the input condition is automatically deleted.;
- 2 Replace the input condition for Job 3 (Cond-2) by the input condition Cond-1.

The current run of the job network skips Job 2.

If you wish to deactivate a job that is currently running (interrupt its execution), you must cancel it from the operating system before deactivation (see *Canceling an Active Job*).

Reactivating an Active Job

This function is used to deactivate an active job and then reactivate it completely.

This function can be used to restore an inconsistent active job (for example, if a symbol was missing during the original activation).

› To reactivate an active job

- 1 On the **Active Jobs** or **All Active Jobs** screen, type V in the line command input field of the selected job, and press ENTER.

A confirmation window prompts you to confirm the reactivation of the selected job.

- 2 Type Y and press ENTER to reactivate the job and close the window.

Reactivation is performed in the background by the Entire Operations Monitor, whereby:

- A prerequisite check is performed.

If the job does not wait for any prerequisite, it is submitted immediately.

- All input and output definitions are recreated.
- The active JCL is generated newly.

Displaying and Modifying an Active Job Definition

➤ To view and modify the definition of an existing job in an active job network

- 1 On the **Active Jobs** or **All Active Jobs** screen, type the line command M next to a job listed, and press ENTER.

A **Job Definition (active)** window similar to example below opens:

```

24.11.16          ***** Entire Operations *****          11:40:06
Active Jobs      Owner EXAMPLE      Network B60-FLOW      Version
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Job Definition (active)                                     |
| Job Name      ==> JOB-01_____      Activated      24.11.16 06:30      |
| Run           ==> 2382                Modified SAG      08.11.16 17:39      |
| Description   ==> Where it all starts_____      |
| Job Type     ==> JOB                    Network ==> B60-FLOW      |
| Execution Node ==> N0082 MVS/ESA        Version ==> (unnamed)      |
| Special Type  ==> _____          Symbol Table ==> EXAM-ST1__      |
| Milestones   ==> _____          Symbol Table Version ==> _____      |
| Restartable  ==> _                    |
| Escape Characters:      Activation ==> @ Submit ==> $      |
| End-of-Job Action Errors set 'not ok' Conditions ==> N      |
| Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9-----PF12---      |
|           Help Add  End  Edit Save Spec Symb Net  JCL      Menu      |
+-----+-----+-----+-----+-----+-----+-----+-----+
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
           Help Add  End  ACond Save ResUs Up   Down  ATask Left  Right Menu
    
```

The window displays all current values of the job. You can modify any job parameter.

- 2 Press PF4 (Edit) to edit any JCL or Natural program according to the **job type**.
- 3 Press PF5 (Save) to save all changes.
- 4 Press PF3 (End) to close the window and return to the **Active Jobs** screen.

These changes only affect the current run of the job.

The following applies when you modify a job definition in an active job network definition:

- The current user ID is stored as the last modifying user of the job for all definition and JCL modifications. This user ID is taken as the submit security user ID. If the Monitor does not use its own user ID, see *Monitor Defaults - Submit Security User Type* in the *Administration* documentation.
- It may be necessary to release the active job after the modification with the line command **R** (resubmit). This causes a new check of all prerequisites.

This section covers the following topics:

- [Fields: Job Definition \(Active\)](#)
- [Special Job Type D - Execution as a Dummy](#)
- [Special PF Keys: Job Definition \(Active\)](#)

Fields: Job Definition (Active)

The input fields in the **Job Definition (active)** window mostly have the same meaning as in the **Master Job Definition** window of the **Network and Job Maintenance** function; see [Fields: Job Definition \(Master\)](#) in the section *Job Maintenance*. However, the **Job Definition (active)** window contains the following additional fields which are protected:

Field	Description
Activated	Activation date and time of the network.
Modified	User ID and time stamp of the last modification.
Run	Run number of the current job run.

Special Job Type D - Execution as a Dummy

In the **Job Definition (active)** window, it is possible to modify the field **Special Type** from empty to **D** (execute as dummy), and vice versa.

- If **D** is removed, an automatic reloading of the JCL is performed. In case of active jobs of the type **NET**, the subnetwork is activated.
- If **D** is set, the job is executed as a dummy job. In this case, it does not play any role, whether already active JCL or an active subnetwork were loaded.

See also: [Job Execution as Dummy Job](#).

Special PF Keys: Job Definition (Active)

PF Key	Name	Function
PF2	Add	Add an active job to the active network. No master job definition is created. JCL is automatically loaded after the job has been added. A job is not automatically activated after being added to an active network. The job is first put into hold status and additional definitions can be added, for example, active time frames. To activate the job, you must enter the line command U on the Active Jobs or All Active Jobs screen.
PF4	Edit	Edit JCL or a Natural program according to the job type .
PF7	Symb	Open the Usable Symbol Tables window and select a symbol table for browsing or modification. See Listing Usable Symbol Tables .
PF9	Spec	Define special parameters for operating system dependent job definitions.

Modifying EOJ Checking and Actions

➤ To modify the End-of-Job checking and actions for the current run of a specific job

- On the [Active Jobs](#) or [All Active Jobs](#) screen, type 0 in the line command input field next to the required job, and press ENTER.

An [End-of-Job Checking + Actions screen](#) similar to the example shown in the section [End-of-Job Checking and Actions](#) appears showing the current definitions. You can modify event and action definitions in the same way as described in this section.



Note: Any modifications made to End-of-Job handling parameters of an active job are valid for the current job run only.

Viewing Long Descriptions of Active Jobs

➤ To view the long text description of an active job

- On the [Active Jobs](#) or [All Active Jobs](#) screen, type P in the line command input field next to the required job, and press ENTER.

An [editor screen](#) similar to the example shown in the section [Job Maintenance](#) appears.

The screen contains the long description of the active job, provided a long description exist for the job master. The description is displayed in Editor format and you can use Editor commands and

PF keys to browse the text. No modification is possible. All text descriptions can only be modified on the master database by using the appropriate job maintenance function as described in [Writing and Viewing Online Documentation for a Job](#).

Displaying Prerequisites for Active Jobs

The message field on the [Active Jobs](#) or [All Active Jobs](#) screen shows only one pending prerequisite (conditions, resources, scheduled times, etc.) for which a job is waiting.

For a complete list of all pending prerequisites, proceed as described in the following instruction.

➤ To display all waiting prerequisites for an active Job

- On the [Active Jobs](#) or [All Active Jobs](#) screen, type W in the line command input field next to the required job, and press ENTER.

A **Prerequisites** window opens with a list of events for which the active job is waiting:

```

17-04-14          ***** Entire Operations *****          10:30:03
Active Jobs      Owner EXAMPLE      Network E60-FLOW      Version
Selection AW_____ Run from 4786_ to 4786_
-----+-----+
!
! Job JOB-012   Run  4786      17-04-14 10:30      Network E60-FLOW      !
! is waiting for the following Prerequisites:      !
!
! _ Job is in passive wait since 19.03 14:43      !
! _ E60-J015-0 - 4786 - RUN not found      !
! _ E60-J019-0 - 4786 - RUN not found      !
!
!
!
!
!
! R Reset Condition  S Set Condition  W Where Used      !
! C Force Prerequisite Check      !
! ---PF1-----PF3-----PF7---PF8-----      !
!   Help           End             Up    Down      !
+-----+-----+
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Add   End   ACond Save  ResUs Up    Down  ATask Left  Right Menu

```

The listed message means that the current run of the job (run number 4786) is waiting for the following:

- The setting of the input condition E60-J015-0 from the same run. See *Messages in Active Jobs Display* in the *Messages and Codes* documentation for a full list of possible messages and their meaning.
- The setting of the input condition E60-J019-0 from the same run. See *Messages in Active Jobs Display* in the *Messages and Codes* documentation for a full list of possible messages and their meaning.

The text `Job is in passive wait since 19.03 14:43` indicates that the job has been in a passive wait for input conditions.

In this case, the job is automatically returned to the active wait by setting one of the input conditions, i.e., the Monitor will execute a prerequisite check.

Before this message, you can force an active prerequisite check using the line command C.

All line commands available in the **Prerequisites** window are described in [Line Commands: Prerequisites for an Active Job](#).

Line Commands: Prerequisites for an Active Job

The following line commands are available in the **Prerequisites** window:

Command	Description
C	Force an active prerequisite check. Only before the message <code>Job is in passive wait since ...</code>
R	Reset: <ul style="list-style-type: none"> ■ If used for a condition wait: reset the condition to false. ■ If used for a resource wait: delete the active prerequisite resource definition. The active job will then no longer wait for the resource. A window opens in which you can confirm the action by entering the condition name and pressing ENTER.
S	Set: <ul style="list-style-type: none"> ■ Used for a condition wait: set the condition to true. A window opens in which you can confirm the action by entering the condition name and pressing ENTER.
W	Where used: <ul style="list-style-type: none"> ■ If used for a condition wait: display the usage of the active condition. For further information, see Viewing the Usage of Conditions Concatenated with an Active Job.

Command	Description
	<ul style="list-style-type: none"> ■ If used for a resource wait: display the usage of the resource. For further information, see Viewing and Modifying Resources Used by Active Jobs.

If the prerequisite is an input condition, you can set or reset the condition manually by typing **S** or **R** in the line command input field of the selected condition.

- Use the line command **S** to set the condition to true;
- Use the line command **R** to reset the condition to false.

In either case, a window opens in which you can confirm the manual (re)setting by entering the condition name. Type in the condition name and press **ENTER** to confirm the operation and close the window.

Viewing and Modifying Resources Used by Active Jobs

You can display the resources used by active jobs.

You can also modify resources for the current run of an active job without changing the original resource specification on the master database for future job runs. This can be useful if different job runs require different amounts of a specific resource, for example paper or tape drives.

➤ To list resources defined as prerequisites for an active jobs

- 1 On the **Active Jobs** or **All Active Jobs** screen, type **L** in the line command input field next to the required job, and press **ENTER**.

A **Prerequisite Resources (active)** screen appears with a list of resources allocated to the active job.

This screen corresponds to the **Prerequisite Resources (Master) screen** which is explained in the section *Job Maintenance*. For explanations of the columns and fields provided on the screen, see [Columns and Fields: Prerequisite Resource Definitions \(Job Maintenance\)](#).

- 2 If you type **W** next to a resource listed on the **Prerequisite Resources (active)** screen and press **ENTER**, an **Active Resource Usage** screen appears. This screen is described in *Listing Jobs Currently Using a Resource* in the *Administration* documentation.

➤ To list resources currently allocated by an active job

- On the **Active Jobs** or **All Active Jobs** screen, type **K** in the line command input field next to the required job, and press **ENTER**.

An **Allocated Resources (active)** screen similar to the example below appears:

```

18-01-25          ***** Entire Operations *****          17:35:30
                   Allocated Resources (active)
-----
Owner SAGTEST      Network B60-FLOW   Version                               Job JOB-05
Run from 105      to 105                               required
Cmd  Run Job      Resource           T  Quantity  D  DNO   allocated
_    105 JOB-05   TESTRESOURCE       R    19.00   J    01-25 13:39

***** Bottom of Data *****
D Delete  M Modify  R Master Definition  W Where Used

Enter--PF1----PF2----PF3----PF5-----PF7----PF8-----
      Help  Add    End    Save      Up    Down
    
```

The screen lists all resources currently allocated by the active job as indicated by the date and time in the allocated column. The information provided on the screen corresponds to the information provided on the **Prerequisite Resources (active)** screen.

➤ **To modify resources for an active job**

- On the **Prerequisite Resources (active)** screen, enter M next to the resource you want to modify, and press ENTER.

A **Prerequisite Resource Definition (active)** window opens.

This window corresponds to the **Prerequisite Resource Definition (Master) window**. You can modify the resources for the active job in the same way as described in *Displaying, Modifying and Adding a Prerequisite Resource Definition* in the section *Job Maintenance*.

Modifying Scheduling Parameters

You can modify the scheduling parameters for a specific job run without changing the originally defined job schedule table

➤ To modify scheduling parameters for an active job

- On the [Active Jobs](#), [All Active Jobs](#) or [Active Jobs Schedule](#) screen, type A in the line command input field next to the required job, and press ENTER.

An **Active Scheduling Parameters** window opens displaying the current schedule values in the same format as defined for the job on the master database.

The fields and special PF keys available in the window correspond to the [fields](#) and [PF keys](#) in the [Scheduling Parameters](#) window of a job master. You can modify the schedule in the same way as described in [Scheduling a Job](#) in the section [Job Maintenance](#).

Browsing Active Job SYSOUT

You can view SYSOUT of an active job depending on the operating system installed at your site:

Operating System	SYSOUT Availability
z/OS	SYSOUT can be browsed immediately after job submission. It is readable until the current last line readable from the spooling system (for example, JES2).
BS2000	The SYSOUT file can be browsed only after job termination. This is because the SYSOUT file is open during execution and cannot be read.
UNIX/Windows	The SYSOUT file can be browsed immediately after job submission. It is readable up to its current last line.

This section covers the following topics:

- [Viewing Job SYSOUT](#)
- [SYSOUT File Types on z/OS and z/VSE](#)

- [SYSOUT Special Commands on z/OS and z/VSE](#)

Viewing Job SYSOUT

> To view job SYSOUT

- 1 On the [Active Jobs](#) or [All Active Jobs](#) screen, type S in the line command input field next to the required job, and press ENTER.

If the job has been resubmitted or rescheduled several times, several job IDs can exist for the selected job. If this is the case, a window opens with a list of job IDs in the following format:

```

16-09-25          ***** Entire Operations *****          10:44:59
Active Jobs      Owner NATQA5      Network NET2522-B  Version v1
Selection AW_____ Run from 2____ to 2____
-----+-----+-----
C  Job          Run Typ  !                               !
 *-----      ---  !  Owner NATQA5      Network NET2522-B  !
s  JOB-ACT01    2 JOB  !  Run 2          Job  JOB-ACT01  ! <==
_  JOB-ACT02    2 JOB  !                               ! req
                               ! Select a Job Id:   !
                               !                   !
                               !                   Repeat   !
                               ! Current:  X      63346    1      !
                               ! Previous: _      63335    0      !
                               !                   !
                               !                   !
                               !                   !
                               !                   !
*****          ! PF3 End PF7 Up PF8 Down          ! ***
A Sched.P C Canc D Deact +-----+ es.
M Mod.  O EOJ P Prose R Resub. S SYSOUT T Stop U Rel. V RA W Wf Y SubC Z Subnet
Command => _____

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Add   End   ACond Save  ResUs Up    Down  ATask Left  Right Menu
    
```

The top half of the window identifies the job by owner, job network, network version, job name and run number, the bottom half contains a list of job IDs; the job ID assigned for the current run is at the top, followed by previous runs.

If there are more than 5 previous executions, you can scroll through the list with PF7 (Up) and PF8 (Down).

- 2 Mark the job ID of the run for which you wish to see the SYSOUT file with any character, and press ENTER.

If only one SYSOUT file exists for the selected job, this window is bypassed and the SYSOUT file is displayed immediately.

In browse mode, the SYSOUT file is displayed in the editor and you can use editor PF keys and browse commands:

```

JobId DUMMY (63346) Type SM File 1----- Columns 001 072
====>                                     SCROLL==> CSR
***** ***** top of data *****
00001 1                J E S 2  J O B  L O G  --  S Y S T E M  D A E F
00002 0
00003 10.45.30 JOB63346 ---- SUNDAY,    25 SEP 2016 ----
00004 10.45.30 JOB63346 IRR010I USERID ESI      IS ASSIGNED TO THIS JOB.
00005 10.45.31 JOB63346 ICH70001I ESI      LAST ACCESS AT 10:45:31 ON SUNDAY
00006 10.45.31 JOB63346 $HASP373 DUMMY   STARTED - INIT 22   - CLASS K
00007 10.45.31 JOB63346 IEF403I DUMMY - STARTED - TIME=10.45.31
00008 10.45.31 JOB63346 -                                     ----TIMINGS (
00009 10.45.31 JOB63346 -STEPNAME PROCSTEP   RC   EXCP   CONN       TCB
00010 10.45.31 JOB63346 -STEP01                00     4     0 0.000091 0.
00011 10.45.31 JOB63346 IEF404I DUMMY - ENDED - TIME=10.45.31
00012 10.45.31 JOB63346 -DUMMY   ENDED.  NAME-                                     TOTAL TC
00013 10.45.31 JOB63346 $HASP395 DUMMY   ENDED - RC=0000
00014 0----- JES2 JOB STATISTICS -----
00015 -   25 SEP 2016 JOB EXECUTION DATE
00016 -           25 CARDS READ
00017 -           63 SYSOUT PRINT RECORDS
00018 -           0 SYSOUT PUNCH RECORDS
00019 -           7 SYSOUT SPOOL KBYTES
00020 -           0.00 MINUTES EXECUTION TIME
***** ***** bottom of data *****

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

Notes about SYSOUT File Display

The following applies when viewing SYSOUT:

- **Name of the SYSOUT file**

In BS2000, UNIX and Windows, the SYSOUT file name appears in the upper left-hand corner of the screen.

- **Automatic Logon to the Operating System Server**

Before a SYSOUT file is displayed, Entire Operations checks whether the user is allowed to access the SYSOUT file. If necessary, an automatic logon is carried out with the user ID, which is entered for the user as a standard user ID for the operating system server (see the *Administration* documentation). If this automatic logon does not function, for BS2000 files, a further attempt will be made with the BS2000 user ID from the fully qualified file name. If this was successful, the current user ID is set accordingly for the operating system node.

If the user does not (implicitly or explicitly) log on to the operating system server with the credentials required to access SYSOUT, the SYSOUT file cannot be displayed.

■ **Line Length under UNIX and Windows**

Lines longer than 250 characters will be abbreviated in the SYSOUT display. They will be marked with >>> in the columns 247 to 250 of the display.

■ **File Truncation**

If the SYSOUT file or spool data set exceeds a given line limit, the display will be truncated: see **SYSOUT Line Limit** in *Default Setting (2)* in the *Administration* documentation.

■ **Number of Previous Executions**

Entire Operations can display SYSOUT of up to 180 previous executions of the job (e.g., repetitions, cyclic executions).

SYSOUT File Types on z/OS and z/VSE

On z/OS and z/VSE, SYSOUT files are identified by the following output types to distinguish their contents:

File Type	Purpose	Applies on
CC	Summary of job steps and condition codes.	
JL	JCL of selected job.	
SI	SYSIN data.	
SM	System messages.	
S0	SYSOUT data.	
LST	List queue.	z/VSE only.
PUN	Punch queue.	z/VSE only.
RDR	Reader queue.	z/VSE only.
XMT	Transmit queue.	z/VSE only.

The file types listed in the table above can be used in the FILE special command described in [SYSOUT Special Commands on z/OS and z/VSE](#).

SYSOUT Special Commands on z/OS and z/VSE

On z/OS and z/VSE, in addition to the standard Editor browse commands, the following special commands are available to display selected SYSOUT files:

Command	Description	Applies on
[FILE] <i>number</i>	Display the SYSOUT file by <i>number</i> , for example, FILE 4. If you omit the FILE keyword, the command scans the current file for a line number and displays the text from this line.	z/OS only.
[FILE] <i>type</i>	Display the SYSOUT file by <i>type</i> , for example, FILE S0. See also SYSOUT File Types on z/OS and z/VSE .	
[FILE] <i>type number</i>	Select the SYSOUT file by <i>type</i> and <i>number</i> , for example, FILE S0 4. See also SYSOUT File Types on z/OS and z/VSE .	z/OS only.
LINE 2	Split all lines too long for display into 2 lines. Return to normal display with an unqualified LINE command.	z/OS only.
NEXT	Display the next SYSOUT file.	
PREV	Display the previous SYSOUT file.	

Square brackets [] around the command keyword FILE denote that the keyword is optional.

55

Maintaining Active Job Conditions

▪ Listing Active Conditions	476
▪ Viewing an Active Condition and Changing Its Status	480
▪ Adding an Active Condition	480
▪ Deleting Single or Multiple Active Conditions	481
▪ Viewing the Usage of Conditions Concatenated with an Active Job	483
▪ Displaying Jobs Using an Active Condition	484
▪ Viewing Calling Jobs of Subnetworks	485

Logical conditions (see the *Concepts and Facilities* documentation) are defined on the master database when linking jobs within a job network. When a job network is activated and running, Entire Operations sets logical conditions automatically (active conditions) according to the occurrence of system events. Alternatively, you can set active conditions manually. You can display and maintain active conditions.

An active condition reflects the current value of the condition for a given job network activation. It can have the value true (the condition exists) or false (the condition does not exist). The run number assigned to the job network at activation is automatically passed to the active conditions defined for the jobs in the network. An active condition is uniquely identified by owner, network, run number and condition name.



Notes:

1. You can define new input conditions for an active job, or delete or modify existing conditions. Any changes to active conditions for active jobs are valid for the current job run only.
2. You can maintain active input conditions in the same way as described for master input conditions in *Defining and Managing Job Input Conditions* in the section *Job Maintenance*.

Listing Active Conditions

➤ To list all active conditions used by all active networks

- On the **Active Jobs**, **All Active Jobs** or **Input Conditions Maintenance** screen, press PF4 (ACond).

An **Active Conditions** screen similar to the example below appears:

```

16-09-25          ***** Entire Operations *****          16:06:18
Owner NATQA5          Active Conditions
-----
Cmd  Owner      Network      Condition          Date      Time      Run  Status
  _  NATQA5      N1587ASY     NET-BEGIN          16-09-19  16:10     297  free
  _  NATQA5      VKA-SYMBA    JOB1-OK            16-09-19  16:00     297  free
  _  NATQA5      VKA-SYMBA    JOB1-OK            16-09-20  21:06     298  free
  _  NATQA5      VKA-SYMBA    JOB2-OK            16-09-19  16:01     297  free
  _  NATQA5      VKA-SYMBA    JOB2-OK            16-09-20  21:07     298  free
  _  SYSDBA      SYSDBA       +ABC-COND          14-04-04  00:00     abs  free
  _  SYSDBA      SYSDBA       +COND-EXAMPLE      14-06-24  00:00     abs  free
  _  SYSDBA      SYSDBA       (C)+HEB-NUR-JCL-P  14-04-08  00:00     abs  free
  _  SYSDBA      SYSDBA       (C)+OGC-JCL-CHECK  15-08-03  00:00     abs  in use
  _  SYSDBA      SYSDBA       +XYZ-COND           14-04-04  00:00     abs  free
  _  SYSDBA      SYSDBA       +ZTEST              16-07-12  00:00     abs  free

***** Bottom of Data *****
D Delete  M Modify  W Where Used
Command => _____

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Add    End          Delet Up    Down          Menu

```

This screen displays a list of all active conditions belonging to the selected owner and networks. The fields and columns are explained in [Columns and Fields: Active Conditions](#).

➤ **To list input conditions used by an active job run**

- 1 On the **Active Jobs** or **All Active Jobs** screen, type I in the line command input field next to the required job, and press ENTER.
- 2 Press ENTER.

An **Input Condition Maintenance screen** with a list of input conditions for the selected job appears.

This screen corresponds to the **Input Condition Maintenance screen** of the **Job Maintenance** function on the master database described in [Defining and Managing Job Input Conditions](#) in the section [Job Maintenance](#).

This section covers the following topics:

- [Columns and Fields: Active Conditions](#)

Column	Field	Description
		1 In use. Can be used by jobs which do not require exclusive usage.
		2 Exclusive. Currently in use by a job; not usable by other jobs.
		3 Destructive. Currently in use by a job; not usable by other jobs. The condition will be deleted after job termination.
	Special Type	Usage of the condition. Possible input values for input fields:
		(blank) Condition is not used for checking JCL.
		C Condition is only used for checking JCL . If this option is selected, the name of the condition in the Condition column is prefixed with (C).

Commands: Active Conditions

The following line commands are available on the [Active Conditions](#) screen:

Line Command	Function
D	Delete the condition. See Deleting Single or Multiple Active Conditions .
M	View the condition and modify its status. See Viewing an Active Condition and Changing Its Status .
W	Display the job that uses the active condition. See Displaying Jobs Using an Active Condition .

The following special PF keys are available on the [Active Conditions](#) screen:

PF Key	Name	Function
PF6	Delete	Delete single or multiple active conditions. See Deleting Single or Multiple Active Conditions .
PF7	Up	Scroll up in the list of active conditions.
PF8	Down	Scroll down in the list of active conditions.

Viewing an Active Condition and Changing Its Status

You can view an active condition and modify its status to change the use made of it by future runs of the job for which it is defined.

➤ To view a condition active and change its status

- 1 On the **Active Conditions screen**, type M in the line command input field next to the required condition, and press ENTER.

A **Condition Modification** window opens.

The fields in the window have the same meaning as the corresponding columns on the **Active Conditions** screen. They are explained in *Columns and Fields: Active Conditions*.

- 2 If required, you can change the **status** of the condition, and press PF5 (Save) to save the modification.
- 3 Press PF3 (End) to close the **Condition Modification** window and return to the **Active Conditions** screen.

Adding an Active Condition

You can add an active condition to change the job flow by defining if for a job before submission.

➤ To add an active condition

- 1 Press PF2 (Add) on the **Active Conditions screen**.

A **Condition Addition** window similar to the example below opens:

```

05.02.16          ***** Entire Operations *****          16:22:43
Owner TESTBED          Active Conditions
-----
Cmd  Owner      Network  Condition          Date   Time   Run  Status
TESTBED--- *-----*-----
_ TESTBED  NET  +-----+ free
_ TESTBED  NET  | free
_ TESTBED  NET  | Condition Addition | free
_ TESTBED  NET  | Owner ==> TESTBED   | free
_ TESTBED  NET  | Network ==> _____ | free
_ TESTBED  NET  | Condition ==> _____ | free
_ TESTBED  NET  | Date / Time ==> _____ | free
_ TESTBED  NET  | Run Number ==> _____ | free
_ TESTBED  NET  | Special Type ==> _ | free
_ TESTBED  NET  | Status ==> _ | free
_ TESTBED  NET  | free
*****
D Delete  M Modif | Enter-PF1---PF2---PF3-----PF5----- | *****
Command => _____ +-----+
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help Add End Delet Up Down Menu

```

The input fields have the same meaning as the corresponding columns on the **Active Conditions** screen. They are explained in [Columns and Fields: Active Conditions](#).

- 2 Enter values for the input fields.
- 3 Press PF5 (Save) to add the active condition.
- 4 Press PF3 (End) to close the **Condition Addition** window and return to the **Active Conditions** screen.

Deleting Single or Multiple Active Conditions

You can delete single or multiple active conditions.



Note: Check carefully the side effects before you delete multiple active conditions.

➤ To delete a single active condition

- 1 On the **Active Conditions screen**, type **D** in the line command input field next to the required condition, and press **ENTER**.

A confirmation window opens.

2 Enter Y and press ENTER to delete the condition and return to the **Active Conditions** screen.

➤ **To delete multiple active conditions**

1 Press PF6 (Delete) on the **Active Conditions screen**.

An **Active Condition Deletion** window opens:

```

27.10.08          ***** Entire Operations *****          15:58:37
Owner SN          Active Conditions
-----
Cmd Owner      Network  Condition          Date    Time  Run  Status
SN*----- *----- *-----
_ SN          BS2-EX-2  JOB-1-OK          27.06.07 16:20  1108 free
_ SN          BS2-EX-2  JOB-1-OK          27.06.07 16:21  1109 free
_ SN          BS2-EX-2  JOB-1-OK          27.06.07 16:21  1110 free
_ SN  +-----+ 7 00:07  1111 free
_ SN  !          ! 7 16:22  1108 free
_ SN  !          Active Condition Deletion  ! 7 16:22  1109 free
_ SN  !          ! 7 16:22  1110 free
_ SN  !  Owner    ==> _____  ! 7 00:08  1111 free
_ SN  !  Run from ==> _____  ! 7 16:23  1110 free
_ SN  !    to     ==> _____  ! 7 00:08  1111 free
_ SN  !  Confirm? ==> Y          ! 7 08:12  1400 free
***** !          ! *****
D Delete !          !
Command !  Enter-PF1---PF3----- ! _____
Enter-PF1 !          Help End          ! F9---PF10--PF11--PF12---
Hel +-----+ Menu
    
```

This screen displays a list of all active conditions belonging to an owner.

You can use several selection criteria for the active conditions to be deleted: see [Fields: Active Condition Deletion](#).

- 2 Enter the required selection criteria for the active conditions to be deleted and press ENTER.

Fields: Active Condition Deletion

The following table explains the fields available on the **Active Condition Deletion** window:

Field	Description
Owner	Owner of the conditions. For possible selection criteria, see <i>Filtering Items in a Result List in Using Entire Operations</i> .
Network	Network of the conditions. For possible selection criteria, see <i>Filtering Items in a Result List in Using Entire Operations</i> .
Condition	Condition(s).
Run from	Start of run number interval to be deleted.
Run to	End of run number interval to be deleted.
Confirm	Y Prompt for each deletion. N Perform deletions without prompting.

Viewing the Usage of Conditions Concatenated with an Active Job

➤ To view condition usage of an active job

- In the **Prerequisites** window, enter the line command W (see [Line Commands: Prerequisites for an Active Job](#)) next to the required condition.

A **Condition** window similar to the example below opens:

```

17-04-14 +-----+ 1:36:41
!
Owner EX ! Owner   EXAMPLE      Condition E60-J013-0
! Network E60-FLOW
! 17-04-14
----- ! -----
C Condit !      Output Condition of
w E60-J0 !      Owner   Network      Run Job
!      EXAMPLE E60-FLOW      10 JOB-013
!      EXAMPLE E60-FLOW      10 JOB-014
!      EXAMPLE E60-FLOW      61 JOB-013
!      EXAMPLE E60-FLOW      61 JOB-014
!
!
!      Input Condition for
!      Owner   Network      Run Job
!      EXAMPLE E60-FLOW      10 JOB-014
***** ! *****
D Delete !
!
!
Command ! -----PF3-----PF7---PF8----- ! _
!      End           Up     Down
!
+-----+
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Add   End   ACond Save           Up     Down                               Menu

```

The left column of the window displays the jobs which can generate the condition. The right column of the window displays the jobs which use this condition as an input condition.

Displaying Jobs Using an Active Condition

➤ To list the job using an active condition

- On the **Active Conditions screen**, type W in the line command input field next the the required active condition, and press ENTER.

Or:

In the **Prerequisites window**, type W in the line command input field next to a condition for which a job is waiting and press ENTER.

An **Active Condition Usage** window similar to the example below opens:

```

18-02-01          ***** Entire Operations *****          11:06:00
Owner SAGTEST          Active Conditions
-----
Cmd Own +-----+
      SAG !                                     !
_   SAG !           Active Condition Usage         !
W   SYS !                                     !
_   SYS !   Owner      SYSDBA          used by Owner   SAGTEST   !
_   SYS !   Network    SYSDBA          Network         SAGNET   !
_   SYS !   Condition  +ABC-COND       Run            2         !
_   SYS !   Run        ABS              Job            JOB-01   !
_   SYS !   Status     3 destruct.     Begin         17-02-25 15:57:43 !
_   SYS !   spec.Type  End              End            17-02-25 15:58:08 !
      !                                     !
      !  -----PF1-----PF3----- !
      !      Help      End          !
+-----+
***** Bottom of Data *****
D Delete   M Modify   W Where Used
Command => _____

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Add    End          Delet Up   Down          Menu

```

This window displays the job, which is using the active condition, or the job, which last used the condition.

If several jobs use the condition simultaneously, only one job will be displayed. If the field **End** is blank, the condition is currently being used. Otherwise, it is not in use.

Viewing Calling Jobs of Subnetworks

This function can be used only for active jobs which are part of an active subnetwork.

➤ To view active jobs calling a subnetwork

- 1 On the **Active Jobs** or **All Active Jobs**, type Y next to a job that belongs to a subnetwork.

A **Calling Job Information** window similar to the example below opens:

```

18-02-01          ***** Entire Operations *****          17:03:25
Active Jobs      Owner SAGTEST      Network SAGNETSUB1 Version
Selection AW_____ Run from 54___ to 54___
-----
C  Job          Run Typ Loc   JobId  Node  Date  Time Message
*-----*
Y  JOB-A        54 JOB                02-01 17:03 ==> Y-Cmd executed <==
- JOB- +-----+ ME not f
- SUBN !                               ! RUN not
      !                               !
      ! Calling Job Information      !
      !                               !
      !                               !
      ! Owner      Network      Run Job      !
      ! Subnetwork SAGTEST    SAGNETSUB1  54 JOB-A    !
      ! called by  SAGTEST    SAGNET      346 J2-SUB  !
      !                               !
      ! -----PF1----PF3-----PF6----- !
      ! Help      End          Calling Network !
***** +-----+ *****
A Sc.P B Brw C Can D Dea E Edit G Gen.JCL H Hold I InCond J JCL K A.Res L Res.
M Mod. O EOJ P Prose R Resub. S SYSOUT T Stop U Rel. V RA W Wf Y SubC Z Subnet
Command => _____

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Add  End  ACond Save ResUs Up   Down ATask Left  Right Menu

```

(If the selected job has no calling job or does not belong to a subnetwork, the message No calling Job occurs instead.)

In the example above, the job JOB-A of the subnetwork SAGNETSUB1 is called by the J2-SUB job of the SAGNET network.

- 2 You can press PF6 (Calling Network) to open the **Active Jobs** screen of the calling network (here: SAGNET) and close the **Calling Job Information** window.
- 3 You can press PF3 to reopen the **Calling Job Information** window from this screen as shown in the following example:

```

18-02-02          ***** Entire Operations *****          09:33:19
Active Jobs          Owner SAGTEST      Network SAGNET      Version
Selection AW_____ Run from 347__ to 347__
-----
C  Job          Run Typ Loc   JobId  Node  Date  Time Message
*-----
_ DEMO-JOBK     347 JOB NAT           42 02-02 09:15 No JCL found for DEMO-
_ DEMO +-----+ or DEMO-
_ C JCLJ !                                           ! 1 requir
_ JCLJ !           Calling Job Information          ! ed
_ JCLJ !                                           ! RU/B60-M
_ JCL2 !           Owner          Network          Run Job          ! $ not al
_ JCL3 ! Subnetwork SAGTEST    SAGNETSUB1     55 JOB-A          ! alid Use
_ JOB- ! called by  SAGTEST    SAGNET         347 J2-SUB       ! failed,
_ JOB- !                                           ! failed,
_ JOB- ! ----PF1----PF3-----PF6----- ! failed,
_ JOB- ! Help End          Calling Network          ! found
***** +-----+ *****
A Sc.P B Brw C Can D Dea E Edit G Gen.JCL H Hold I InCond J JCL K A.Res L Res.
M Mod. O EOJ P Prose R Resub. S SYSOUT T Stop U Rel. V RA W Wf Y SubC Z Subnet
Command => _____

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help Add End ACond Save ResUs Up Down ATask Left Right Menu

```

Special PF Key

PF Key	Name	Function
PF6	Calling Network	Opens the Active Jobs screen of the calling network with a list of all its active jobs.

56

Maintaining Active JCL (Job Control Language)

- Defining JCL for an Active Job 490
- Browsing Active JCL 490
- Editing Active JCL 491
- Exchanging Active JCL 492
- Regenerating Active JCL 492

Related Topics:

See also the following in the section *Job Maintenance*:

- [Pregenerating Active JCL](#)
- [Defining and Managing JCL for a Job](#)
- [Editing Master JCL and Natural Sources](#)

Defining JCL for an Active Job

➤ To insert or modify the job control definition of an active job

- On the [Active Jobs](#) or [All Active Jobs](#) screen, type J in the line command input field next to the required job, and press ENTER.

A **Job: JCL Definition (active)** window opens displaying the current definitions.

How to modify the JCL definition is described in [Defining the Job Control \(JCL\)](#) in the section *Job Maintenance*.



Note: The modifications to the JCL definition of an active job are only valid for the current run.

Browsing Active JCL

You can view the actual JCL submitted to the operating system for execution. It is produced from the master JCL when the job or network is activated. The symbols are replaced with values from the active symbol table. If it is dynamic JCL, the generation is performed at this time. The active JCL is stored in the active Entire Operations database.

➤ To browse active JCL

- 1 On the [Active Jobs](#) or [All Active Jobs](#) screen, type B in the line command input field next to the required job, and press ENTER.
- 2 The active JCL is displayed in the editor. You cannot modify the source.

Editing Active JCL

For the current job run, you can edit the active JCL of an active job. Editing active JCL is useful, for example, for correcting JCL errors after job failure.

The following applies when editing active JCL:

- While editing active JCL, an implicit logon is triggered to establish the connection to the required node. The logon is not prompted if you have already logged on to the node within this session. For further information on logging on a node, see [Logon Function](#) in the section *Using Entire Operations GUI Client*.
- The source changes for the current job run only. The master JCL definition remains unchanged. If you want to edit the JCL for all runs, you must edit the master JCL.
- Any changes to JCL made from the **Active Jobs** or **All Active Jobs** screen affect the current job run only.
- Browsing or editing of active JCL is prohibited, while **JCL regeneration** for the same active job is in progress.
- If you modify active JCL before the job is submitted, Entire Operations submits the job automatically according to its defined prerequisites.
- If you modify the JCL after the job terminated and want to resubmit the job with the new JCL, issue the R line command for the job on the **Active Jobs** or **All Active Jobs** screen (see [Resubmitting an Active Job](#)).

➤ To edit active JCL

- 1 On the **Active Jobs** or **All Active Jobs** screen, type E in the line command input field next to the required job, and press ENTER.

The active JCL is displayed in the editor.

- 2 Modify the source as required by using editor commands and PF keys. See also [Editing Master JCL and Natural Sources](#) in the section *Job Maintenance*.

Exchanging Active JCL

When modifying an active job, you can specify another JCL member and library by overtyping the current values in the corresponding input fields. This allows you to run a different job in place of the old one using the same input conditions, End-of-Job handling, etc. The replacement is valid for the current network run only.

When you have specified a different JCL member and/or library, Entire Operations replaces the old JCL with the new and notifies you of the replacement with a message. The new member can use dynamically generated JCL as described in [Dynamic JCL Generation \(JCL Location MAC\)](#).



Note: After exchanging the JCL, the job must be restarted with the line command R (resubmit). For further information, see [Resubmitting an Active Job](#).

Regenerating Active JCL

The JCL for operating system jobs is generated (copied to the active database) when Entire Operations activates a job network. Entire Operations allows you to regenerate the JCL while the job is in the active database. This is useful when you wish to refresh the variables in dynamically generated JCL in MAC-type jobs or when you wish to (re)submit active jobs with their original JCL after editing their JCL for a specific job run.

Even if pregenerated JCL exists, this command regenerates the active JCL from the master JCL. Symbol values are taken in their current status from the active symbol table. If necessary, the active symbol table should be checked beforehand.

This section covers the following topics:

- [Symbol Prompting During JCL Regeneration](#)
- [Regenerating Active JCL](#)
- [Resubmitting the Active Job after JCL Regeneration](#)

Related Topic:

- [Pregenerating Active JCL](#) in the section *Job Maintenance*

Symbol Prompting During JCL Regeneration

If a symbol entry is defined for at least one symbol of the symbol table of the job, the symbol is prompted now. If symbol entry is cancelled, the JCL is not regenerated.

Symbol prompting during the JCL regeneration can be switched on or off by a global switch in the Entire Operations defaults. See the corresponding option setting described in *Default Setting (2)* in the *Administration* documentation.

The (re)generation of active JCL is always being performed online, i.e., not by the Entire Operations Monitor. This also applies to MAC-type jobs.

See also *Symbol Prompting during Network or Job Activation* in the section *Symbol Table and Symbol Maintenance*.

Regenerating Active JCL

➤ To regenerate active JCL

- 1 On the **Active Jobs** or **All Active Jobs** screen, type G in the line command input field next to the required job, and press ENTER.

A confirmation window opens.

- 2 Enter Y (yes) and press ENTER to confirm the JCL regeneration and close the window.
- 3 The message JCL regenerated for job appears in the message field of the selected job on the **Active Jobs** or **All Active Jobs** screen.

Resubmitting the Active Job after JCL Regeneration

After regeneration of the JCL with the line command G (generate JCL), the active job is *not* automatically restarted. You have to resubmit the job.

In addition, regenerating JCL for a NET type job (re)activates the subnetwork. However, it does not automatically start a subnetwork as a consequence. The subnetwork receives a new run number because of this action.

➤ To start a job (and a subnetwork) after regeneration of the JCL

- On the **Active Jobs** or **All Active Jobs** screen, enter the line command R (Resub.) next to the required job.

See also *Resubmitting an Active Job of NET type (Subnetwork)*.

IX

Schedule Maintenance

Schedules contain the planned execution dates of job networks. They can contain periodic and/or explicit schedule dates. You can define an unlimited number of schedules, and one schedule can be referenced in different job networks.

If a schedule table is based on a predefined calendar, execution dates can be defined relative to holidays (for example: the last working day of a month).

General Scheduling Considerations

Maintaining Schedules

Scheduling a Network

Scheduling a Job

Related Topic:

- *Calendar Maintenance*

57

General Scheduling Considerations

- Optional Schedule Definition 498
- Schedule Extraction Times 498
- Manual and Automatic Activations on the same Day 498
- Multiple Network Activations 498
- Influence of Deactivations on Schedules 498
- Imported Schedules 499
- Schedule Dependencies across the Turn of the Year 499
- Using Calendars 499

Optional Schedule Definition

You need not define a schedule for a network. If no schedule is defined and all schedule definitions are left empty, the network is never activated automatically by the Entire Operations Monitor.

Schedule Extraction Times

All network schedules are checked by the Monitor at least once a day, usually at midnight or after the first start of the Monitor on a new day. The networks with a schedule entry for the current day are activated. If no **Earliest Start Time** is defined, execution starts immediately after activation, depending on the earliest start time defined at the job level. New or modified networks with a schedule entry for the current day are activated also if the activation time is not reached already.

The extraction date is always used for the job time frame calculation, even if the network is activated some time later than planned, because the Monitor was down.

Manual and Automatic Activations on the same Day

A scheduled activation will not be rejected if a manual or API activation of the same network has already been made for the same day.

Multiple Network Activations

You can define multiple network activations in a network schedule.

For detailed information, see [Defining Multiple Network Activations](#).

Influence of Deactivations on Schedules

The day of the activation is removed during a deactivation from the schedule only, if the planned activation was originated by a schedule extraction by the Monitor. The day of the activation is not removed from the schedule if a manual activation is deactivated.

Imported Schedules

If a schedule is imported by the Import/Export utility, a check for the current day in the schedule will be performed. If the current day is a schedule day, it will be excluded explicitly from the schedule automatically. If you want make the schedule active for the current day, you must remove this explicit exclusion definition from the schedule.

Reason: The exclusion of the current day during import is done to prevent an unwanted automatic start of an imported job network by the Entire Operations Monitor.

Schedule Dependencies across the Turn of the Year

Schedule dependencies can also be defined across the turn of the year. This applies in particular to the week, which begins in the old year and ends in the new year.

This functions only if the schedules and calendars used are defined for both years.

Using Calendars

You need not use calendars when defining a schedule. If no calendar is specified, all days are treated as workdays.

You may use calendars, which belong to the owner of the network, and calendars of other owners.

If a calendar is specified, a schedule definition date is used only if it is a calendar workday. Networks are not activated on holidays.

For more information on calendars, see the section [Calendar Maintenance](#).



Notes:

1. The Monitor searches for the calendar under the network owner. If it does not find the calendar, the Monitor searches for it system-wide under SYSDBA.
2. Calendars are year-dependent. If a calendar cannot be found for the current year, the network is not scheduled and an error message is issued. Be sure to define calendars for the coming year before they are needed. From the month of November of the past year onwards, the Entire Operations Monitor issues warning messages to the log.
3. Calendars are linked to schedules, not to networks.

4. A calendar modification causes the automatic recalculation of all linked schedules, and a recalculation of the current schedules of all networks linked to these schedules. This automatic recalculation is performed by the Entire Operations Monitor in the background. Be aware that this may have influence on many networks, and may take some time. To check which networks were affected, check the Entire Operations log.

58

Maintaining Schedules

- Listing Schedules 502
- Displaying, Adding or Modifying a Schedule 504
- Deleting a Schedule Definition 511

Listing Schedules

> To list all schedule definitions of an owner

- From the **Main Menu**, select the **Schedule Maintenance** option.

A **Schedule Maintenance** screen similar to the example below appears:

```

28.03.15          ***** Entire Operations *****          17:51:47
Owner NOPALL          Schedule Maintenance
-----
Cmd  Owner      Schedule  Description
  _  NOPALL----  *-----
  _  NOPALL      AGBB551
  _  NOPALL      ALLDAY      all days
  _  NOPALL      I1029186A   last week of year: 1V
  _  NOPALL      I1029186B   first week of year: 6W
  _  NOPALL      I1029186C1  months / days
  _  NOPALL      I1029186C2  months / weekdays
  _  NOPALL      I1029186S1
  _  NOPALL      I1029186S2
  _  NOPALL      I1044152
  _  NOPALL      KOEZEIT
  _  NOPALL      MO-WE-FR    Monday, Wednesday, Friday
  _  NOPALL      N1573A
  _  NOPALL      N1792A      NFS Apps Daily (Mon-Fri)
***** m o r e *****
C Copy  D Delete  F List Def.  L List  M Modify  S Description  W Where used
Command => _____

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Add   End                Up    Down        Left  Right Menu
    
```

If one or more schedules are defined for the owner, they appear in the main information section of this screen.

This section covers the following topics:

- [Columns: Schedule Maintenance Screen](#)
- [Line Commands: Schedule Maintenance](#)

- [Special PF Keys: Schedule Maintenance](#)

Columns: Schedule Maintenance Screen

The following columns are available on the [Schedule Maintenance screen](#):

Column	Description
Cmd	One-character line command input field. Possible values are listed in the command section of the screen (see also Line Commands: Schedule Maintenance).
Owner	Owner of the schedule. The first line contains the prefix specification used to obtain the list of networks: enter an asterisk (*) and press ENTER to list networks belonging to the current owner (see screen header) and networks of all other owners for which you are authorized, or enter ABC* and press ENTER to list networks belonging to all owners whose names begin with ABC.
Schedule	User-defined schedule name. The first line contains the prefix specification used to obtain the list of networks: enter an asterisk (*) to list all networks belonging to the specified owners, or enter ABC* to list networks whose names begin with ABC.
Description	Short description of the schedule.

Line Commands: Schedule Maintenance

You can perform several functions on any schedule listed on the [Schedule Maintenance screen](#) using the following line commands:

Line Command	Description
C	Copy the selected schedule to a new one.
D	Delete the selected schedule .
F	List (browse) the selected schedule (as definition).
L	List (browse) the selected schedule (in calendar format).
M	Modify selected schedule definition.
S	Add or modify the schedule's short description.
W	Where used: display using networks.

Special PF Keys: Schedule Maintenance

The [Schedule Maintenance screen](#) provides the following special PF key:

PF Key	Name	Function
PF2	Add	Add a schedule definition.

Displaying, Adding or Modifying a Schedule

This section covers the following topics:

- [Viewing a Schedule Definition](#)
- [Viewing a Schedule Definition as a Calendar](#)
- [Adding a Schedule Definition](#)
- [Modifying a Schedule Definition](#)
- [Fields: Schedule Definition](#)
- [Special PF Keys: Schedule Definition](#)
- [Copying a Schedule Definition](#)

Viewing a Schedule Definition

➤ To view a schedule definition

- 1 On the [Schedule Maintenance screen](#), type F in the line command field next to the schedule you want to view. Press ENTER.

A [Schedule Definition screen](#) appears with the current values for the network.

The fields and special PF keys provided on the screen are explained in [Fields: Schedule Definition](#) and [Special PF Keys: Schedule Definition](#).

- 2 Press PF3 (Exit) to return to the **Schedule Maintenance** screen.

Viewing a Schedule Definition as a Calendar

➤ To view the calendar of a schedule definition

- 1 On the [Schedule Maintenance screen](#), type L in the line command input field next to the schedule you want to view. Press ENTER.

A calendar view similar to the example below is shown on the screen:

21.12.09	***** Entire Operations *****												14:59:51	
Owner SN	Schedule A-1				Year 2009				Calendar DEMOCAL					
	July				August				September					
Monday	31			28					
Tuesday	28		29					
Wednesday	29		30					
Thursday	30							
Friday	31		28								
Saturday						
Sunday						
	October				November				December					
Monday	30			28					
Tuesday	15	29				
Wednesday	28		30					
Thursday	29							
Friday	30							
Saturday						
Sunday						
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11--PF12---														
Help			End			Up			Down			Calen		

The current six months of a schedule object are displayed.

- 2 Press PF7 (Up) to display the first half of the year, or PF8 (Down) to display the second half of the year.

Scroll the years by using PF7 (Up) to display the schedule for the previous year, and PF8 (Down) to display the schedule for the following year.

- 3 Press PF3 (End) to abort the function and return to the list of schedules.

Adding a Schedule Definition

Related Topic:

- *Copying a Schedule Definition*

- **To add a schedule definition**

- 1 Press PF2 (Add) on the **Schedule Maintenance** screen.

The **Add New Schedule** window opens.

- 2 Enter a schedule name and press ENTER.

A **Schedule Definition** screen similar to the example below appears:

```

23.10.08          ***** Entire Operations *****          11:17:02
Owner SN          Schedule Definition          Schedule A-1
-----
Calendar Owner ==> SN_____ Calendar ==> _____
Monthly Schedule:
Month List ==> 1_  2_  3_  4_  5_  6_  7_  8_  9_ 10 11 12

Month Days ==> 28__ 29__ 30__ 31__ _____ _____ _____ _____
                _____ _____ _____ _____ _____ _____ _____
                _____ _____ _____ _____ _____ _____ _____

Weekly Schedule:
Month list ==> 1_  2_  3_  4_  5_  6_  7_  8_  9_ 10 11 12

Week Days ==> 1_  3_  4A  __  __  __  __  ( 1 Mon, 2 Tue, 3 Wed, ... )

Explicit Dates:
10.09.08_ 11.12.08_ 07.08.08_ _____ _____ _____ _____
_____ _____ _____ _____ _____ _____ _____
_____ _____ _____ _____ _____ _____ _____

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      End      Save      Cal  Reset      Menu
    
```

- 3 Define the schedule by entering values for the fields (see [Fields: Schedule Definition](#)) and using **special PF keys**.
- 4 Press PF5 (Save) to save the schedule definition.
- 5 Press PF3 (Exit) to return to the [Schedule Maintenance screen](#).

The new schedule is now listed on the [Schedule Maintenance screen](#).

Modifying a Schedule Definition

➤ **To modify an existing schedule definition**

- 1 Type M in the line command field of the selected schedule on the [Schedule Maintenance screen](#).
- 2 Press ENTER.
- 3 A [Schedule Definition screen](#) opens with the current values for the network.

The fields and special PF keys available are described in [Fields: Schedule Definition](#) and [Special PF Keys: Schedule Definition](#).

- 4 Modify any value.

- 5 Press PF5 (Save) to save the schedule definition.
- 6 Press PF3 (Exit) to return to the [Schedule Maintenance screen](#).

Fields: Schedule Definition

The fields on the [Schedule Definition screen](#) are explained in the following table.

Field	Description
Owner	Owner of the schedule
Schedule	Name of the schedule
Calendar Owner	Owner of the assigned calendar Default: the schedule owner
Calendar	Assigned calendar You can select a calendar by using a wildcard.
Monthly Schedule / Month List	List of months for the schedule They are always correlated with the Month Days or Week Days fields. Enter AL to define all months. Example: 01 02 03 04
Month Days (Monthly Schedule)	The days entered here are correlated with the list of months. Each combination of month and day is a schedule date. A schedule day may be followed by a letter with a special meaning: A next workday, if holiday B previous workday, if holiday W is workday of month, e.g. 10W V is workday, counted backwards. Weekday dependency within a month: These entries start with the letter K: see Weekday Dependency within a Month . Special Days (can also be followed by the above letters) LD means always the last day of the month, e.g. LDB AL means all days of the month. Month Days will be sorted in order AL, Kn<flag>, Kn+m<flag>, Kn-m<flag>, LD<flag>.
Weekly Schedule / Month List	A list of months for the schedule. They are always correlated with the Month Days or Week Days fields.

Field	Description
	<p>Enter AL to define all months.</p> <p>Example: 01 02 03 04</p>
<p>Week Days (Weekly Schedule)</p>	<p>The days entered here are correlated with the list of months.</p> <p>Each combination of month and weekday is a schedule date.</p> <p>1 is Monday, 2 is Tuesday, and so on.</p> <p>A weekday may be followed by a letter with a special meaning:</p> <p>A next workday, if holiday B previous workday, if holiday W is workday of week, e.g. 1W V is workday, counted backwards X or blank exactly this weekday</p> <p>Examples:</p> <p>1 Monday</p> <p>1A Monday, but next workday, if holiday</p> <p>2V 2nd last workday of week</p>
<p>Explicit Dates</p>	<p>A list of explicit dates in the date format specified within Entire Operations defaults.</p> <p>If the date is followed by a minus sign, the date will be excluded from the schedule.</p> <p>Example: 21.12.16-</p> <p>The network is not activated on 21.12.2016.</p> <p>Note:</p> <ol style="list-style-type: none"> 1. Explicit dates of a schedule list have a higher priority than all periodic definitions. 2. There is another network-specific list of explicit dates, which overrides all definitions made on the schedule level. <p>See also Defining Dates for Explicit Network Execution.</p>

Weekday Dependency within a Month

Monthly day definitions may also start with the letter K. The meaning is:

Field Content	Explanation	Example	Example Explanation
K<n><f>	weekday <n> within a month	K2 K2A	All Tuesdays within the given month list All Tuesdays within the given month list; use the next day if this day is a holiday.
K<n>+<m><f>	weekday <n> at position <m> within a month	K2+3	All third Tuesdays within the given month list
K<n>-<m><f>	weekday <n> at position <m> within a month; position counted backwards from the end	K2-1	All last Tuesdays within the given month list

Parameter Description

<n>	Weekday number Valid values are 1 thru 7 (1 = Monday, etc.)
<m>	The order number of the weekday. If preceded by a + (plus) sign, it is counted from the beginning of the month. If preceded by a - (minus) sign, it is counted backwards from the end of the month. <m> absent means all weekdays of month.
<f>	Flag (optional): A next workday, if holiday B previous workday, if holiday

Special PF Keys: Schedule Definition

You can perform the following functions using these PF keys on the [Schedule Definition screen](#):

PF Key	Name	Function
PF8	Cal	Displays the used calendar (only if a calendar is defined for this schedule).
PF9	Reset	Reset all fields on the screen.

Copying a Schedule Definition

When defining a new schedule, you can use an existing schedule and all its definitions as a model for the new schedule.

➤ **To copy a schedule definition**

- 1 On the **Schedule Maintenance screen**, type C in the line command field next to the schedule you want to copy. Press ENTER.

A **Schedule Definition Copy** window similar to the example below opens:

```

16-07-07          ***** Entire Operations *****          18:57:34
Owner SAGTEST          Schedule Maintenance
-----
Cmd  Owner      Schedule  Description
  _  SAGTEST--- *-----
  _  SAGTEST    HEUTE
  _  SAGTEST    JUNE      Schedule for June
  _  SAGTEST    SCHED-JUNE  Schedule for June
c   SAG +-----+
  _  !
  _  !          Schedule Definition Copy          !
  _  !
  _  !      From          To          !
  _  !      Owner    ==> SAGTEST__  Owner    ==> SAGTEST  !
  _  !      Schedule ==> SCHED-NEW_  Schedule ==> _____ !
  _  !
  _  ! PF3 End          !
  _  !
  _  !          !          !
  _  !          !          !
*****          !          ! *****
C Copy  +-----+ sed
Command => _____

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Add   End          Up    Down          Left  Right Menu
    
```

The name of the owner and schedule from which you want to copy the new schedule appear in the **From** fields.

- 2 Enter the name of the schedule to copy to in the **To** field. (If necessary, you can also enter a different owner and schedule in the **From** fields).

To open selection windows for schedules, use an asterisk (*) as a wildcard.



Note: You cannot copy a schedule to an existing schedule.

- 3 Press ENTER.

The window closes and the new schedule appears in the list on the *Schedule Maintenance* screen.

Deleting a Schedule Definition

➤ To delete a schedule definition

- 1 Type `D` in the line command field of the selected schedule on the *Schedule Maintenance* screen. Press `ENTER`.

A window opens in which you can confirm the deletion by entering the schedule name.

- 2 Type in the schedule name and press `ENTER` to delete the schedule.



Note: You cannot delete a schedule, which is defined (used) in at least one network. To find out which networks are using the schedule, use the `W` (Where used) line command on the *Schedule Maintenance* screen.

59 Scheduling a Network

- Defining a Network Schedule 514
- Defining and Deleting Time Frames for a Network 517
- Defining Dates for Explicit Network Execution 521
- Defining Multiple Network Activations 523
- Viewing a Network Schedule Definition as a Calendar 525
- Producing a Network Start Summary 526
- Schedule Modification and Planned Starts 528
- Displaying Next Network Starts - Single Network 530
- Displaying Next Network Starts - System-wide 535
- Displaying the Network Execution History 540

Defining a Network Schedule

> To define a schedule for a network

- 1 On the **Network Maintenance screen**, type S in the **Cmd** column next to the required network. Press ENTER.

A **Network Schedule Definition** screen similar to the example below appears:

```

17-02-07          ***** Entire Operations *****          11:46:58
                    Network Schedule Definition
                    Owner SAGTEST      Network B60-FLOW
-----
Schedule Owner ==> SAGTEST___
Schedule       ==> _____ effective from ==> _____
-----
Schedule      Earliest   Latest   Days      Days
Cmd Dependency Start      Start   later  Deadline later
_ +WY+123     09:25:00 09:25:00 1__  17:00:00 1__
_              08:00:00 09:00:00 ___  _____ 2__
_              _____  _____  ___  _____  ___
_              _____  _____  ___  _____  ___
_              _____  _____  ___  _____  ___
-----
Number of Activations ==> 10___ every ==> 30___ minutes
or activate at ==> _____
-----
Day Deadline   ==> _____
-----
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      End   Exp1D Save          Sch.L Sch.M Delet          Menu

```

- 2 Enter the required values or replace existing entries.

All **fields**, **line commands** and **special PF keys** provided on the screen are explained in the following section.

This section covers the following topics:

- [Fields: Network Schedule Definition](#)
- [Line Commands: Network Schedule Definition](#)

- [Special PF Keys: Network Schedule Definition](#)

Fields: Network Schedule Definition

The input fields of the [Network Schedule Definition screen](#) are described in the following table:

Field	Description
Schedule Owner	Owner of the schedule to be used. Wildcard selection is possible. Default: Owner of the network
Schedule	Name of the schedule to be used. Schedule comprises a link to a calendar. Definition of dates on which the network is to be executed.
effective from	Schedule effective from (date and time) If a schedule should not become effective immediately, enter the date (and time) of the validity begin here. If no time is specified, the schedule becomes effective at 00:00.
Cmd	Input field for line commands: see Line Commands: Network Schedule Definition .
Schedule Dependency	Schedule dependency of a time frame definition. See Adding and Modifying a Time Frame Definition .
Earliest Start	Earliest start time for the network. No job in the network can start before this time. The Earliest Start time is used to compute network schedule times. See also Date and Time Formats (section <i>Using Entire Operations</i>) and Behavior for Schedule Times .
Latest Start	Latest start time for the network. The first job in the network must start by this time, if possible. If not, a warning message is sent and the job is not submitted. See also Date and Time Formats (section <i>Using Entire Operations</i>) and Behavior for Schedule Times .
Deadline	Time by which the last job in the network must be finished. The deadline is used to compute the earliest and latest start times for the jobs in the network. See also Date and Time Formats (section <i>Using Entire Operations</i>) and Behavior for Schedule Times .

Field	Description
Days later	<p>This field is to be used if Latest Start and/or Deadline are later than 24 hours after the Earliest Start.</p> <p>See also Behavior for Schedule Times.</p>
Number of Activations	<p>This field is used to activate a network more than once a day. In this case, enter the number of activations per day.</p> <p>If the network is to be activated once per schedule day only, leave this field blank.</p> <p>Valid input values: 2 to 999</p> <hr/> <p>every ... minutes:</p> <p>If a network is to be activated more than once per day, you can enter the time interval between two subsequent network activations.</p> <p>This field can be used without Number of Activations, too. In this case, the number of activations will be limited by the defined latest network start time.</p> <p>Unit: minutes</p> <p>See also Defining Multiple Network Activations for further information.</p>
or activate at	<p>In these fields, you can enter up to 10 activation times, as an alternative to Number of Activations.</p> <p>Format: HH:MM or HHMM or HH</p> <p>Examples: 20:11, 1314, 18</p> <p>Note: If activation times are defined for a network, the job-specific time frames will be adapted to fit into the network time frame.</p> <p>See also Defining Multiple Network Activations for further information.</p>
Day Deadline	<p>If a Day Deadline is defined, the Deadline of multiple activations (every ... minutes or Activate at) is limited to this time on the day of the activation.</p> <p>Format: HH:MM</p>

Line Commands: Network Schedule Definition

The line commands available for the time frame columns on the [Network Schedule Definition screen](#) are described in the following table:

Line Command	Description
D	Delete the time frame definition. See also Deleting Time Frame Definitions for a Network .
I	Insert an empty line before the current line.
M	Add or modify the schedule dependency of the time frame definition. See also Adding and Modifying a Time Frame Definition .

Special PF Keys: Network Schedule Definition

You can perform the following function from the **Network Schedule Definition** screen using these PF keys:

PF Key	Name	Function
PF4	ExplD	Explicit Date. Define explicit execution dates at the network level. See also Defining Explicit Dates for a Single Network only and Date and Time Formats .
PF7	Sch.L	Schedule list. Display the linked schedule in a half-year calendar format.
PF8	Sch.M	Schedule modification. Invoke the schedule definition of the defined schedule.
PF9	Delete	Delete (reset) all fields of the screen.

Defining and Deleting Time Frames for a Network

- [Rules for Time Frame Definitions](#)
- [Adding and Modifying a Time Frame Definition](#)
- [Time Frame Selection Rules for a Network](#)
- [Behavior for Schedule Times](#)

- [Deleting Time Frame Definitions for a Network](#)

Rules for Time Frame Definitions

Hierarchical Order

Time frames defined on the job level override time frame definitions made on the network level for this job. If no time frames are defined here, the job network defaults apply.

Active Jobs

If this function is invoked for an active job, the time frames contain the complete calculated date and time. Both can be modified.

Adding and Modifying a Time Frame Definition

➤ To add or modify a time frame definition for a network

- 1 If you want to create or modify a time frame definition, enter or replace the required values in the time frame columns on the [Network Schedule Definition screen](#).

For valid input values, see [Fields: Network Schedule Definition](#).

For valid time formats, see [Date and Time Formats](#) in the section *Using Entire Operations*.

See also [Time Frame Selection Rules for a Network](#) and [Behavior for Schedule Times](#).

- 2 If you want to add or modify the schedule dependency for a time frame, type M in the **Cmd** column next to the line that contains the required time frame.

A **Schedule Dependency Definition for Time Frame** window similar to the example below opens:

```

17-03-03          ***** Entire Operations *****          17:44:58
                    Network Schedule Definition
                    Owner EXAMPLE          Network B60-FLOW
-----+-----+-----+-----+-----+-----+-----+-----+-----+
Schedul !
Schedul !          Schedule Dependency Definition for Time Frame          !
!
! Owner   EXAMPLE          Job          !
Sche ! Network B60-FLOW          Condition          !
Cmd Depe !
m ! The time frame will be used on this(+)/all other(-) dates,          !
_ ! if the execution date has the following position in the          !
_ ! job network schedule or calendar:          !
_ !
_ ! Usage ==> _ + only          - except          !
!          A after Holiday too B before Holiday too          !
Number !          Type ==> _ H is Holiday X is Workday          !
or acti !          C Calendar Day W Workday S Schedule Day          !
!          in Period ==> _ W Week          M Month Y Year          !
Day Dea ! at Position ==> _____          !
! -----PF1-----PF3-----PF5-----PF9-----          !
! Help End Save Delete          !
-----+-----+-----+-----+-----+-----+-----+-----+
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help End ExplD Save Sch.L Sch.M Delet Menu

```

The input fields contained in the window are described in [Fields: Schedule Dependency Definition](#).

Time Frame Selection Rules for a Network

Time frames are selected according to the following rules:

- A blank line with no time frame dependency entry is applied as the network start date if no other time frame dependency entry matches the network start date.
- From top to bottom, each line with a defined schedule dependency condition is applied to the network. If a condition is satisfied, the respective time frame values are used for the active network.

Example: The definition +CW+3 means this definition applies if the network starts on Wednesday. -CW+3 means this definition applies if the network starts on all other weekdays except Wednesday.

- If none of the defined lines is selected, the network uses the default time settings. See also *Default Setting (2)* in the *Administration* documentation.

Behavior for Schedule Times

1. If both **Latest Start** or **Deadline** and **Days later** are empty, the Entire Operation defaults specified for **Default latest Start after Earliest Start** and **Default Deadline after Earliest Start** on the **Default Setting (2)** screen applies (see the *Administration* documentation).
2. If **Latest Start** or **Deadline** is given and **Days later** is empty, **Days later** is automatically filled as described above.
3. If applying any of the above rules and **Days later** of **Latest Start** or **Deadline** becomes greater than 999, it remains empty and an error message is issued.
4. If **Latest Start** or **Deadline** is before **Earliest Start** and their **Days later** fields are empty, the **Days later** value of **Earliest Start** (or the following day, if necessary) is used for the **Days later** fields of **Latest Start** and **Deadline**.
5. If **Deadline** is before **Earliest Start** and **Days later** is empty, **days later** is set to 1 so this time follows the earliest start time.

Deleting Time Frame Definitions for a Network

➤ To delete a single time frame definition

- 1 On the **Network Schedule Definition** screen, type D in the **Cmd** column next to the time frame definition you want to delete.

A confirmation window opens.

- 2 Enter Y (Yes) to confirm the deletion. N (No) cancels the action. Press ENTER.

The selected time frame definition is removed from the **Network Schedule Definition** screen.

➤ To delete all time frame definitions

- 1 On the **Network Schedule Definition** screen, press PF9 (Delete).

A confirmation window opens.

- 2 Enter Y (Yes) to confirm the deletion. N (No) cancels the action.

Press ENTER.

All time frame definitions are removed from the **Network Schedule Definition** screen.

➤ To delete a time frame dependency from a time frame definition

- 1 In the **Schedule Dependency Definition for Time Frame** window, press PF9 (Delete).

A confirmation window opens.

- 2 Enter Y (Yes) to confirm the deletion. N (No) cancels the action.

Press ENTER.

The time frame dependency is removed from the time frame definition on the **Network Schedule Definition** screen.

Defining Dates for Explicit Network Execution

You can define explicit dates on which a network always or never executes, regardless of any other schedule definitions. Explicit dates supersede all other schedule definitions.

You can define explicit dates for all networks linked to a schedule or for a single network only.

This section covers the following topics:

- [Defining Explicit Dates for a Schedule](#)
- [Defining Explicit Dates for a Single Network only](#)

Defining Explicit Dates for a Schedule

➤ To add or modify explicit dates for a schedule

- On the **Schedule Definition** screen, enter the required dates in the input lines under **Explicit Dates** as shown in the example of a [Schedule Definition screen](#).

For valid input values, see the field [Explicit Dates](#) in *Fields: Schedule Definition*.

Defining Explicit Dates for a Single Network only

You can define explicit schedule dates on the network level. Dates defined here are network-specific in contrast to schedule definitions which affect all linked networks.



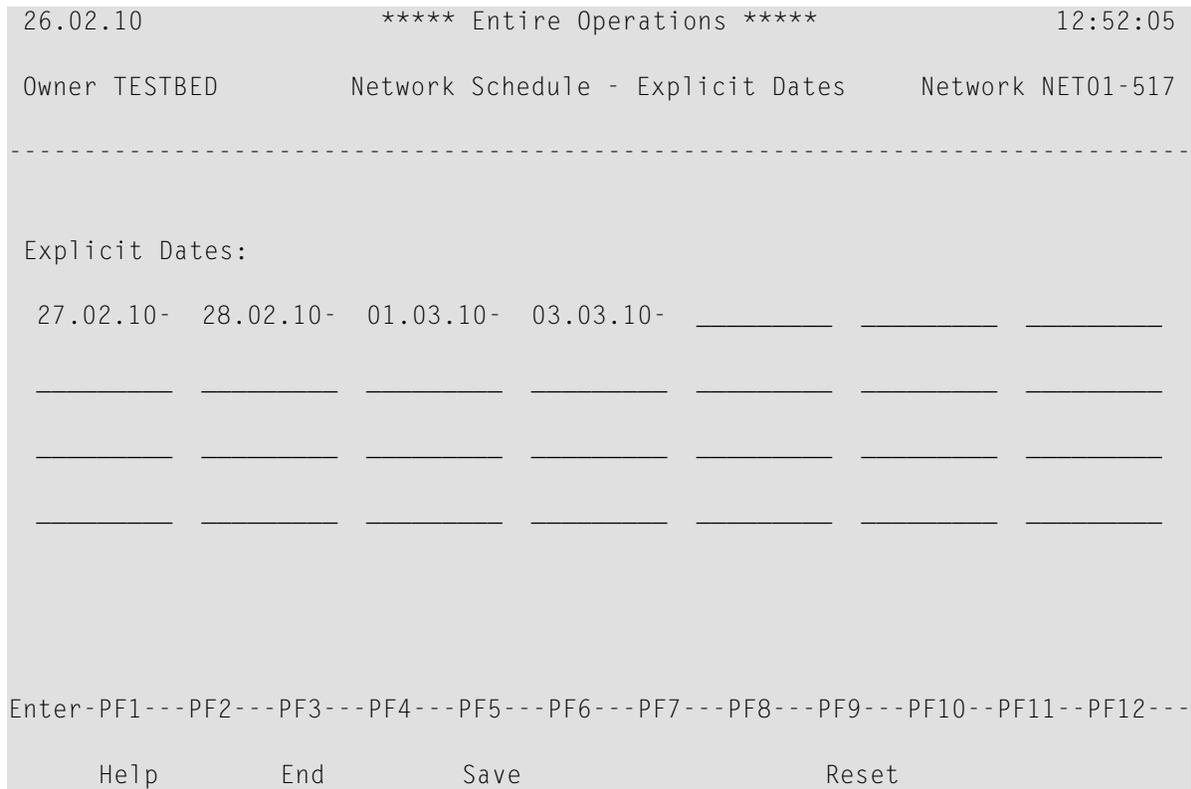
Notes:

1. Any modification made with this function causes the recalculation of the current schedule of the network.
2. Explicit dates override all definitions made in schedule(s) linked to the network.
3. Explicit exclusion dates on the network level will be set implicitly by the network activation canceling function.
4. Deletions of planned activations can cause the setting of “dates to be excluded” in this table (see also [Fields: Network Schedule – Explicit Dates](#)).

➤ **To define explicit dates for a single network only**

- 1 Press PF4 (ExpID) on the **Network Schedule Definition** screen.

A **Network Schedule - Explicit Dates** screen similar to the example below appears:



Enter or replace the dates you want to explicitly use for the selected network. The **input fields** and **special PF key** are described in the following section.

- 2 Press PF3 (End) to return to the **Network Schedule Definition** screen.

Special PF Keys: Network Schedule – Explicit Dates

You can perform the following function from the **Network Schedule - Explicit Dates screen** using this PF key:

PF Key	Name	Function
PF9	Reset	Reset (delete) all definitions.

Fields: Network Schedule – Explicit Dates

The input fields of the **Network Schedule - Explicit Dates screen** are described in the following table:

Field	Description
Explicit Dates	<p>A list of explicit dates in the date format specified within Entire Operations defaults.</p> <p>If the date is followed by a minus sign (-), the date will be excluded from the schedule.</p> <p>Example: 21.12.16-</p> <p>The network is not activated on 21.12.2016.</p>

Defining Multiple Network Activations

You can repeat network activations multiple times.



Note: If activation times are defined for a network, the job-specific time frames will be adapted to fit into the network time frame.

› To define multiple network activations

- On the **Network Schedule Definition screen**, enter (or replace) either or both of the following:
 - In the **Number of Activations field**, enter the required number of activations.
 - In the **every ... minutes field**, enter the time interval to be used for activation.

Or:

On the **Network Schedule Definition screen**, enter the following:

 - In the input fields next to **or activate at**, enter single or multiple times to be used for network activation.
- If required, enter a time limit for activation in the **Day Deadline**.
- Press PF5 (Save) to save your entries.

The rules that apply when defining multiple activations are described in the following section:

- **Activation End**

- Calculation of Time Frames for Multiple Activations

Activation End

Multiple activations are limited

1. until end of schedule day;
2. until latest start;
3. by the defined number of activations.

which ever comes first.

Calculation of Time Frames for Multiple Activations

The following applies if both **Number of Activations** (na) and **every ... minutes** (em) have a non-zero value:

- The latest start time of a time frame (tf_latest_start) is interpreted as the latest start time of the first multiple activation.
- For activation $1 \leq n \leq na$, the latest start time of the specific activation will be set to $tf_latest_start + (n - 1) * em$.
- The multiple activation end time (mae) will be calculated as $mae = schedule_date + tf_latest_start + (na * em)$

If the multiple activation end time (mae) is after the defined deadline of a time frame, it will be set to the deadline of the time frame.

If the multiple activation end time (mae) is after the end of the schedule day, it will be set to the end of the schedule day.



Note: A network is no longer treated as “already activated on the current day” if there was a multiple activation on the day before, which resulted in some runs that were executed after midnight. In this case, automatic activations (due to schedule) will not be blocked by the “activation on current day” check.

Viewing a Network Schedule Definition as a Calendar

> To display the calendar view of a network schedule

- On the **Network Maintenance screen**, type **W** in the line command field of the selected network. Press **ENTER**.

A **Network Schedule** screen similar to the example below appears:

```

16-07-02          ***** Entire Operations *****          16:20:20
Owner SAGTEST      Network Schedule B60-FLOW   Year 2016

                July                August                September
Monday           . . . . .                . . . . .                . . . . .
Tuesday          . . . . .                . . . . .                . . . . .
Wednesday        . . . . .                . . . . .                . . . . .
Thursday         . . . 28                . . . . .                . . . . .
Friday           . . . . .                . . . . .                . . . . .
Saturday         . . . . .                . . . . .                . . . . .
Sunday           . . . . .                . . . . .                . . . . .

                October               November              December
Monday           . . . . .                . . . . .                . . . . .
Tuesday          . . . . .                . . . . .                . . . . .
Wednesday        . . . . .                . . . . .                . . . . .
Thursday         . . . . .                . . . . .                . . . . .
Friday           . . . . .                . . . . .                . . . . .
Saturday         . . . . .                . . . . .                . . . . .
Sunday           . . . . .                . . . . .                . . . . .

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      End   Overv          Up    Down  Calen Histo NxtSt

```

The current half of the year is displayed in the calendar format. The dates on which the network is to be activated appear on the schedule.

The network based on this schedule is activated every weekday from January to June except on Saturday and Sunday.

You can enter any valid year in the **Year** field in the header section of the screen to display the schedule for the specified year.

If a calendar is linked to the schedule but not defined for the displayed year, you are notified by an error message.



Note: There are two different formats for all schedule and calendar displays. The format can be modified in the **Default Setting (1)** option of the **Entire Operations Defaults** described in the *Administration* documentation.

Special PF Keys: Network Schedule

You can perform the following functions from the **Network Schedule** screen using these PF keys:

PF Key	Name	Function
PF4	Overv	<p>Overview</p> <p>Produces a network start summary for a given date: see Producing a Network Start Summary.</p> <p>(Only if your Natural online system runs under z/OS)</p> <p>Before you press this key, place the cursor on a valid date.</p>
PF7	Up	Displays the previous half of the year or previous years.
PF8	Down	Displays the next half of the year or following years.
PF9	Calen	<p>Calendar</p> <p>Displays the calendar linked to this schedule (if defined).</p> <p>The calendar can be modified as described in the section Calendar Maintenance.</p>
PF10	Histo	<p>History</p> <p>Displays the previous execution dates of the network.</p> <p>See also Displaying the Network Execution History.</p>
PF11	NxtSt	<p>Next Starts</p> <p>Displays the next network and job activations for the network and whether they are scheduled or manual activations.</p> <p>See also Producing a Network Start Summary.</p>

Producing a Network Start Summary

This section describes how to produce a network start summary for a single network. The report looks similar to the [example of a Network Start Summary](#) shown in the [Reporting](#) section.

If you want to produce a **Network Start Summary** for all network activations on a given day, see the corresponding report described in the section [Reporting](#).

Canceling Planned Starts

➤ To cancel one or more of the planned starts listed

- Press PF5 (Cancel).

A confirmation window opens for each start to be cancelled:

```

29.10.08          ***** Entire Operations *****          13:08:28
Owner EXAMPLE          Schedule Definition
+-----+
- !
!      Owner EXAMPLE      Network E50-USRT      !
!
! The following planned starts would be          !eekly      E Explicit Dates
! cancelled due to your schedule modification:  !
!
!           +-----+
!           !           !
!           ! Please confirm           !
!           ! Start Cancel of E50-USRT (142)           !
!           ! by entering E50-USRT           !
!           !           ==> _____           !
!           ! PF3 End           !
!           +-----+
!
!
!
!
! -----
E ! PF5 Cancel      PF6 Do not cancel          !F9---PF10--PF11--PF12---
+-----+a1                                     Menu

```

Keeping Planned Starts

➤ To keep all listed starts active, regardless of schedule modifications you have made

- Press PF6 (Do not cancel).

Displaying Next Network Starts - Single Network

This section covers the following topics:

- [Displaying the Next Network and Job Starts](#)
- [Canceling a planned Network Start](#)
- [Symbol Prompting](#)
- [Modifying an Active Symbol Table](#)

Displaying the Next Network and Job Starts

➤ To display the next network and job starts for a network

- Press PF11 (NxtSt) on the [Network Schedule screen](#).

A **Next Start Times** window similar to the example below opens:

```

+-----+
!
!Owner TESTBED      Network NET01-517
!Next Start Times
!
!  Cmd  Date      Act.  Start  Run Type
!  _    01.03.10  07:29  07:29  4789 Schedule, active
!  _    01.03.10  09:29  09:29  4790 Schedule, active
!  _    01.03.10  11:29  11:29  4791 Schedule, active
!      02.03.10  00:00  07:29      Schedule Table
!      03.03.10  00:00  07:29      Schedule Table
!      04.03.10  00:00  07:29      Schedule Table
!      05.03.10  00:00  07:29      Schedule Table
!      08.03.10  00:00  07:29      Schedule Table
!      09.03.10  00:00  07:29      Schedule Table
!      10.03.10  00:00  07:29      Schedule Table
!  ----- Bottom of Data -----
!  D Delete  M Start Time  S Active Symbols
!  PF1 Help  PF2 Add  PF3 End  PF7 Up  PF8 Down
+-----+

```

This window displays a list of planned activations for one network only.

The **Type** column displays the activation type (scheduled, manual).

You can delete manual activations with the D (delete) line command. Scheduled activations can only be changed by modifying the schedule definition.

 **Note:** To display a system-wide list of planned activations, press PF11 (NxtSt) from the **Network Maintenance** or **Active Job Networks** screen. For further information, see *Displaying Next Network Starts - System-Wide*.

Columns: Next Start Times

The following table explains the columns for the data listed in the **Next Start Times** window:

Column	Description
Cmd	One-character input field for line commands. Possible values are listed in the command section of the screen (see also <i>Line Commands: Next Start Times</i>).
Date	Date of planned network start. See also <i>Date and Time Formats</i> .
Act.	Time of planned network start. See also <i>Date and Time Formats</i> .
Start	Time of planned start of network in its operating system.
Run	Run number for this activation.
Type	Displays the current status of the activation. Possible status types are, for example: <ul style="list-style-type: none"> ■ Schedule Table, ■ on request ■ Schedule, active ■ Activation in Progress ■ Aw. Symbol Prompting ■ Symbol Entry in Progress

Special PF Keys: Next Start Times

The **Next Start Times** window provides the following special PF key:

PF Key	Name	Function
PF2	Add	Activates a network manually: see <i>Activating a Job Network Manually</i> in the section <i>Network Maintenance</i> .

Line Commands: Next Start Times

The following line commands are available for planned starts (activations) of job networks to which run numbers are assigned:

Line Command	Description	
D	Delete Cancels a planned network start .	
M	Start Time Modifies the start time of a planned network start.	
S	Active Symbols	
	When you enter this command for a network whose Type is:	... you can ... :
	Schedule, active	... modify the active symbols in the active symbol table(s) of the network.
	Awaiting Symbol Prompting	... perform symbol prompting for this planned start.

Canceling a planned Network Start

You can cancel a planned network start only if a run number has already been assigned to the network.

A planned network start without a run number can be removed by modifying the **network schedule definition** or the schedule linked to the network schedule definition.

If an activation is removed for a scheduled date, the date is excluded from the **network schedule definition (explicit dates)** to prevent further activations for that date.

Symbol Prompting

You can perform symbol prompting only if a run number is assigned to the network planned to start (be activated).

➤ To perform symbol prompting for a planned start

- 1 In the **Next Start Times** window, enter S in the line command field of the appropriate job or network with **Type** = Aw. Symbol Prompting.
- 2 Press ENTER.
- 3 A **Symbol Prompting** screen appears.

For further information, see *Symbol Prompting during Network or Job Activation* in the section *Symbol Table and Symbol Maintenance*.

Modifying an Active Symbol Table

➤ To modify the active symbols in a network's active symbol table(s)

- 1 In the **Next Start Times** window, enter S in the line command field of the appropriate job or network with **Type** = Schedule, active.
- 2 Press ENTER.
- 3 An **Active Symbol Tables** window similar to the example below opens:

```

29.10.08          ***** Entire Operations *****          14:31:54
Owner EXAMPLE          Schedule DEMO-NET   Year 2007          Calendar DEMOCAL
+-----+-----+-----+-----+-----+-----+-----+-----+
!
Monday             ! Owner EXAMPLE   Network DEMO-NET   Run 136           !
Tuesday            !                   Active Symbol Tables                !
Wednesday          !                                     !
Thursday           !      Cmd Table           Cmd Table           !
Friday             !      _ SN-01            _                   !ting             !
Saturday           !      _ DEMO             _                   !ting             !
Sunday             !      _                  _                   !ting             !
!                  _                   _                   !ting             !
!                  _                   _                   !ting             !
Monday             !      _                   _                   !ting             !
Tuesday            !      _                   _                   !ting             !
Wednesday          !      _                   _                   !Progress        !
Thursday           !      _                   _                   !ting             !
Friday             !      _                   _                   !ting             !
Saturday           !      _                   _                   !-----         !
Sunday             ! Select Symbol Table by marking.           !
!                  PF3 End                    !
+-----+-----+-----+-----+-----+-----+-----+-----+
Enter-PF1---P +-----+-----+-----+-----+-----+-----+-----+
                Help      End  Overv          Up   Down  Calen Histo NxtSt

```

- 4 Mark the appropriate symbol table with any character. Press ENTER.

An **Active Symbol Table** screen similar to the example below appears:

```

29.10.08          ***** Entire Operations *****          14:52:44
Owner   EXAMPLE          Active Symbol Table SN-01
Network DEMO-NET          Run 136
-----
C Symbol          F P Value          modified by
_ A1             A N 'HUGO'          SN          07.12.05 14:09
_ A2             A N '''QUOTED'''      SN          07.12.05 14:10
_ A3             A E 'XXX  '          SN          26.03.06 15:33
_ A4             A E ' '              SN          22.04.06 15:42
_ A5             A E AAA              SN          27.04.06 11:28
_ A6             N A 777              SN          17.05.06 10:25
_ A7             A E PROMPTED AT 14:25 SN          20.05.06 14:20
_ A8             A E DFFDDFFD        SN          20.05.06 14:20
_ CLASS          A A G                SN1         22.05.06 11:45
_ CLASS-PRIV     A E *** empty ***   SN1         22.05.06 11:42
_ DATE           D A 20060818        SN          18.08.06 15:19
_ EMPTY          A E YYYYYYYYYY     SN          11.09.06 11:31
_ HUGO           A E A VERY LONG STRING TO TEST S SN          22.10.06 10:31
_ JOBLIB         A E NOP.DEV.LOAD    SN          18.11.06 14:21
***** More *****
D Delete  M Modify
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Add  End      Save      Up      Down      Print

```

5 Enter M in the line command field of the appropriate symbol. Press ENTER.

An **Active Symbol Modification window** opens for the symbol selected.

The window is identical to the **Master Symbol Addition** window. For further information, see the following descriptions in the *Symbol Table and Symbol Maintenance* section:

- *Entering Long Symbol Values and Modifying Active Symbols,*
- *Fields and Columns: Master and Active Symbols,* and
- *Special PF Keys: Master Symbol Addition.*

6 Modify the symbol(s) as required.

7 Press PF5 (Save) to save the modified active symbol table.

8 Press PF3 (End) twice to close the **Active Symbol Modification** window and return to the **Next Activations** window.

Columns: Active Symbol Tables

The following table explains the columns for the data listed in the [Active Symbol Tables](#) window:

Column	Description
Cmd	One-character input field for line commands Select an active symbol table by any character.
SymTab	Name of the symbol table
Version	Version of the symbol table

Line Commands: Active Symbol Tables

Use the following line command to perform the described function on any network start listed on the [Next scheduled Network Starts](#) screen.

Line Command	Description
any character	List or modify active symbol table. The Active Symbol Table window opens.

Next Planned Network Starts**> To display a system-wide list of all planned job and network starts**

- Press PF11 (NxtSt) from the **Network Maintenance** or **Active Job Networks** screen.

A **Next scheduled Network Starts** screen similar to the example below appears:

```

16-07-18          ***** Entire Operations *****          10:35:21
                   Next scheduled Network Starts
-----
C Owner          Network          Run Version  Job          Date  Start Type
*-----*-----
_ SAGNET         N1459C          22171      22171      07-18 18:30 Schedule, acti
_ SAGNET         N1459C          22172      22172      07-18 19:30 Schedule, acti
_ SAGTEST        SAGTEST         39         39         07-18 20:00 Schedule, acti
_ SAGNET         N1459C          22173      22173      07-18 20:30 Schedule, acti
_ SAGTEST        SAGTEST         40         40         07-18 20:30 Schedule, acti
_ SAGTEST        SAGTEST         41         41         07-18 21:00 Schedule, acti
_ SAGNET         N1459C          22174      22174      07-18 21:30 Schedule, acti
_ SAGTEST        SAGTEST         42         42         07-18 21:30 Schedule, acti
_ SAGTEST        SAGTEST         43         43         07-18 22:00 Schedule, acti
_ SAGNET         N1459C          22175      22175      07-18 22:30 Schedule, acti
_ INCIDENT       I5095089A      1088      1088      07-18 23:40 Schedule, acti
_ INCIDENT       I1042163       576       576       07-19 00:00 Schedule, acti
_ SAGNET         A-ITRAC        325       325       07-19 00:00 Schedule, acti
***** m o r e *****
D Deactivate    M Start Time    S Active Symbols
Command => _____

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      End              Up      Down      Left  Right Menu

```

This screen displays a chronological list of all planned network or job starts, both scheduled and manual.

Line Commands: Next Scheduled Network Starts

Use the line commands described in [Line Commands: Next Start Times](#) to perform the described functions on any network start listed on the [Next scheduled Network Starts](#) screen.

Columns: Next Scheduled Network Starts

The following table explains the columns for the data listed on the [Next scheduled Network Starts screen](#):

Column	Description
C	One-character input field for line commands: see Line Commands: Next Scheduled Network Starts .
Owner	Owner of the network to be activated Above this column, you can make an owner preselection. If this field only contains an asterisk (*), all owners will be displayed. Your selection is stored in the user profile and is available when you invoke this function the next time.

Column	Description
Network	<p>Network to be activated</p> <p>Above this column, you can make a network preselection. If this field only contains an asterisk (*), all networks for the current owner selection will be displayed. The network preselection is possible only if an owner was selected uniquely.</p> <p>Your selection is stored in the user profile and is available when you invoke this function the next time.</p>
Run	Run number for this activation.
Version	Version of the network
Job	If the activation is for one job only, the job name appears in this column.
Date	<p>Date of activation</p> <p>You can enter a start date and/or time in the fields above the Date and Start columns to display only those activations due to start after that time.</p> <p>For valid date and time input formats, see Date and Time Formats.</p>
Start	Starting time of activation.
Type	<p>Displays the current status of the activation. Possible status types are, for example:</p> <ul style="list-style-type: none"> ■ Schedule Table, ■ on request, ■ Schedule, ■ active, ■ Aw. Symbol Prompting, ■ Symbol Entry in Progress.

Modifying Planned Network Start Times

You can modify a network start time only if a run number is assigned to the network planned to start (be activated).

➤ To modify the start time of a planned network activation

- 1 Enter **M** in the line command field of the appropriate network. Press **ENTER**.

A **Start Time Modification** window opens:

```

29.10.08          ***** Entire Operations *****          15:17:32
Owner SN          Schedule TIME-PCSN1 Year 2006
+-----+-----+
!               !               !               !               !
Monday          !   +-----+-----+               !               !
Tuesday         !   !               !               !               !
Wednesday       !   !               Start Time Modification   !               !
Thursday        !   C !               !               !               !
Friday          !   m !   Owner    ==> SN                    !               !
Saturday        !   !   Network  ==> TIME-PCSN1             !               !
Sunday          !   !   Job      ==> -                      !               !
                !   !   Run     ==> 527                   !               !
                !   !               !               !               !
Monday          !   !   Start   !               !               !
Tuesday         !   !   planned ==> 06.07.06 00:00         !               !
Wednesday       !   !   new    ==> 06.07.06 00:00         !               !
Thursday        !   !               !               !               !
Friday          !   !   Keep predefined Job Time Frames ==> N !               !
Saturday        !   - !               !   --- !               !
Sunday          !   D !   -PF1-----PF3-----PF5----- !               !
                !   P !   Help   End   Save                !               !
Enter-PF1---P +---+-----+-----+               +-----+-----+
                Help   End   Overv           Up   Down   Calen Histo NxtSt

```

- 2 You can enter a new date and start time, but they must be greater than or equal to the current values.
- 3 Press PF5 (Save) to save the new start time.
- 4 Press PF3 (End) to return to the [Next scheduled Network Starts](#) screen.

Fields: Start Time Modification

The input field of the **Start Time Modification** window is described in the following table:

Field	Description	
Keep predefined Job Time Frames	N	All job time frames will be adapted (default value).
	Y	Jobs with master time frame definitions will not be adapted.

Notes:

1. A start time up to the current time plus “activation before earliest start” causes an immediate activation of the jobs of the network.
2. The job time frames will be shifted with the same time offset as the planned network start, as long as the job time frames fit into the network time frame.

Displaying the Network Execution History

The execution history is a record of past schedules. It shows the days on which the network was scheduled for execution. The data on a **History** screen cannot be modified.

There is history data on past network executions for the current and for the two preceding years (provided the network already existed at that time).

> To display the execution history

- 1 Type **X** in the line command field of the selected network. Press **ENTER**.

A **History** screen similar to the example below appears. The display is in calendar format starting with the current half year. The dates on which the network ran appear on the history:

```

29.10.08          ***** Entire Operations *****          15:26:21
Owner EXAMPLE          History DEMO-NET   Year 2007          Calendar DEMOCAL

                January                February                March
Monday          .  8  .  .  .          .  . 19  .          .  .  .  .
Tuesday         .  . 16  .  .          .  .  .  .          .  .  .  .
Wednesday       .  .  .  .  .          .  .  .  .          .  .  .  .
Thursday        . 11 18  .          .  . 15  .          1  .  .  .  .
Friday          . 12 19  .          .  . 16  .          2  .  .  .  .
Saturday        .  .  .  .          .  .  .  .          .  .  .  .
Sunday         .  .  .  .          .  .  .  .          .  .  .  .

                April                May                June
Monday          .  .  .  .  .          .  .  .  .          .  .  .  .
Tuesday         .  .  .  .          .  .  .  .          .  .  .  .
Wednesday       .  .  .  .          .  .  .  .          .  .  .  .
Thursday        .  .  .  .          .  .  .  .          .  .  .  .
Friday          .  .  .  .          .  .  .  .          .  .  .  .
Saturday        .  .  .  .          .  .  .  .          .  .  .  .
Sunday         .  .  .  .          .  .  .  .          .  .  .  .

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      End   Day          Up   Down

```

You can enter any valid year in the **Year** field in the header section of the screen to display the execution history for the specified year.

- 2 Press **PF8** (Down) to display the next half of the year and **PF7** (Up) to display the previous half of the year. You can also use these PF keys to change to the previous or following year. If no history data exists for a given year, an appropriate message is issued.

- 3 To display the execution history for a particular day:

Under z/OS: Place the cursor on the appropriate day and press PF4 (Day).

Or:

Else: Press PF4 (Day). A window opens. Enter the appropriate day and press ENTER.

A **Network Start History** screen appears with the detailed execution history for one day:

```

16-07-07          ***** Entire Operations *****          12:27:17
Owner  SAGTEST          Network Start History          Network SAGNET
                                         Date 16-07-04
-----
   Run  Version      Job          Date      Time
   7                (Network)  16-07-04  10:33:18
   8                (Network)  16-07-04  14:30:50

***** Bottom of Data *****

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      End                Up      Down

```

If history data exists for the selected network and date, execution information on all network starts is provided on the screen as shown in the example above. The screen columns are explained in [Columns: Network History](#).

This section covers the following topic:

- [Columns: Network History](#)

Columns: Network History

The columns of the [Network Start History screen](#) are described in the following table:

Column	Description
Run	Run number of the job network that executed.
Version	Network version (if available) of the network that executed.
Job	Job name if a single job was activated for execution, or (Network) if the entire network was activated.
Date	Date on which the network executed. See also Date and Time Formats .
Time	Time at which the network executed. See also Date and Time Formats .

60

Scheduling a Job

- Defining Scheduling Parameters for a Job 544
- Maintaining Schedule Dependencies for a Job 547
- Defining Schedule Dependencies for an Input Condition 552
- Sending a Late Message to One or More Users 552
- Viewing Job Accounting Data 553

Defining Scheduling Parameters for a Job

This feature allows you to define a schedule for the job master. You can define the time frame for the schedule, insert a message, assign account information and edit the schedule dependency information.

Related Topic:

■ *Rules for Time Frame Definitions*

» To define scheduling parameters at the job level

- 1 Type S in the line command input field of the selected job on the **Job Maintenance** screen.
- 2 Press ENTER.

A **Scheduling Parameters** window similar to the example below opens:

```

16-06-29          ***** Entire Operations *****          17:19:30
Job Maintenance   Owner SAGTEST   Network B60-FLOW   Version
+-----+-----+-----+-----+-----+-----+-----+
!
!                               Scheduling Parameters                               !
! Job JOB-1-TEST                                                    ! E
!
! Estimated Elapsed Time ==> _____ Min.   Average ==>   0.00   !
! History Elapsed Time  ==>   0.00   0.00   0.00   0.00   0.00   !
!                               0.00   0.00   0.00   0.00   !
! Cyclic Interval       ==> _____ min   !
!
! Earliest Start Time   ==> 13:14:00  ___ days later   !
! Latest Start Time     ==> 23:00:00  ___ days later   !
! Deadline Time         ==> 23:30:00  ___ days later   !
!
! Schedule Dependency   ==> N                                     ! 1
!                                                                ! *
! ---PF1---PF3-----PF5-----PF9---PF10---PF11---PF12-- !
!   Help  End         Save          LMsg Acct  ScDep  Menu  !
+-----+-----+-----+-----+-----+-----+-----+

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Add  End         Save          Up    Down                               Menu

```

If scheduling parameters have already been defined, they appear in this window and you can modify them for the job.

When defining a new schedule, some default values are already displayed. You can replace them as required.

The input fields are explained in [Fields: Scheduling Parameters](#).

- 3 Press PF5 (Save) to save the job scheduling parameters.
- 4 Press PF3 (End) to return to the **Job Maintenance** screen.

This section covers the following topics:

- [Fields: Scheduling Parameters](#)
- [Special PF Keys: Scheduling Parameters](#)

Fields: Scheduling Parameters

The fields of the [Scheduling Parameters window](#) are described in the following table.

Field	Description
Estimated Elapsed Time	<p>The expected total running time (in minutes) for the job.</p> <ul style="list-style-type: none"> ■ If not specified explicitly, this value is derived from the average running time of previous runs of the same job, displayed in the field directly below. ■ If, for a real dummy job (job type DUM), this value is not equal to zero (0), then the dummy job is executed for exactly this amount of time. This enables you to perform time simulations, etc. ■ Active scheduling parameters for a running dummy job: <p>Modification of the estimated elapsed time causes the end time of the dummy job to be recalculated as sum of the start time and the new estimated elapsed time. If the new calculated end time is lower than the current time, the current time will be used instead.</p> <ul style="list-style-type: none"> ■ For temporary dummy jobs (e.g. dummy due to a condition), the defined estimated elapsed time is ignored. Temporary dummy jobs always have the execution time zero. <p>Special value: For jobs of the type NET, 999999 can be specified. In this case, all start times after that are set to the ending times in the subnetwork.</p>
Average	Average running time computed from five previous runs (see the field History Elapsed Time).
History Elapsed Time	Shows real running times of previous runs of the job. The average running time of all previous runs is used to compute the probable time the job will finish.
Earliest Start Time	<p>The job will not be started before this time.</p> <p>Use days later if the Earliest Start Time is more than one day later than the network start time.</p>

Field	Description
	See also Date and Time Formats (section <i>Using Entire Operations</i>) and Behavior for Schedule Times .
Latest Start Time	The job will not be started after this time. If this time precedes the Earliest Start Time and days later is null, days later is set to 1 so this time follows the Earliest Start Time . See also Date and Time Formats (section <i>Using Entire Operations</i>) and Behavior for Schedule Times .
Deadline Time	Time by which the job must be finished If the job cannot be finished before this time, it is not submitted. Dummy jobs with Estimated Elapsed Time not equal to zero (0) are finished at this time at the latest. See also Date and Time Formats (section <i>Using Entire Operations</i>) and Behavior for Schedule Times .
days later	You can use this field for Earliest Start Time , Latest Start Time and Deadline Time to define a time span lasting longer than 24 hours. See also Date and Time Formats (section <i>Using Entire Operations</i>) and Behavior for Schedule Times .
Schedule Dependency	A Y (Yes) here indicates that this job has a schedule dependency. An N (No) indicates that no schedule dependency is defined for the job. See also Maintaining Schedule Dependencies for a Job .
Cyclic Interval	This field is only available for cyclic jobs. Wait time (in minutes) between two executions of a cyclic job (special type C). Default: Monitor wait time

Special PF Keys: Scheduling Parameters

The following special PF keys are provided in the **Scheduling Parameters** window:

PF Key	Name	Description
PF9	LMsg	Late Message Define users to be informed if the job was not executed in the defined time range. See also Sending a Late Message to One or More Users .
PF10	Acct	Accounting Display accounting information for the job.

PF Key	Name	Description
		See also Viewing Job Accounting Data .
PF11	ScDep	Schedule dependency Define days on which the job is to be replaced by a dummy job. See also Maintaining Schedule Dependencies for a Job .

Maintaining Schedule Dependencies for a Job

You can define whether job activation or input condition usage is valid (and checked) only on particular days of the week, month or year.

The date and time defined for job execution in the [Scheduling Parameters window](#) is then checked against the current schedule of the current network. For jobs in a subnetwork, we recommend you to specify that the topmost calling network is checked instead of the current network by using [Schedule Usage](#) option.



Note: The job execution date and time can be different (one or more days later) from the job activation date and time.

> To define schedule dependencies at the job level

- Press PF11 (ScDep) in the [Scheduling Parameters window](#).

A **Schedule Dependency Definition for Job** window similar to the example below opens:

```

17-10-06          ***** Entire Operations *****          15:29:43
Job Maintenance   Owner SAGTEST   Network SAGNET   Version
+-----+-----+-----+-----+-----+-----+-----+-----+
!           +-----+-----+-----+-----+-----+-----+
!           !                                           !
! Job DE !           Schedule Dependency Definition for Job           ! B
!           !                                           ! B
! Estima ! Owner   SAGTEST           Job       DEMO-JOB           !
! Histor ! Network SAGNET           Condition           !
!           !                                           !
!           ! The Job will be activated only (+) / not activated (-), !
!           ! if the execution date has the following position in the !
! Earlie ! schedule or calendar:           Schedule Usage ==> _ !
! Latest !                                           !
! Deadli !           Usage ==> _ + only           - except           !
!           !           A after Holiday too B before Holiday too ! 3
! Schedu !           Type ==> _ H is Holiday   X is Workday           !
!           !           C Calendar Day   W Workday   S Schedule Day ! *
! ---PF1- !           in Period ==> _ W Week           M Month   Y Year           ! s
! Help !           at Position ==> _____           !
+-----+ ! ---PF1-----PF3-----PF5-----PF9-----+
!           Help      End      Save      Delete           !
+-----+-----+-----+-----+-----+-----+-----+-----+
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Add   End   Save   Up   Down   Menu

```

The input fields contained in the window are described in [Fields: Schedule Dependency Definition](#).

This section covers the following topics:

- [Fields: Schedule Dependency Definition](#)
- [Examples of Schedule Dependency Definitions](#)
- [Deleting Schedule Dependency Entries](#)

Fields: Schedule Dependency Definition

The input fields of the **Schedule Dependency Definition** window provided for a job and/or input condition and the schedule dependency of a network are described in the following table:

Field	Description	
Schedule Usage	Possible input values:	
	blank	Checks the schedule of the current network (default). This setting is recommended for jobs in main networks.
	T	Checks the schedule of the topmost network that calls the job. This setting is recommended for jobs in subnetworks.
Usage (only or except)	Possible input values:	
	+	For a job or condition: The job or condition is activated only if the execution date satisfies the criteria entered in the Type, in Period and at Position fields.
		For a network: The object is valid only for schedule days matching the following definitions.
	A	Like +, but the first workday after an otherwise matching holiday is valid, too.
	B	Like +, but the last workday before an otherwise matching holiday is valid, too.
	-	For a job or condition: The job or condition is activated except when the execution date satisfies the criteria entered in the Type, in Period and at Position fields.
For a network: The object is valid for all schedule days except if they match the following definitions.		
Type	Possible input values:	
	H	Any holiday (no period and/or position required) The relative position to the holiday can be specified.
	X	Any workday (no period and/or position required) The relative position to the workday can be specified.
	C	Calendar day within the defined period A relative position refers to the real calendar period.
	W	Workday within the defined period A relative position refers to the workdays in the period.
	S	Schedule day within the defined period A relative position refers to the schedule days in the period.

Field	Description	
in Period	Possible input values:	
	W	Current week
	M	Current month
	Y	Current year
at Position	<p>Schedule dependency position</p> <p>The position within the defined period depending on the specified type.</p> <p>Positive values (+) are relative to the period begin, negative values (-) are relative to the period end. For type H and X, this is the day difference.</p> <p>Valid value ranges are:</p> <p>For a week: -7 to -1 and +1 to +7 For a month: -31 to -1 and +1 to +31 For a year: -366 to -1 and +1 to +366 A value of zero (0) is not allowed.</p> <p>Note: You can specify up to three days for the period Week. Example: 24 is Tuesday and Thursday.</p> <p>See also Examples of Schedule Dependency Definitions.</p>	

Examples of Schedule Dependency Definitions

The following are example combinations for day calculation:

Type	Period	Position	Calculated Day
C	W	+2	Tuesday
		-1	Sunday
	M	+1	01
		-1	28 ... 31
	Y	+1	January 01
		-3	December 29
W	M	-1	Last workday of the month
H	n/a	+1	Day after a holiday

Special PF Keys: Schedule Dependency Definition for Job

The following special PF key is provided in the [Schedule Dependency Definition for Job window](#):

PF Key	Name	Function
PF9	Delete	Delete entries in the fields Usage, Type, in Period and at Position . See also <i>Deleting Dependency Entries</i> .

Deleting Schedule Dependency Entries

➤ To delete a schedule dependency entry for a job

- 1 In the **Schedule Dependency Definition for Job** window, press PF9 (Delete).

The following window opens:

```

16-07-07          ***** Entire Operations *****          13:51:11
Job Maintenance      Owner SAGTEST      Network B60-FLOW      Version
+-----+-----+-----+-----+
!           +-----+-----+-----+-----+
!           !                                           !
! Job J0 !           Schedule Dependency Definition for Job !
!           !                                           !
! Estima ! Owner   SAGTEST           Job      JOB-1-TEST !
! Histor ! Network B60-FLOW           Condition !
!           +-----+-----+-----+-----+
!           !                                           !
!           ! Please confirm                               !
! Earl ! the Deletion of DEPENDENCY                       !
! Late ! by entering 'Y' ==> _                             !
! Dead !                                                    !
!           ! PF3 End                                     !
! Sche +-----+-----+-----+-----+ ! 1
!           !                                           !
! ---PF1- !           C Calendar Day  W Workday  S Schedule Day ! *
! Help ! in Period ==> _  W Week           M Month    Y Year      !
!           ! at Position ==> _____ !
+-----+ ! ---PF1-----PF3-----PF5-----PF9----- !
!           ! Help      End      Save      Delete      !
+-----+-----+-----+-----+

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      Help  Add   End   Save   Up    Down                               Menu

```

- 2 Enter Y (Yes) and press ENTER to delete all field entries.
- 3 Press PF3 (End) to close the window.

Defining Schedule Dependencies for an Input Condition

You can define an input condition that is not used on all activation dates of a job network. This can be done for the following reasons:

- A predecessor job or network does not run on all schedule dates of the successor job or network.
- Some predecessor jobs are also schedule-dependent.

➤ To define schedule dependencies for an input condition

- 1 On the **Input Conditions Maintenance screen**, type S in the line command input field of the selected condition.
- 2 Press ENTER.

A **Schedule Dependency Definition for Input Condition** window opens.

The fields contained in the window correspond to the **fields** of the **Schedule Dependency Definition for Job window**. They are explained in *Maintaining Schedule Dependencies for a Job*.

- 3 Make your definitions and choose PF5 to save your entries.

The schedule dependency is allocated to the input condition.

- 4 Press PF3 to return to the **Input Conditions Maintenance** screen.

The schedule dependency is indicated in the **Sched.Dep.** column.

Sending a Late Message to One or More Users

You can define recipients (users) to be informed in the case that the job could not be executed in the defined time range.

➤ To send a message informing one or more users that the job could not be submitted

- In the **Scheduling Parameters** window, press PF9 (LMsg).

A **Message and Message Recipients window** opens in which you can define message destinations.

The fields contained in the window are explained in *Field and Columns: Message and Message Recipients* in the section *End-of-Job Checking and Actions*.

Viewing Job Accounting Data

You can view the start and end times, elapsed time and CPU time for previous runs of a job. The information is taken from the Entire Operations log file.



Note: This function is also available for job networks: see [Displaying Accounting Data for Jobs in a Selected Network](#) in the section *Network Maintenance*.

You can also generate reports from accounting data (see [Example of Accounting Data](#) as described in the section *Reporting*).

➤ To display job accounting data

- 1 In the [Scheduling Parameters window](#), press PF10 (Acct).

A **Date / Run Number Selection** window similar to the example below opens:

```

29.10.08          ***** Entire Operations *****          15:32:20
Owner SN          Job Maintenance          Network BIG-1
+-----+-----+-----+-----+-----+-----+-----+-----+
!
!                      Scheduling Parameters                      !
! Job JOB-014                                               !2
!                                               Tape Drives ==> ___ !1
!                                               !1
! Estimated Elapsed Time ==> _____ Min.   Average ==>   7.00 !1
! History Elapsed Time  ==>   2.00   12.00          !
!                                               !1
!                                               !2
! Earliest +-----+-----+-----+-----+-----+-----+ !1
! Latest St !                                               ! !1
! Deadline ! Owner SN Network BIG-1 Job JOB-014          ! !
!         !                      Date / Run Number Selection ! !2
! Schedule !                                               ! !
!         ! From Date 10.07.07 00:00 to 10.07.07 16:00 ! !**
! ---PF1---P ! From Run 1_____ to 99999          !12-- !
! Help E !                                               !nu !g
+-----+-----+-----+-----+-----+-----+ !-----+
Enter-PF1---PF +-----+-----+-----+-----+-----+ +1---PF12---
      Help Add End Save Up Down Menu

```

The window displays the time and run number interval.

Enter dates, times and run numbers to specify the period and number range for which you want to view accounting data: see [Fields: Range Specification for Accounting Data](#).

- 2 Press ENTER.

A **Job Accounting Data** screen similar to the example below appears:

```

29.10.08          ***** Entire Operations *****          15:46:18
Owner  SN                Job Accounting Data                Network DEMO-NET
-----
Job      Run  JobId  Date      Start      Stop      Elapsed min CPU Time sec
JOB1     748  1450  10.05.07  10:00:09   10:00:09   0.01      0.01
          749  1783                12:00:30   12:00:31   0.01      0.01
          750  1005                14:00:22   14:00:23   0.01      0.01
          752  1867  10.05.07  12:00:16   12:00:16   0.01      0.01
          757  3975  04.06.07  10:00:06   10:00:06   0.01      0.01
          758  4314                12:00:17   12:00:18   0.01      0.01
          759  4679                14:00:09   14:00:10   0.01      0.01
          765  4085  10.06.07  14:08:57   14:08:57   0.01      0.01

          -----
Averages from 10.05.07 10:00 to 10.06.07 14:08 are          0.01      0.01
***** Bottom of Data *****
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
          End                Up      Down
    
```

The screen contains job execution time information collected by the Entire Operations Monitor. The job elapsed time is in minutes and the CPU time is in seconds. Average values are available at the bottom of the screen.

For further information, see [Fields and Columns: Reports](#) in the section *Reporting*.

This section covers the following topics:

- [Fields: Range Specification for Accounting Data](#)
- [Job Accounting Data: Operating System Considerations](#)

Fields: Range Specification for Accounting Data

The fields of the **Date / Run Number Selection** window are described in the following table:

Field	Description
From Date ... to	<p>Enter start (From) and end (To) dates and times to specify the time frame for which to display accounting data.</p> <p>Default is the current date, from midnight until the current time.</p> <p>For valid date and time input formats, see Date and Time Formats.</p>
From Run ... to	<p>Enter start (From) and end (To) run numbers to specify the number range for which to display accounting data.</p> <p>Default is from 1 to 99999.</p>

Job Accounting Data: Operating System Considerations

BS2000

Since the BS2000 LOGOFF message does not contain seconds, the elapsed time cannot be calculated exactly.

UNIX

CPU time information is written to the SYSOUT of UNIX jobs. For this purpose, the shell `times` command is used. The output is enclosed in the messages EOR0303 and EOR0304.

Example:

```
%% EOR0303 - times - Begin
0m0.121s 0m0.025s
0m0.043s 0m0.066s
%% EOR0304 - times - End
```

Notes:

- With Entire System Server for UNIX Version 2.1.2 PL 4 and above, this output is used to determine the CPU time consumption of UNIX jobs. The CPU time is included in the Entire Operations accounting data.
- For Sun Solaris:

The `times` command output is useful only if the script frame (*.BF) is executed under the Korn shell. Therefore, the *.BF scripts on Sun Solaris are started under the Korn shell.

X **Calendar Maintenance**

61 Calendar Maintenance

▪ General Rules and Restrictions	560
▪ Listing Calendars	561
▪ Displaying or Modifying a Calendar Definition	564
▪ Adding a Calendar Definition	567
▪ Defining Workdays and Holidays	568
▪ Where Used - Listing Schedules Using a Calendar	571
▪ Deleting a Calendar Definition	571

Calendars are referenced by schedules which are defined in the network maintenance facility. If a schedule depends on a calendar, calendar-dependent definitions can be made in the schedule. If a schedule is not linked to a calendar, the schedule assumes that all days are workdays.

Any number of calendars can be defined to the system. Calendars can belong to an owner or be used system-wide.

In the calendar maintenance facility, you can add, delete or update a calendar (system-wide calendars can only be modified by the system administrator).



Note: For further information on calendars, see *Calendar* in the *Concepts and Facilities* documentation.

General Rules and Restrictions

This section covers the following topics:

- [How Calendars Work](#)
- [Calendar Types](#)
- [Calendar Names](#)

How Calendars Work

Entire Operations calendars include workdays and holidays (non-workdays).

Entire Operations does not activate any networks on a day defined as a holiday.

For example, a job network is scheduled to run on every first day of a month and the schedule is based on a calendar in which Saturdays and Sundays are defined as non-workdays:

If the first day of a month is a Saturday or Sunday, Entire Operations does not start the network on these days, but either delays activation until the next workday following the holidays or brings activation forward to the last workday previous to the holidays, depending on the schedule definition.

Calendar Types

There are two types of calendars in Entire Operations:

- **System-wide calendars**

System calendars can be modified by authorized users only.

- **Owner-specific calendars**

When you define a calendar, it is automatically associated with your owner. You can modify only those calendars that belong to your owner.

You can specify a calendar belonging to your owner for a schedule, but you can also select a system calendar.

Calendar Names

A calendar can be valid for several years, so you are not forced to change the name of the calendar at the end of the year. Just define the calendar for the next year. For this reason, you should use calendar names independent of the year.

You must define a calendar separately for each year you want to use it. Networks cannot be started if a calendar is missing for the year for which it is needed. To be notified about such cases, select the Calendar undefined for year option in Global Messages for Events (see the *Administration* documentation). In November and December, the Entire Operations Monitor writes warning messages to the log if a calendar currently used is not defined for the next year.

Reserved Calendar Names

Calendar names must not begin with the prefix =EOR=. This prefix is reserved for **special usage in symbol functions** (see the section *Symbol Table and Symbol Maintenance*).

Listing Calendars

> To list calendars

- Select **Calendar Maintenance** option from the Main Menu.

The **Calendar Maintenance** screen appears with a list of calendars already defined:

```

17-11-22          ***** Entire Operations *****          12:03:32
Owner SAGTEST          Calendar Maintenance
-----
Cmd  Owner      Calendar  Year  Description
   _  SAGTEST--- *-----  ----
   _  SAGTEST    ADMIN     2017  Calendar for 2017
   _  SAGTEST    ADMIN     2018  Calendar for 2018
   _  SAGTEST    CAL-TEST  2016  Demo Calendar for 2016
   _  SAGTEST    CAL-TEST  2017  Demo Calendar for 2017
   _
   _
   _
   _
   _
   _
   _
   _
   _
   _
***** Bottom of Data *****
C Copy  D Delete  L List  M Modify  S Description  W Where used
Command => _____

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Add   End       Save       Up       Down              Menu

```

The list of calendars contains system-wide calendars and calendars belonging to the selected owner (here: SAGTEST). Calendars belonging to other owners can also be displayed. You can enter an asterisk (*) in the **Owner** column to display all calendars available. See also the description of **Owner** in *Columns: Calendar Maintenance*.

A calendar is identified uniquely by its owner name and year.

This section covers the following topics:

- [Columns: Calendar Maintenance](#)
- [Line Commands: Calendar Maintenance](#)

- Special PF Keys: Calendar Maintenance

Columns: Calendar Maintenance

The following table explains the columns contained in the [Calendar Maintenance screen](#):

Column	Description
Cmd	One-character line command input field. For possible values, see Line Commands: Calendar Maintenance .
Owner	Owner of calendar (*). You can modify only your own calendars and those of linked owners.
Calendar	Calendar name (*).
Year	Year of the calendar (*).
Description	Short description. Use the line command S to add or modify a short description.

* You can filter the list by entering selection criteria in the input field below the column heading; see [Filtering Items in a Result List](#).

Line Commands: Calendar Maintenance

The following line commands are available on the [Calendar Maintenance screen](#):

Line Command	Description
C	Copy the selected calendar when defining a new one. See Copying Calendars .
D	Delete the selected calendar. (If not specified for any schedule.) See Deleting a Calendar Definition .
L	Display the selected calendar. (You cannot modify the calendar with this line command.) See Displaying or Modifying a Calendar Definition .
M	Modify selected calendar. Only owner-specific calendars can be modified. See Displaying or Modifying a Calendar Definition .
S	Opens a window to enter a calendar short description. See Adding or Modifying a Calendar Short Description .

Line Command	Description
W	List schedules for which this calendar is specified. See <i>Where Used - Listing Schedules Using a Calendar</i> .

Special PF Keys: Calendar Maintenance

The following special PF key is provided on the [Calendar Maintenance screen](#):

PF Key	Name	Function
PF2	Add	Add a calendar. See <i>Adding a Calendar Definition</i> .

Displaying or Modifying a Calendar Definition

➤ To display or modify a calendar definition

- 1 On the [Calendar Maintenance screen](#), type L in the line command input field next to the calendar you want to view. (You cannot modify the calendar with this line command.)

Or:

Type M in the line command input field next to the calendar you want to modify.

- 2 Press ENTER.

A calendar definition screen similar to the example below appears:

```

17-11-21          ***** Entire Operations *****          17:51:23
Owner SAGTEST          Calendar ADMIN          Year 2017

          July              August              September
Monday      3 10 17 24 31          7 14 21 28          4 11 18 25
Tuesday     4 11 18 25          1  8 15 22 29          5 12 19 26
Wednesday   5 12 19 26          2  9 16 23 30          6 13 20 27
Thursday    6 13 20 27          3 10 17 24 31          7 14 21 28
Friday      7 14 21 28          4 11 18 25          1  8 15 22 29
Saturday    1  8 15 22 29          5 12 19 26          2  9 16 23 30
Sunday      .  .  .  .  .          .  .  .  .  .          .  .  .  .

          October            November            December
Monday      2  9 16 23 30          6 13 20 27          4 11 18 25
Tuesday     3 10 17 24 31          7 14 21 28          5 12 19 26
Wednesday   4 11 18 25          1  8 15 22 29          6 13 20 27
Thursday    5 12 19 26          2  9 16 23 30          7 14 21 28
Friday      6 13 20 27          3 10 17 24          1  8 15 22 29
Saturday    7 14 21 28          4 11 18 25          2  9 16 23 30
Sunday      .  .  .  .  .          .  .  .  .  .          .  .  .  .

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
          Help          End          Up          Down

```

If the calendar is for the current year, the screen displays six months from the current month of the specified calendar year by default.

For other years, the first six months are displayed initially.

Current holidays are marked as dots (.) or not highlighted. For more information, see [Calendar Display Modes](#).

You can now modify the calendar as described in [Defining Workdays and Holidays](#).

- 3 Press PF7 (Up) to display previous month, or PF8 (Down) to display next months.

Scroll the years by using PF7 (Up) to display the calendar with the same name of the previous year, and PF8 (Down) to display the calendar with the same name of the following year.

- 4 If no more definitions are available at either end, Entire Operations prompts you to define a calendar with the same name of the appropriate year in the calendar definition window.
- 5 If you selected the line command M, you can modify the calendar as described in [Defining Workdays and Holidays](#).
- 6 When you are finished, press PF3 (End) to return to the **Calendar Maintenance** screen.

This section covers the following topics:

- [Calendar Display Modes](#)

- [Adding or Modifying a Calendar Short Description](#)

Calendar Display Modes

- For terminals which support highlighting or colors:
 - Holidays appear not highlighted.
 - Workdays appear highlighted.
- If neither color nor highlighting are available:
 - Holidays appear as dots (.).
 - Workdays appear as normal calendar dates.

Set the **Calendar Display** option in the Entire Operations defaults (see the *Administration* documentation) to determine the default display mode.

Adding or Modifying a Calendar Short Description

➤ **To add or modify a calendar short description**

- 1 On the [Calendar Maintenance screen](#), enter the line command S next to the required calendar.

A **Calendar Description** window similar to the example below opens:

```
+-----+
!                                     !
!                               Calendar Description                       !
!                                     !
! Owner      ==> SAGTEST                                           !
! Calendar   ==> CAL-TEST                                           !
! Description ==>                                                 !
! Demo Calendar_____!
!                                     !
! Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11- !
!                                     End                               !
+-----+
```

Enter text and press ENTER.

- 2 Press PF3 (End).

The window closes and the calendar description is displayed in the calendar list on the [Calendar Maintenance screen](#).

Field	Description
Owner	Name of owner for whom the calendar is to be defined. The current owner is entered by default.
Calendar	Calendar name.
Year	Calendar year. The current year is entered by default.

Copying Calendars

When defining a new calendar, you can use an existing calendar as a model for the new definition.

➤ To copy calendars

- 1 Type **C** in the line command input field of the calendar to be copied.
- 2 Press **ENTER**.

A copy window opens with the name and year of the existing calendar.

- 3 In the **to calendar** field, enter a new name. (You cannot modify the year.)
- 4 Press **ENTER**.

The copy window closes and the new calendar appears in the list on the [Calendar Maintenance screen](#).

- 5 You can use the **M** line command to modify the new calendar.

Defining Workdays and Holidays

Defining a calendar consists of defining workdays and holidays (non-workdays). This is done in two steps:

1. Define weekly holidays. These are the non-workdays in every week, such as weekends.
2. Define special holidays or personal vacations.



Note: A network schedule can override workdays marked on a calendar, because a network runs on a day specified in a calendar, only if this day is also specified in the schedule.

➤ To define weekly holidays (non-workdays)

- 1 On the [calendar definition screen](#), press **PF4** (Wkdy).

A window opens with a list of weekdays:

```
+-----+
!
! Calendar ADMIN !
!   Year 2017   !
!
! Please mark weekly !
! holidays:      !
!
!   _ Monday    !
!   _ Tuesday   !
!   _ Wednesday !
!   _ Thursday  !
!   _ Friday    !
!   _ Saturday  !
!   _ Sunday    !
!
!   S Set  R Reset !
!   PF3 End        !
+-----+
```

- 2 Enter S (Set) next to the weekday(s) you want to define as holidays.

To unmark a weekly holiday to a workday, mark it with R)

Or:

Enter R (Reset) next to the defined weekly holidays you want to change back to weekdays.

- 3 Press ENTER.

The window closes and the defined holidays appear as dots (.) or are not highlighted.

The following example shows a calendar in which Saturday and Sunday are defined as weekly holidays:

```

17-11-22          ***** Entire Operations *****          14:56:29
Owner SAGTEST          Calendar ADMIN          Year 2017

          July              August              September
Monday      3 10 17 24 31          7 14 21 28          4 11 18 25
Tuesday     4 11 18 25          1  8 15 22 29          5 12 19 26
Wednesday   5 12 19 26          2  9 16 23 30          6 13 20 27
Thursday    6 13 20 27          3 10 17 24 31          7 14 21 28
Friday      7 14 21 28          4 11 18 25          1  8 15 22 29
Saturday    . . . . .          . . . . .          . . . . .
Sunday      . . . . .          . . . . .          . . . . .

          October           November           December
Monday      2  9 16 23 30          6 13 20 27          4 11 18 25
Tuesday     3 10 17 24 31          7 14 21 28          5 12 19 26
Wednesday   4 11 18 25          1  8 15 22 29          6 13 20 27
Thursday    5 12 19 26          2  9 16 23 30          7  . 21 28
Friday      6 13 20 27          3 10 17 24          1  8 15 22 29
Saturday    . . . . .          . . . . .          . . . . .
Sunday      . . . . .          . . . . .          . . . . .

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      End   Wkdy          Up   Down
    
```

 **Note:** The **format of the calendar** display depends on the setting of the **Calendar Display** option in the Entire Operations defaults (see the *Administration* documentation) and may therefore differ slightly from the format illustrated above.

➤ **To define or remove special holidays**

- 1 Mark dates: Overwrite or remove the required date(s) with any character and press ENTER.

The message `Date ... removed` confirms the action and the marked holiday(s) appear as dots (.) or are not highlighted.

Or:

Unmark a date: Overwrite dots (.) or days that are not highlighted and press ENTER to change marked holidays back to workdays. The message `Date ... added` confirms the action and the dots are replaced by dates.

Multiple selections in one input operation are possible.

- 2 If required, press PF8 (Down) or PF7 (Up) to display months not shown in the current view and mark or unmark additional dates.
- 3 When you are finished, press PF3 (End) to return to the **Calendar Maintenance screen**. The newly defined calendar appears in the list.

Where Used - Listing Schedules Using a Calendar

➤ To list schedules using a calendar

- 1 On the **Calendar Maintenance screen**, type `W` in the line command input field next to the required calendar.
- 2 Press `ENTER`.

A **Calendar Usage** window similar to the example below opens:

```
+-----+
!           Calendar Usage           !
!                                     !
! Owner      EXAMPLE                 !
! Calendar   EXAMPLECAL             !
!                                     !
! is used by                          !
! Owner      Schedule                !
! EXAMPLE    B60-FLOW                !
! EXAMPLE    MAY-DATES              !
! EXAMPLE    Q1-ADMIN               !
!                                     !
!                                     !
!                                     !
!                                     !
! ***** Bottom *****          !
! PF3 End   PF8 Down                !
+-----+ ←
```

The window indicates the name of the selected calendar and all schedules associated with the calendar.

- 3 Press `PF3` (End) to close the window and return to the **Calendar Maintenance** screen.

Deleting a Calendar Definition

➤ To delete a calendar definition

- 1 Type `D` in the line command input field of the selected calendar on the **Calendar Maintenance** screen. Press `ENTER`.

A window opens in which you can confirm deletion by entering the calendar name.

- 2 Enter the calendar name.
- 3 Press ENTER to perform the deletion.

You cannot delete a calendar if it is specified for at least one schedule. If you wish to delete a calendar which is still used by a defined schedule, Entire Operations notifies you with an appropriate message. Use the line command `W` to find out which schedule(s) are associated with the calendar (see [Where Used - Listing Schedules Using a Calendar](#)). In this case, you cannot delete the calendar until you have done either of the following:

- Specified another calendar for the relevant network schedule(s).

Or:

- Deleted the network for which the calendar is specified.

You can delete only the calendars you are permitted to delete.

XI Working with Mailboxes

62 Working with Mailboxes

- Viewing Mailbox Messages 576
- Turning Automatic Notifications On and Off 578
- Fields, Columns and Commands: Messages and Requests 578
- Message Types 579
- Origin of Mailbox Messages 582
- Group Mailbox 582
- SYSDBA Mailbox 582
- User Mailbox 582
- Clearing Mailboxes 583

Mailboxes are used for message switching to Entire Operations users. If a message requires a reply, it can be prompted in the mailbox window. Any input condition can be assigned to a user interaction (see also *Input Condition: Mailbox*). At the same time, a user ID can be associated with up to ten mailboxes. The message `You have mail` notifies you of mailbox requests. You can then take appropriate steps and manually set the conditions necessary for the job to continue.

 **Note:** You are allowed to display the contents of all mailboxes to which you are linked. The link is defined in the Entire Operations user definition.

Related Topics:

- *Mailboxes, Message Sending* in the *Concepts and Facilities* documentation,
- *Mailbox Definition* in the *Administration* documentation.

Viewing Mailbox Messages

If a mailbox assigned to you has any entries, the message `You have Mail` appears in the top left-hand corner of your screen:

```
EOR1500 - You have Mail
17-02-17          ***** Entire Operations *****          18:03:01
Owner SAGTEST          Main Menu          User ID SAGTEST
-----
Main Menu          DC Solutions
1 Network and Job Maintenance          20 ENTIRE Output Management
2 Active Job Networks          21 NATURAL ISPF
3 Schedule Maintenance
4 Calendar Maintenance
5 Log Information
6 Symbol Tables
7 System Administrator Services
8 Reports
9 Cross References
10 Import/Export
11 Help

Command => _____

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help          End          Owner Mail
```

If you want to turn this notification off, proceed as described in [Turning Automatic Notifications On and Off](#).

➤ **To view the messages for a mailbox**

- 1 Type MAIL in the Command line or press PF12 (Mail) on the Main Menu.

The MAIL direct command is described in the *Direct Commands* documentation.

- 2 Press ENTER.

The **Mailbox - Messages and Requests** screen appears:

```

11.02.10          ***** Entire Operations *****          10:01:23
User SN          Mailbox - Messages and Requests
-----
Cmd Date Time  Mailbox  Owner      Network    Run   Job
 11.02 10:01  -----
_ 11.02 08:01  SYSDBA    Symbol replacement error during submission
_                               ==> XSETAB01 / IMPXP2 / 1026 / JOB-LONG
_ 11.02 08:01  SYSDBA    Symbol replacement error during submission
_                               ==> XSETAB01 / IMPXP1 / 1619 / JOB-LONG
_ 11.02 00:18  SYSDBA    Network not terminated - no deactivation
_                               ==> XSETGR / T-COND / 1755
_ 11.02 00:18  SYSDBA    Network not terminated - no deactivation
_                               ==> TESTBED / NET01-537 / 1004
_ 11.02 00:18  SYSDBA    Network not terminated - no deactivation
_                               ==> TESTBED / NET01-533 / 504
_ 11.02 00:18  SYSDBA    Network not terminated - no deactivation
_                               ==> TESTBED / NET01-517 / 4717
***** m o r e *****
Sequence      : D (A/D)
Mark lines with any character for specific actions.
Command => _____
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      End      Save      Up      Down      Menu

```

The table view is filled dynamically with data if scrolled to the end of the table.

All messages and requests of all mailboxes to which the current Entire Operations user (owner at logon) is linked are listed in the table. For explanations of possible messages, see [Message Types](#).

The window columns and input fields provided to order, group or filter the messages are described in [Fields, Columns and Commands: Messages and Requests](#).

Turning Automatic Notifications On and Off

The notification `You have Mail` is turned on by default.

However, you can turn the notification on and off during the current session.

➤ To turn automatic notifications on and off

- Use the direct command `SET` as described in the *Direct Commands* documentation.

Fields, Columns and Commands: Messages and Requests

This section describes the fields, columns and commands provided on the [Mailbox - Messages and Requests screen](#).

- [Fields and Columns: Messages and Requests](#)
- [Line Commands](#)
- [Special PF Keys](#)

Fields and Columns: Messages and Requests

The following table explains the columns and fields of the **Mailbox - Messages and Requests** screen.

Column/Field	Description	
Cmd	One-character line command input field. Mark with any character. The requested action depends on the message type .	
Date/Time	Date and time when the message was sent. You can enter a start date and time in the input fields directly below the column headings. Enter * for the current date. See also Date and Time Formats .	
Mailbox	Mailbox to which the message was sent. Preselection is possible.	
Sequence	Sort order of the message list	
	Valid input values:	
	A	Ascending order. Earliest messages come first.
	D	Descending order. Latest messages come first.
Owner	Name of the owner. Preselection is possible.	
Network	Name of the network. Preselection is possible.	
Run	Number of the run. Preselection is possible.	

Column/Field	Description
Job	Name of the job. Preselection is possible.
<i>description</i>	Message text: see Message Types for explanations.

Using the preselection fields, you can select which messages you want to see. In the preselection fields, you may use wildcards.

Line Commands

➤ **To perform a specific function, like symbol prompting or setting a condition**

- Mark the **Cmd** field in the appropriate line with any character and press ENTER.

The function you can perform depends on the message type.

For more information, see the section [Message Types](#).

Special PF Keys

PF Key	Name	Function
PF7	Up	Scroll list of messages towards the top.
PF8	Down	Scroll list of messages towards the bottom.

Message Types

The types of messages that can be shown on the [Mailbox - Messages and Requests screen](#) are listed in the following section. The numbers in the example messages denote the job run numbers.

- [Waiting for Condition](#)
- [Symbol Prompting](#)
- [Information Messages](#)

Waiting for Condition

```
PAPER-READY - 549 - RUN not found
```

A message of this type means that a scheduled network or job is waiting for a condition to be set.

➤ **To set the condition**

- Mark the Cmd field of this line with any character. Press ENTER.

This function is used to perform manual actions within a network.

Symbol Prompting

```
Symbol Prompting E60-FLOW Run 530 (OVER START TIME)
```

A message of this type means that a scheduled network or job is waiting for symbols to be entered or modified.

> To perform symbol prompting

- 1 Mark the Cmd field of this line with any character. Press ENTER.

The Symbol Prompting screen appears.



Note: Symbol prompting via mailbox is only used for automatic jobs started by schedule and cannot be used for jobs started interactively. Definition of the recipient mailbox for symbol prompting messages is described in the subsection [Specifying Recipients for Network Messages](#) in the section *Network Maintenance*.

For further information on symbol prompting, see [Symbol Prompting during Network or Job Activation](#) in the section *Symbol Table and Symbol Maintenance*.

The message also displays the time until the next network start (in hours), or, if the network has exceeded its start time, displays the message

```
OVER START TIME
```

- 2 If this is the case, you are prompted for a new start time:

```

04.12.15          ***** Entire Operations *****          14:41:14
User SAG          Mailbox - Messages and Requests
-----
Cmd Date Time   Mailbox   Owner      Network    Run   Job
-----
_  28.11 00:00 SYSDBA   Symbol Prompt N983Z1 Run 763 (OVER START TIME)
_
_
_
x      Owner NOPALL      Network N2224-241      Run 7          ME)
_
_  The planned start time was in the past ==> 02.12.15  14:53
_  You can enter a new start time          ==> 02.12.15  14:53          ME)
_
_  or deactivate the activation with PF6          ME)
_
_  PF3  End    PF6  Deactivate          ME)
_
***** m o r e *****
Sequence   : A (A/D)
Mark lines with any character for specific actions.
Command => _____

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      End      Save      Up      Down      Menu

```

- 3 Enter the new start time.
- 4 Press **ENTER** to confirm the new start time. The new start time will be entered as the earliest start time in all active network jobs. Alternatively, you can deactivate the activation by pressing **PF6** (Deactive).

Information Messages

```

NETWORK DEMO-NET ACTIVATED ==> SN / DEMO-NET / 195 /
MESSAGE

```

This is an information message.

➤ To delete the message from the list on the Mailbox - Messages and Requests screen

- 1 Mark the Cmd field of this line with any character. Press **ENTER**.
A window opens.
- 2 Delete the message by entering **Y** and pressing **ENTER**.

Origin of Mailbox Messages

Messages can also be triggered by any other events detected by the Entire Operations Monitor:

1. If a condition is defined as dependent on a mailbox, the Entire Operations Monitor sends a request to the assigned mailbox. For further information, see [Input Condition: Mailbox](#) in the section [Job Maintenance](#).
2. If symbol modification without prompting is used and an error occurs, the request for symbol prompting is sent to the assigned mailbox. For further information, see [Specifying User Exits for Symbol Modification](#) in the section [Symbol Table and Symbol Maintenance](#) and [Specifying Recipients for Network Messages](#) in the section [Network Maintenance](#).
3. Depending on the result of an End-of-Job check, messages can be sent to Entire Operations users. Messages can also be sent to a particular mailbox. For further information, see [Message Sending](#) in the section [End-of-Job Checking and Actions](#).

Group Mailbox

All users linked to a group mailbox have access to the same range of messages. If a message is handled by any of these users, it no longer appears in anyone's mailbox.

SYSDBA Mailbox

All generated messages and requests, for which no user is defined, are sent to the SYSDBA mailbox, which is accessible to the owner SYSDBA.

User Mailbox

This kind of mailbox is available to only one user. Its name is identical to the user ID of the Entire Operations user. The user mailbox does not have to be defined or explicitly assigned to the user.

Clearing Mailboxes

Mailbox entries are cleared in accordance with the maximum length of time specified in the Entire Operations default settings for long-term logging. For further information, see the **Long Term Log** option described in *Default Setting (1)* in the *Administration* documentation.

XII

Log Information

63

Log Information

- Displaying Logged Information 588
- Output of Log Information 594
- Monitoring Entire Operations Activities 596

Entire Operations logs all important system events. You can view logs specific to an owner, network or job and specify selection criteria for the log report to be produced.

You can also use the monitoring function to observe all activities currently running in your Entire Operations environment.

Related Topic:

- [Defining Job-Specific Log Information](#) in the section *Job Maintenance*

Displaying Logged Information

Entire Operations logs all important system events. Additionally, more detailed user-defined logs are available at the job level. These must be specified for the job in the job definition.

➤ **To display all logged information**

- 1 Select the **Log Information** option from the Main Menu.

A **Log Display Selection** screen similar to the example below appears:

```

27.10.08                ***** Entire Operations *****                16:51:08
                        Log Display Selection
-----
Log Date ==> 27.10.08 Thru ==> 27.10.08    Max. Lines ==> _____
Log Time ==> 00:00:00 Thru ==> 19:00:00

Owner ==> _____                * Select   Blank All
Network ==> _____              * Select   Blank All

Run ==> 1_____ Thru ==> 99999

Job ==> _____                  * Select   Blank All

User ==> _____                 * Select   Blank all

Command => _____
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11--PF12---
      Help           End                                           Menu
    
```

You can specify the date range, owner, network, run number range, job name and user ID for which to display log information. The fields are explained in *Fields: Log Display Selection*.

- 2 Enter the selection criteria for the desired log information. Press ENTER.

A **System Log** screen similar to the example below appears:

```

Entire Operations System Log -----Columns 014 088
====>                                     Scroll====> PAGE
M User ID  Owner      Network   Job          Run Date  Time      Message
TASK 3    INCIDENT  I5033788A1  6585 20.01 00:00:33.8 Activation
TASK 3    INCIDENT  I5033788A1  6585 20.01 00:00:33.8 Awaiting S
TASK 1    EXAMPLE  E52-LOGRES  4556 20.01 00:00:33.8 ... Owner
TASK 1    EXAMPLE  E52-LOGRES  4556 20.01 00:00:33.8 ... Run 45
TASK 1    EXAMPLE  E52-LOGRES  4556 20.01 00:00:33.8 ... networ
TASK 3    INCIDENT  I5033788A1  6586 20.01 00:00:33.8 Activation
TASK 3    INCIDENT  I5033788A1  6586 20.01 00:00:33.8 No Network
TASK 1    EXAMPLE  E52-LOGRES  4556 20.01 00:00:33.8 List of ac
TASK 3    INCIDENT  I5033788A1  6586 20.01 00:00:33.8 Activation
TASK 1    EXAMPLE  E52-LOGRES  4556 20.01 00:00:33.8 Determined
TASK 1    EXAMPLE  E52-LOGRES  4556 20.01 00:00:33.8 ... Ob Jo
TASK 1    EXAMPLE  E52-LOGRES  4556 20.01 00:00:33.8 ... NV
TASK 3    INCIDENT  I5033788A1  6586 20.01 00:00:33.8 Awaiting S
TASK 1    EXAMPLE  E52-LOGRES  4556 20.01 00:00:33.8 ... JM
TASK 3    INCIDENT  I5033788A1  6587 20.01 00:00:33.9 Activation
TASK 3    INCIDENT  I5033788A1  6587 20.01 00:00:33.9 No Network
TASK 3    INCIDENT  I5033788A1  6587 20.01 00:00:33.9 Activation
TASK 3    INCIDENT  I5033788A1  6587 20.01 00:00:33.9 Awaiting S
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      End   Refre Rfind      Up    Down      Left  Right Curso

```

```

Entire Operations System Log -----Columns 054 128
====>                                     Scroll====> PAGE
M      Run Date  Time      Message
***** top of data *****
1060 17.11 00:00:00.0 Network to be started on 17.11 at 00:00
131 17.11 00:00:00.0 Network to be started on 17.11 at 00:00
1390 17.11 00:00:00.0 Network to be started on 17.11 at 00:00
1959 17.11 00:00:16.1 Activation in Progress
1959 17.11 00:00:16.3 Network to be started on 17.11 at 01:00
1959 17.11 00:00:16.3 Network activation performed
131 17.11 00:00:36.9 Job ready to submit
131 17.11 00:00:37.0 Job ready to submit
1390 17.11 00:00:37.0 Job ready to submit
1959 17.11 00:00:37.4 Prerequisite Check begins
1959 17.11 00:00:37.4 Start Time 17.11 01:00
492 17.11 00:00:42.3 Activation with Default Version
492 17.11 00:00:42.3 No Network Versions
492 17.11 00:00:42.3 Activation Network 22.11 13:00
492 17.11 00:00:42.3 Awaiting Symbol Table Activation
2611 17.11 00:00:43.2 Activation with Default Version
2611 17.11 00:00:43.2 No Network Versions
2611 17.11 00:00:43.3 Activation Network 22.11 06:20
2611 17.11 00:00:43.3 Awaiting Symbol Table Activation
131 17.11 00:00:58.3 Dummy Job terminated
131 17.11 00:00:58.3 Network: First job JOB1 started
131 17.11 00:00:58.3 Dummy Job (Schedule) terminated
1390 17.11 00:00:58.4 Dummy Job terminated
1390 17.11 00:00:58.4 Network: First job JOB1 started
131 17.11 00:01:16.4 Network ended ok
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      End  Refre Rfind      Up      Down      Left  Right Curso
  
```

The above screen is an example of a system log using the "display all" option for the current date (blank selection criteria).

- [Fields: Log Display Selection](#)
- [Column Headings: System Log](#)
- [Browse Commands: System Log](#)
- [PF Keys: System Log](#)

- Extended Log

Fields: Log Display Selection

The **Log Display Selection** screen provides the following input fields:

Field	Description
Log Date ... Thru	The date (or a start date) for which the log entries are selected. Default is the current date and 00:00:00 for time in the format HH:MM:SS. See also Date and Time Formats .
Log Time ... Thru	The time (or a start time) for which the log entries are selected. Default is the current date and the current time in the format HH:MM:SS. See also Date and Time Formats .
Max. Lines	The maximum number of lines in the log display. <i>Blank</i> (value 0) means: no limit (default). This setting can also be specified as a default value in the user profile: see the section <i>User Maintenance</i> in the <i>Administration</i> documentation.
Owner	Enter name of the owner whose log is to be displayed. Enter asterisk * and press ENTER to select owner from a list. Leave blank to select log of all owners.
Network	Enter name of the network whose log is to be displayed. Enter asterisk * and press ENTER to select network from a list. Leave blank to select log of all networks.
Run ... Thru	Enter run number range to be displayed. Can be used only if owner and network have been selected. Default: 1-9999 (all).
Job	Enter name of the job whose log is to be displayed. Enter asterisk * and press ENTER to select job from a list. Leave blank to select log of all jobs.
User	Enter the user ID for which the log is to be displayed. To display the log for the Monitor, enter the name of the Monitor task. Enter an asterisk * and press ENTER to define a group of users. A selection window does not open. To select the log of all users, either leave this field empty or enter only an asterisk *.

Column Headings: System Log

The **System Log** screen contains the following columns:

Column	Description
M	Overtyping the asterisk (*) in this field with any character to display more specialized information as defined for the job by the user. For more information see the subsection User-defined Log .
User ID	User ID as defined in host TP environment.
Owner	Owner name in Entire Operations.
Job	Entire Operations job name.
Network	Name of the job network.
Run	Job run number.
Date	Date of log entry. See also Date and Time Formats .
Time	Time of log entry. See also Date and Time Formats .
Message	Message text. Depending on the default display option set (see <i>User Definitions and Profile Settings</i> in the <i>Administration</i> documentation), the message text is prefixed with a message code (if one exists), for example: E0R2260 - Network activation performed.

Browse Commands: System Log

Use the following Editor commands to browse the Log:

Command	Description
BOTTOM	Scroll down to the last Log message at bottom of list.
FIND	Find a string. FIND 'TASK 1', for example, finds the next occurrence of the string TASK 1.
LEFT	Scroll left in the list. LEFT 5, for example, scrolls 5 columns to the left.
RIGHT	Scroll right in the list. RIGHT 5, for example, scrolls 5 columns to the right.
TOP	Scroll up to the first Log message at the top of the list.

PF Keys: System Log

Use the following PF keys to browse the Log:

PF Key	Name	Function
PF4	Refre	Update the screen with the most recent Log messages at bottom of data.
PF5	Rfind	Find the next occurrence of the string last entered with the FIND command.
PF7	Up	Scroll list toward top of data.
PF8	Down	Scroll list toward bottom of data.
PF10	Left	Scroll left (20 characters).

PF Key	Name	Function
PF11	Right	Scroll right (20 characters).

Extended Log

An asterisk (*) in the **M** column of any job means that a more specialized extended log is available according to the specifications made in the original job definition.

➤ To display the extended log

- 1 Type over the asterisk * in the **M** column of the selected job on the **System Log** screen with a non-blank character. Press ENTER.

The extended log appears.

- 2 Press PF3 (End) to return to the system log.

Types of Extended Log

The following types of extended log are available:

■ JCL log

If (in the Job Maintenance facility) you specified JCL logging, the JCL of the job run selected from the System Log screen is displayed in Editor format, browse mode. The JCL log source is dependent upon the operating system, in which the job has run.

■ SYSOUT log

If you specified the job SYSOUT to be logged after job termination, the SYSOUT of the job run selected from the System Log screen is displayed in Editor format, browse mode;

■ Operating system messages

If you specified operating system messages to be logged according to the criteria defined in the Job Maintenance facility, the messages of the job run selected from the System Log screen are displayed in Editor format, browse mode.

In later versions of NOP, more types of extended log may be added.

■ Active JCL modifications

If you specified to log the changes made to an active/pregenerated JCL using the corresponding option described in *Default Setting (2) (Administration documentation)*, the changes are visible in detail in the extended logging screen.

Output of Log Information

Log data can be written to the Natural standard output for printing or for transfer to other programs.

The batch utility NOPLP01P serves this purpose. It must be executed in a Natural batch job. The system files of Entire Operations must be correctly assigned.

Natural Program Call

```
LOGON SYSEOR
NOPLP01P <P-FROM> <P-TO> <P-LANGUAGE> <P-OWNER> <P-NETWORK> <P-DBENV> <P-RUN-FROM> ↵
<P-RUN-TO> <P-JOB> <P-ACTIVE-JCL> <P-SYSOUT> <P-CHGJCL> <P-EXTLOG>
FIN
```

Parameters

Parameter	Format	Use	
P-FROM	A14	Start date (with time as an option) Formats: YYYYMMDD YYYYMMDDHHIIS If time is omitted, the start is at the beginning of the day.	
P-TO	A14	End date (with time as an option) Formats: YYYYMMDD YYYYMMDDHHIIS If time is omitted, the end is at the end of the day.	
P-LANGUAGE	I04	Language code (optional)	
		1	English
		2	German
		If this parameter is omitted, the language setting from the Entire Operations defaults will be used.	
P-OWNER	A10	Owner range. Wildcard is allowed at the end. For all owners, use an asterisk '* '.	
P-NETWORK	A10	Network range. Wildcard is allowed at the end. For all networks, use an asterisk '* '.	
P-DBENV	A10	For future use. Use a hyphen ' - '.	
P-RUN-FROM	P13	Beginning of run number range. For all runs, use 1 here.	
P-RUN-TO	P13	End of run number range. For all runs, use 99999 here.	

Parameter	Format	Use	
P-JOB	A10	Job range. Wildcard is allowed at the end. For all jobs, use an asterisk ' * '.	
P-ACTJCL	A01	Y	Print the active JCL too.
		N	Do not print the active JCL.
P-SYSOUT	A01	Y	Print the logged SYSOUT too.
		N	Do not print the logged SYSOUT.
P-CHGJCL	A01	Y	Print the changes of the active JCL. In a batch report, the changes to the active JCL will be added.
		N	Do not print the changes of the active JCL (default).
P-EXTLOG	A01	Y	Print all extended logging (active JCL, SYSOUT, changes of the active JCL)
		N	Do not print any extended logging.

Example of a Call

```
LOGON SYSEOR
NOPLP01P 20081027 20081028120000 1 EXAMPLE NET* - 1 99999 * Y Y N
FIN
```

Outputs all log data in English from 27/10/2008 00:00:00 to 28/10/2008 12:00:00, for owner EXAMPLE, network names starting with NET, all run numbers, all jobs of these networks, with active JCL, with logged SYSOUT, without changes to the active JCL, without using the global 'extended logging' switch.

Example of an Output

1===== Entire Operations Log - 27.10.08 00:00:00 thru 28.10.08 12:00:00 ===== Page: 6								
Date	Time	User-ID	Owner	Network	Run	Job	Code	Message
27.10.08	10:22:36	TASK 1	UKSJU	SCHEDDEP			7715	Network SCHEDDEP did not run on 27.10.08
27.10.08	10:22:36	TASK 1	UKSJU	SCHTEST98			7715	Network SCHTEST98 did not run on 27.10.08
27.10.08	10:22:36	TASK 1	UKSJU	TESTEXP			2710	Calendar WORKDAYS undefined for 2007
27.10.08	10:22:36	TASK 1	USW	V-AT-VWLS			2710	Calendar K-USW undefined for 2007

27.10.08	10:22:36	TASK 1					7065	Schedules of 26 Networks extracted
27.10.08	10:22:38	TASK 1	EXAMPLE	E01-CONTI	526		7370	Symbol Table EX-ST-COMN activated
27.10.08	10:22:38	TASK 1	EXAMPLE	E10-PAR-01	151		7370	Symbol Table EX-ST-COMN activated
27.10.08	10:22:38	TASK 1	EXAMPLE	E60-FLOW	4767		7370	Symbol Table EXAM-ST1 activated
27.10.08	10:22:39	TASK 1	GFR	BS2000	714		7370	Symbol Table GFR-ST1 activated
27.10.08	10:22:39	TASK 1	GFR	BS2000	714		7725	Awaiting Symbol Prompting
27.10.08	10:22:39	TASK 1	GFR	BS2000	714		7725	Message Sending: No Recipient defined
27.10.08	10:22:39	TASK 1	GFR	BS2000	714		2060	Symbol Prompt Request sent to SYSDBA
27.10.08	10:22:40	TASK 1	EXAMPLE	E01-CONTI	526	E01-J01	1990	Time Frame Setting
27.10.08	10:22:40	TASK 1	EXAMPLE	E01-CONTI	526	E01-J02	1990	Time Frame Setting
27.10.08	10:22:40	TASK 1	EXAMPLE	E01-CONTI	526	E01-J03	1990	Time Frame Setting
27.10.08	10:22:40	TASK 1	EXAMPLE	E01-CONTI	526	E01-J04	1990	Time Frame Setting
27.10.08	10:22:40	TASK 1	EXAMPLE	E01-CONTI	526	E01-J05	1990	Time Frame Setting
27.10.08	10:22:40	TASK 1	EXAMPLE	E01-CONTI	526	E01-J06	1990	Time Frame Setting
27.10.08	10:22:40	TASK 1	EXAMPLE	E01-CONTI	526		2110	Network activated on 24.02 at 10:22

Monitoring Entire Operations Activities

Activity monitoring is a non-conversational display of Entire Operations events on a special terminal, for example in the computer center. Activity monitoring can be started on this terminal and can be controlled from any other terminal.

Activity monitoring is used to permanently display Entire Operations events that currently occur in your Entire Operations environment.

The following applies to the monitoring feature:

- The activity monitoring facility receives its data from the Entire Operations log file.

- The activity monitoring facility only displays the most important messages. Error messages and other very important messages appear highlighted.
- The default setting displays events starting from the current time minus one hour. As the display is updated, it scrolls automatically. The start time and display refresh interval can be modified via the **Control of Activity Monitoring** option of the **Special Functions** system administration services described in the *Administration* documentation.
 - [Viewing Entire Operations Activities](#)
 - [Fields and Columns: Activity Display](#)

Viewing Entire Operations Activities

➤ To start activity monitoring and display activities

- 1 At a Natural command prompt, enter the following command and press ENTER:

```
LOGON library-name
```

where *library-name* is the name of your Entire Operations library, usually SYSEOR.

When you are logged on to the specified library, enter the following at the command prompt:

```
ACTIVITY
```

Or:

In an Entire Operations command line, enter the following direct command and press ENTER:

```
ACTIVITY
```

Activity monitoring is started, and an **Entire Operations Activity Display** window similar to the example below opens:

```

16-09-27          ***** Entire Operations *****          17:51:37
Owner EXAMPLE          Main Menu          User ID NATQA5
-----
Main Menu          DC Solutions

1 Networ +-----+ t
2 Active !
3 Schedu !          Entire Operations Activity Display !
4 Calend !
5 Log In ! Query Interval (in minutes) ==> 60___ !
6 Symbol !
7 System ! The query interval is the time span until the !
8 Report ! next interruption query by the activity display. !
9 Cross ! In between, it can only be interrupted from !
10 Import ! another terminal running Entire Operations. !
11 Help !
      ! PF3 End !
      ! !
      ! !
Command => +-----+ ___

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help          End          Owner Mail
    
```

You can modify the **Query Interval** (update interval) as required.

- 2 Press ENTER.

An **Activity Display** screen similar to the example below appears:

```

05.11.08          *** Entire Operations ***          15:24:17
Date 05.11.08          Activity Display          Monitor active 15:24:12
from 14:37:32          Display Interval 20   sec.   Next Interrupt 16:23:14
-----
Network   Job           Run JobId Time  Message
UKSJUTEST1 TEST1JOB1      2  5094 14:37 Monitor Error 936 in EJA-EV-N Line 553
UKSJUTEST1 TEST1JOB1      2  5094 14:37 Message Sending: No Recipient defined
UKSJUTEST1 TEST1JOB1      2  5094 14:37 Monitor Task 1 Restart by NOPMET-P
UKSJUTEST1 TEST1JOB1      2  5094 14:37 Monitor Error 936 in EJA-EV-N Line 553
UKSJUTEST1 TEST1JOB1      2  5094 14:37 Message Sending: No Recipient defined
UKSJUTEST1 TEST1JOB1      2  5094 14:37 Monitor Task 1 Restart by NOPMET-P
QAN02z/OS-M          2           14:38 Message Sending: No Recipient defined
E40-REC-01          1032  2257 14:38 Message Sending: No Recipient defined
E40-REC-01          1033  2440 14:38 Message Sending: No Recipient defined
E50-USRT            753           14:38 Message Sending: No Recipient defined
E50-USRT            754           14:38 Message Sending: No Recipient defined
E50-USRT            755           14:38 Message Sending: No Recipient defined
E50-USRT            756           14:38 Message Sending: No Recipient defined
E50-USRT            757           14:38 Message Sending: No Recipient defined
E50-USRT            758           14:38 Message Sending: No Recipient defined
SUB-1                2           14:38 Message Sending: No Recipient defined
SUB-1                3           14:38 Message Sending: No Recipient defined
***** m o r e *****

```

The **Display Interval** (query interval) is equivalent to the Monitor and cannot be changed.

The fields and columns contained on the screen are described in [Fields and Columns: Activity Display](#).

- When the query interval has expired and the **Next Interrupt** time is reached, a window similar to the example below prompts you to restart the **Activity Display** screen by entering Y, or end monitoring by entering N. Press ENTER.

```

11.03.15          ***** Entire Operations *****          11:36:54
Date 11.03.15          Activity Display          Monitor active 11:36:45
from 11:20:17          Display Interval 20   sec.   Next Interrupt 11:37:14
-----
Network   Job           Run JobId Time  Message
11:20 EOR2046 - Natural Message Cache disabl
11:23 EOR2045 - Natural Message Cache enable
11:23 DCAGD1-P 1380 SYSNOM / MMO NSC---L Rc:
11:23 NAT0807 - Logon to this library is not
11:23 LG-LEV-P 790 - program stack
11:23 . 2:DCAGD1-P/1450 1:MENU EOR/2800
11:24 EOR2045 - Natural Message Cache enable
N1459C    JOB-1          17202    11:26 Dummy Job terminated
N1459A    JOB-1          17223    11:30 Activation in Progress
N1459C    JOB-1          +-----+
N1459A    JOB-1          |                                     |
N1459C    JOB-1          | Activity Display has expired         | 1, avail. 0 of 1
N1459A    JOB-1          |                                     | at 11:58
N2009A    J001          | Restart ==> Y (Y,N)                 | - Invalid User ID
N2009A    J001          |                                     | h
N2009A    J001          +-----+ L Generation Error
N2009A    J001          63 8465 11:30 EOR7182 - JCL Load - Invalid User ID
***** m o r e *****

```

Fields and Columns: Activity Display

The fields and columns of the **Activity Display** screen are described in the following table:

Field/Column	Meaning
from	Date and time of the first message on the screen. See also <i>Date and Time Formats</i> .
Display Interval	The time after which the display is automatically refreshed.
Monitor active	The time the Monitor was last active.
Next Interrupt	The time when the Activity Display will be interrupted to prompt you for continuation. This is the end of the query interval specified in the Entire Operations Activity Display window .
Owner	Owner of the active network.
Network	Name of the active network.
Job	Name of the active job.
Run	Run number as assigned to the active network by Entire Operations.
JobId	Job identifier as assigned by the operating system or by the job entry subsystem.
Time	Last action or check time for the job.

Field/Column	Meaning
Message	Last message issued for the job by Entire Operations. For a list of possible messages, see <i>Messages in Active Job Display</i> in the <i>Messages and Codes</i> documentation.

XIII

Symbol Table and Symbol Maintenance

Purpose and Use of Symbol Tables and Symbols

Adding and Maintaining Symbol Tables

Finding Symbol Tables Associated with Jobs and Networks

Defining Symbols and Symbol Values

Symbol Replacement

Functions for Symbol Replacement

64 Purpose and Use of Symbol Tables and Symbols

- Symbol Tables 606
- Symbols 607
- Subnetworks and Recovery Jobs 609
- User Exits for User-specific Symbol Maintenance Tasks 609

Related Topic:

- *Symbol Tables and Symbols* in the *Concepts and Facilities* documentation

Symbol Tables

A symbol table is used to maintain common parameter definitions that apply to all jobs of a network. A symbol table contains a list of defined symbols with values that can be used for parameter substitution during JCL generation.

You can maintain symbol tables by using either Entire Operations maintenance functions or user-written programs, user exits and application programming interfaces (APIs) supplied by Entire Operations. This allows modification of symbol tables and symbol values during job network processing.

Symbol tables are associated with owners. Each owner can have several symbol tables. You can only maintain the symbol tables that belong to your owner.

The symbol table to be referenced by a job and/or network must be specified in the job or network master definition or in the active queue for a single job run. Exceptions are [global symbol tables](#).

Each network activation initiates its own active copy ([active symbol table](#)) of the linked symbol table(s). This allows you to schedule networks with different parameter sets, even a long time in advance. Any occurrence of a symbol name in the JCL or in a script is replaced by its current value.

This section covers the following topics:

- [Global Symbol Tables](#)
- [Active Symbol Tables](#)
- [Reserved Symbol Table Name](#)
- [Symbol Table Versions and Version Usage](#)

Global Symbol Tables

A master symbol table defined by the owner SYSDBA is considered a global symbol table because it can be referenced by multiple jobs and networks without explicit specifications in the job and network definitions. A global symbol table with the name A is supplied for the owner SYSDBA by default. It can be accessed and referenced by all owners.

All other owners can also define a global symbol table with the name A for their networks. Their global symbol table A then takes priority over the global symbol table A defined by SYSDBA. See also [Symbol Table Types and Symbol Search Order](#).

Active Symbol Tables

During job or network activation, an active copy of each symbol table used by a job or network is created by copying it to the active database. This is then called the active symbol table. Symbols to be replaced are taken from the active symbol table. This enables different values of the same symbols to be generated for different runs of the network or job.

To identify an active symbol table uniquely, both the network name and run number are required.

For scheduled network activation, you can select the time to activate the symbol table in the network definition. This can be either of the following:

- Directly after the schedule extraction; this allows enough time for manual symbol replacement.
- During the network activation. In this case, however, manual symbol replacement is not possible.

Active symbols are maintained analogously to master symbols. Therefore, only the maintenance of master symbol tables and symbols is described in the section [Adding and Maintaining Symbol Tables](#).



Note: All modifications of active symbols apply to the current run only.

Reserved Symbol Table Name

The symbol table name with the prefix =EOR= is reserved for internal control purposes. This prefix must not be used for any user-defined table names.

Symbol Table Versions and Version Usage

You can maintain several versions of a master symbol table as described in *Symbol Table Versioning* in the *Concepts and Facilities* documentation. Exceptions are **global symbol tables** for which versioning is not allowed.

If you have several symbol table versions, you can define a validity period for each version. See the section [Maintaining the Usage of Symbol Table Versions](#).

Symbols

A symbol is a variable definition that contains the format, prompting type, prompting text (optional) and the value(s) of the symbol.

Symbols can be defined by using Entire Operations maintenance functions, or they can be set and modified by any program, which invokes the Entire Operations symbol API **NOPUSY6N** (see the section *API Routines*) to set, reset, or modify a symbol.

For detailed information on defining symbols in JCL (including text objects) or Natural macro sources, see [Editing Master JCL and Natural Sources](#) and [Inserting Text Objects into the JCL](#) in the section *Job Maintenance*.

This section covers the following topics:

- [Global Symbols](#)
- [Predefined Symbols](#)
- [Symbols for Replacement](#)
- [Symbol Prompting](#)

Global Symbols

A master symbol contained in a global symbol table is considered a global symbol because it can be used by multiple jobs and networks without extra definitions for each job and/or network that references the symbol. See also [Global Symbol Tables](#).

Predefined Symbols

Predefined symbols provided by Entire Operations are described in the section [Predefined Symbols](#).

Symbols for Replacement

Symbols are usually replaced during the activation of a job network or job, that is, during the loading of the active JCL to the active database. In certain cases, symbol replacement can even be used in file names, messages texts, etc.

A symbol to be replaced is identified by an [escape character](#) defined in the Entire Operations default and network settings, or used as a prefix in JCL and Natural macro sources. You specify the symbols in JCL or macro sources as described in [Defining Escape Characters](#).

If a symbol is not found in the symbol table specified for a job or network, the symbol is searched for in the [global symbol table\(s\)](#). The first match is used to replace the symbol. A symbol is not replaced if it cannot be found in any table; an appropriate error message is then issued instead. See also [Symbol Table Types and Symbol Search Order](#).

For further information, see the separate chapter [Symbol Replacement](#).

Symbol Prompting

You can define whether a symbol is prompted for symbol modification during or before job network activation.

- During a manual network or job activation, the prompting is performed immediately. See the section [Symbol Prompting during Network or Job Activation](#).
- For scheduled (automatic) activations, prompting requests are sent to a mailbox linked to the network. The network activation does not continue until the prompting is performed by a user. See also [Specifying Recipients for Network Messages](#) in the section *Network Maintenance*.

Subnetworks and Recovery Jobs

Subnetworks and **recovery jobs** defined for a network require predefined symbols in order to pass information to the jobs of the network. These predefined symbols have names that begin with P-C-. For detailed information, see [Predefined Symbols for Subnetworks and Recovery Jobs](#).

User Exits for User-specific Symbol Maintenance Tasks

You can use user exit routines that perform user-specific symbol maintenance tasks and symbol replace functions. For detailed information, refer to the following sections:

User Exits:

- [User Exits for Validation Checks of Symbol Values](#)
- [User Exits for Symbol Functions](#)

Global User Exits in the Administration documentation:

- *Global Symbol Modification Exit*
- *Global Symbol Not Found Exit*

65 Adding and Maintaining Symbol Tables

- Listing Master Symbol Tables 612
- Copying Master Symbol Tables - Cloning Symbol Table Versions 614
- Adding a Symbol Table Master 618
- Displaying and Modifying a Master or Active Symbol Table 624
- Maintaining the Usage of Symbol Table Versions 627
- Deleting a Master Symbol Table 631

- [Special PF Keys: Master Symbol Tables](#)

Columns: Master Symbol Tables

The columns contained in the [Master Symbol Tables window](#) are described in the following table.

Column	Description
Cmd	One-character line command input field. For possible values, see Line Commands: Master Symbol Tables .
Owner	Owner of the symbol table. The selection field of this column denotes the owner for whom symbol tables are displayed in the column Symbol Table . This field is preset with the owner under which you are currently working. You can enter any owner to which your user ID belongs. Enter an asterisk (*) and press ENTER to display a selection list of owners.
Symbol Table	Name of the symbol table. In the selection field of this column you can enter selection criteria for the symbol tables to be listed for the owner. For example, enter DE* and press ENTER to list all symbol tables beginning with DE. This field appears at first with an asterisk (*) and displays all of the owner's symbol tables.

Line Commands: Master Symbol Tables

Use the following line commands to perform the described functions on any symbol table listed in the [Master Symbol Tables window](#):

Line Command	Description
C	Copy master symbol tables, cloning symbol table versions.
D	Delete a master symbol table. If several symbol table versions exist, the symbol table version selection window opens. Mark the version you want to delete.
L	List and modify symbols in a symbol table. If several symbol table versions exist, a Symbol Table Version Selection window opens first. Mark the required version and press ENTER.
U	Maintain the usage of symbol table versions .
W	List jobs which use a symbol table.

Special PF Keys: Master Symbol Tables

The [Master Symbol Tables window](#) provides the following special PF key:

PF Key	Name	Function
PF2	Add	Add a master symbol table: see Adding a Symbol Table Master .

The following sections describe the functions you can perform on master symbol tables using available line commands and PF keys.

Copying Master Symbol Tables - Cloning Symbol Table Versions

When defining a new master symbol table, you can select an existing table from the **Master Symbol Tables** window as a model for the new table.

➤ To copy a master symbol table, or to clone a symbol table version

- 1 In the [Master Symbol Tables](#) window, type C in the line command input field of the table to be copied. Press ENTER.

The following window opens:

```

16-09-17          ***** Entire Operations *****          17:51:02
Owner SAGTEST          Main Menu          User ID NATQA5
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
!
!          Symbol Table Master Definition Copy          !
!  Identical Symbol detected - Please mark the desired Value  !
!  M  Owner      SymTab      Symbol          modified by      !
!  Copy from:                                           !
!  SAGTEST      EXAM-TABLE CLASS          NATQA5  16-05-12 14:18 !
!  Version (unnamed)                                     !
!  Exit SYCRN1-N Library SYSEOR                          !
!  _ Value *** 3 multiple values ***          Multiple Y          !
!  Copy to:                                           !
!  SAGTEST      EXA-SYMBOL CLASS          NATQA5  16-05-11 11:34 !
!  Version (unnamed)                                     !
!  Exit SYCRN1-N Library SYSEOR                          !
!  _ Value *** 1 multiple values ***          Multiple Y          !
!
! PF3 End
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
!          Help Add  End      Save  Up  Down  Menu          !
+-----+-----+-----+-----+-----+-----+-----+-----+
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help          End                                Owner Mail

```

The name of the owner, symbol table and symbol table version to copy from already appear in the **From** fields. (If necessary, you can also enter a different owner, symbol table and symbol table version in the **From** fields).

- 2 To open selection windows for owners, symbol tables and symbol table versions, use an asterisk (*) as a wildcard.
- 3 Enter the name of the owner, symbol table and eventually the symbol table version to copy to in the **To** fields.

Notes:

- If an alternate symbol table is defined in the **Entire Operations Defaults** (see the *Administration* documentation), you can copy a symbol table from there by setting the **Alt. File** field to Y.
- You can only copy a symbol table to another owner if you are authorized to access networks belonging to that owner.
- The current owner is used as the target symbol table owner. If necessary, use the direct command SET OWNER (see SET in the **Direct Commands** documentation) to change the current owner before a symbol table copy action. The target symbol table owner field is write protected due to profile and security reasons.

- You can copy a symbol table to an existing table and merge the two tables.
- 4 Press ENTER to copy the table to a new table or to combine it with an existing table.
- If you have simply copied to a new table, this new table is now listed in the **Master Symbol Tables** window.
 - If you have combined two tables, this is performed in the following manner:
 - All symbols and their values are copied successively from the source table to the target table.
 - If an identical symbol with identical value(s) exists in both the source and the target table, then this symbol is not copied.
 - If, however, the values of this identical symbol differ, the following window opens:

```

18.12.13      !                               ! ns *****      11:04:40
Owner      ←
+-----+
Networ + !                                     ←
!
←
+-----+
!                                     ←
!                                     ←
!           Symbol Table Master Definition Copy      ←
!                                     ←
!   Identical Symbol detected - Please mark the desired Value  ←
!                                     ←
!   M   Owner      SymTab      Symbol      modified by      ←
!                                     ←
!   Copy from:                                     ←
!                                     ←
!   SN      A-1      NUM-1      SN      18.12.13 ←
11:04 !
!           Version v10.05                                     ←
!                                     ←
!           Exit      Library                                     ←
!                                     ←
!   _ Value 7777.567890      Multiple N      ←
!                                     ←
!   Copy to:                                     ←
!                                     ←
!   SN      A-1      NUM-1      SN      18.12.13 ←
11:04 !
!           Version v12.44                                     ←
!                                     ←
!           Exit      Library                                     ←
!                                     ←
!   _ Value 8888.567890      Multiple N      ←
!                                     ←
!                                     ←
!   PF3 End                                     ←
!
←
+-----+
Enter-P ! ←
+-----+
H +-----+

```

The input field and columns are explained in *Fields and Columns: Different Symbol Values*.

- 5 Use any character to mark the value to be retained in the target table after copying:

In the above example, the symbol STEPLIB was found in the source as well as in the target table.

- If you mark Value NATOP.V110, then the symbol STEPLIB in the table DEMO-2 will have Value NATOP.V110 after copying. Value NATOP.V210 will be overwritten.
- If you mark Value NATOP.V210, the symbol STEPLIB in the table DEMO-2 retains this value.



Note: Every time a symbol value in the target table is overwritten by the pertinent value in the source table, a corresponding message is written to the log. When the whole copying process has been completed, a message is also written to the log.

This section covers the following topic:

Fields and Columns: Different Symbol Values

Field/Column	Description	
M	Input field for value selection.	
Value	Symbol value. Enter any character to select the symbol value to be retained.	
Owner	Owner of the source and of the target symbol table.	
SymTab	Names of source and target symbol tables.	
Symbol	Name of the symbol which exists in both symbol tables.	
modified by	User ID of the person who last modified the table as well as date and time of last modification.	
Multiple	Possible entries:	
	Y	Multiple symbol values exist.
	N	Multiple symbol values do not exist.

Adding a Symbol Table Master

You can add a symbol table master by either copying an existing table (see [Copying Master Symbol Tables - Cloning Symbol Table Versions](#)) or creating a new symbol table with PF5 (Add). This requires that you define at least one symbol master along with the table.

This section provides instructions for using PF5 (Add).

> To add a master symbol table

- 1 Press PF2 (Add) in the [Master Symbol Tables window](#).

A **Master Symbol Addition** window similar to the example below opens:

```

19.12.13          ***** Entire Operations *****          09:24:15
Owner   SN          Master Symbol Table A-1          Version v12.44
+-----+-----+-----+-----+-----+-----+-----+-----+
!
!               Master Symbol Addition               !
!
! Symbol Table ==> A-1_____          Owner SN          ! .05.13 15:25
!   Version ==> v12.44_____          ! .03.12 13:16
! Symbol Name ==> SHORT1_____          Network          ! .12.13 11:04
!   Format ==> _                      Run                ! .03.12 15:03
!   Value ==> short-value _____          ! .05.13 16:44
!
! _____          !
! Be careful with insertions or deletions.          !
! The 2 lines will be concatenated to one symbol value.  !
!   Prompting ==> _                      Update Mode ==> _  !
!   Prompt Text ==> _____          !
! _____          !
! _____          !
! _____          !
! _____          ! *****
! Enter-PF1---PF2---PF3-----PF5----PF8----PF9---PF10---PF11--  !
!   Help Add  End    Save  RgChk Mult Copy  Exit          ! F11--PF12---
+-----+-----+-----+-----+-----+-----+-----+-----+

```

- 2 Enter the name of a new master symbol table and define the first symbol for it. The input fields are described in *Fields: Master and Active Symbols*. The special PF keys available are explained in *Special PF Keys: Master Symbol Addition*.
- 3 If you wish to add more symbols to the new table, press PF2 (Add) from the **Master Symbol Addition** window. This saves the first definition and clears the window.

You can now add another symbol. You can add any number of symbols to a master symbol table.

- 4 Press PF5 (Save) to save the new master symbol table.
- 5 Press PF3 (End) to close the window and return to the **Master Symbol Tables** window. The new master symbol table appears in the list.

You can add any number of master symbol tables for your owner.

This section covers the following topics:

- [Special PF Keys: Master Symbol Addition](#)
- [Fields: Master and Active Symbols](#)

▪ Example

Special PF Keys: Master Symbol Addition

You can perform the following functions from the **Master Symbol Addition** window using these PF keys:

PF Key	Name	Function
PF8	RgChk	<p>Numeric value check.</p> <p>The value range check is only performed for values of the format N. In this case, a special window appears here.</p> <p>This PF key is available for master definitions only.</p>
PF9	Mult	<p>Assign multiple values for one symbol.</p> <p>These can be used in subsequent replacements of the same symbol. A special window opens.</p>
PF10	Copy	<p>Copy a complete master symbol table, even from the alternate Entire Operations data file.</p> <p>This PF key is available for master definitions only.</p>
PF11	Exit	<p>Define and edit a validity checking user exit for this symbol: see <i>User Exit for Validating Symbol Values</i>.</p> <p>This PF key is available for master definitions only.</p>

Fields: Master and Active Symbols

The fields in the **Master Symbol** or **Active Symbol** window and subordinate windows are described in the following table.

The fields are either write-protected or available for input depending on whether you view or modify the symbols. The table indicates when a field does not apply to active symbols.

Master Symbol - Main Window:	
Symbol Table	<p>Name of the master symbol table.</p> <p>Symbol tables contain the symbols (variables) required for dynamic JCL generation.</p> <p>Symbols can be maintained manually or modified by programs.</p>
Version	Optional version of the symbol table.
Symbol Name	Name of the symbol to be used as a variable.
Owner	Owner of the symbol table.
Network	Current job network (active symbol tables only).

Format	Format of the symbol (variable). Possible values:	
	A	Alphanumeric (including special characters)
	D	Date
	H	Alphanumeric; hidden <ul style="list-style-type: none"> ■ The symbol value will not be visible in the Value field. ■ In lists and logs, the symbol value will be displayed as <code>*** hidden ***</code>.
	L	Alphanumeric; conversion to lower case
	N	Numeric See Value for the numeric field format.
	U	Alphanumeric; conversion to upper case
Run	Current job run (active symbol tables only).	
Value	<p>Value of the symbol to be substituted. To enter a fixed length or an empty string, include the value in quotes. Use two apostrophes to get a quote within a string.</p> <p>Example: 'Fixed Length '</p> <p>Note:</p> <ol style="list-style-type: none"> 1. If a value is longer than 40 characters, continue on the second Value line. The maximum value is 80 characters. Be careful with insertions and deletions. The two lines are concatenated to one symbol value. 2. The maximum length for a numeric value is N10.4 (10 digits before the decimal point and 4 digits after the decimal point). A decimal point can appear at any position. 3. A date must use the format YYYYMMDD. 4. The value of a hidden symbol (format H) can be entered, but it will be invisible. <p>See also Defining Multiple Symbol Values and Specifying a Range Check for Numeric Symbol Values.</p>	
Prompting	Specifies whether the user is to be prompted for this symbol during manual activation of the job network. See also Symbol Prompting during Network or Job Activation .	
	Possible values:	
	A	Prompt for symbol at each activation.
	E	Prompt only if no value is specified in the table.
	N	Never prompt for symbol.

Modification Mode	M	Write back modified active symbol to the master table as well. Note: The symbol is also written back, if a respective global setting is available. See <i>Entire Operations Defaults</i> in the section <i>System Administrator Services</i> of the <i>Entire Operations Administration Documentation</i> .
	else	No special action.
Prompt Text	Optional text to be displayed when symbol prompting is performed, to help the user specify the correct value. See also Symbol Prompting during Network or Job Activation .	
Multiple Symbol Values Window:		
Values ... to ...	Assign multiple values to a symbol. These can be used in subsequent replacements of the same symbol. See also Defining Multiple Symbol Values and Specifying a Range Check for Numeric Symbol Values . You can enter up to 100 values here. The maximum (internal) value length is 80. These values are used for subsequent substitutions. Scroll with PF7 (Up) and PF8 (Down). Note: If an Adabas version older than Version 8 is used, the absolute size of these multiple-value symbols is limited by the block size of the Adabas DATA.	
Numeric Range Check Window:		
Not applicable to active symbols Used to define a range of numbers to be checked for valid input of either numeric symbol values. See also Specifying a Range Check for Numeric Symbol Values .		
Value from	Start (from) and end (to) number of the range of numbers to be checked. Valid input values: 1 to 999.	
Value to		
Definition of Symbol Check Exit Window:		
See also User Exit for Validating Symbol Values .		
Library	Name of the library in which the user exit resides You can use an asterisk (*) as a wildcard to select a library from a list. Note: You cannot select the libraries SYSLIB and SYSLIBS. They are reserved for internal use.	
User Exit	Name of the user exit If the name of a valid library is entered in Library , you can use an asterisk (*) as a wildcard to select a user exit from a list.	

Example

The following example illustrates the use of the symbol FILE-1 in the master symbol table DEMO:

```

19.12.13          ***** Entire Operations *****          09:27:53
Owner   SN              Master Symbol Table A-1              Version v12.44
+-----+-----+-----+-----+-----+-----+-----+-----+
!
!               Master Symbol Modification                   !
!
! Symbol Table ==> A-1_____ Owner SN                      ! .05.13 15:25
!   Version   ==> v12.44_____                             ! .12.13 09:27
! Symbol Name ==> FILE-1_____ Network                    ! .12.13 11:04
!   Format    ==> A           Run                          ! .12.13 09:26
!   Value     ==> NOP.DEMO.SRCE_____                     ! .03.12 15:03
!                                                     _____ ! .05.13 16:44
! Be careful with insertions or deletions.                 !
! The 2 lines will be concatenated to one symbol value.    !
!   Prompting ==> E           Update Mode ==> _            !
!   Prompt Text ==> _____                             !
! _____                                               !
! _____                                               !
! _____                                               !
! _____                                               !
! _____                                               ! *****
! Enter-PF1---PF2---PF3-----PF5-----PF8-----PF9---PF10---PF11-- !
!   Help Add End Save RgChk Mult Copy Exit F11--PF12--- !
+-----+-----+-----+-----+-----+-----+-----+-----+

```

For each occurrence of the symbol FILE-1 in the JCL of the job for which symbol table A-1 with version v12.44 is specified, the value NOP.DEMO.SRCE is substituted. For example, the JCL line:

```
//XYZ DD DSN=@FILE-1,DISP=SHR
```

as defined in the original JCL, becomes:

```
//XYZ DD DSN=NOP.DEMO.SRCE,DISP=SHR
```

at job activation time.



Note: Symbols preceded by the submission escape character in the JCL are replaced by their current values when the job is submitted.

The user can modify the resulting JCL for a single job run as described in [Editing Active JCL](#) in the section [Active Job Networks](#).

When the job network which uses this symbol table is activated manually, the user is prompted for this symbol. The text in the **Prompt Text** field is displayed during symbol prompting to help

the user specify the correct value. The user can leave the specified value or modify it for the network run.

Displaying and Modifying a Master or Active Symbol Table

Modifying a master symbol table or an active symbol table consists of adding, modifying or deleting symbols and their current values.

➤ To display or modify a master or an active symbol table

- 1 In the **Master Symbol Table** window, type L (List Symbols) in the **Cmd** input field next to the required symbol table, and press ENTER.

Or:

In a **Master Job Definition** or **Job Definition (active)** window, press PF7 (Symb).

A **Usable Symbol Tables** window opens.

In the **Cmd** column, type L next to the required symbol table, and press ENTER.

- 2 A **Master Symbol Table** or **Active Symbol Table** screen appears for the selected symbol table:

```

19.12.13          ***** Entire Operations *****          10:21:50
Owner   SN              Master Symbol Table A-1              Version v12.44
Network                                         Run
-----
C Symbol          F P Value                                modified by
*-----
_ ABCD           A E dcba                                     SN      24.05.13 15:25
_ FILE-1        A E NOP.DEMO.SRCE                                     SN      19.12.13 09:27
_ MULT-01       A E *** 5 multiple values ***                       SN      19.12.13 09:35
_ NUM-1         N N 888.567890                                       SN      19.12.13 10:19
_ SHORT1       A E short-value                                       SN      19.12.13 09:26
_ TEST 1       A N parameter missing: SYMBOL                       SN      09.03.12 15:03
_ Z1           A E AAAAAA                                           SN      23.05.13 16:44

***** Bottom of Data *****
C Copy   D Delete   L List   M Modify
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Add   End   Copy Save           Up   Down Print

```

The screen lists the symbols defined for the table by name, format, prompt value and current symbol value. Information is provided on the right concerning the user and date and time when the symbol was last modified.

You can add, modify and delete a symbol from the symbol table.

This section covers the following topics:

- [Columns: Symbol Table Master](#)
- [Special PF Keys: Symbol Table Master](#)

Columns: Symbol Table Master

The columns on the [Master Symbol Table screen](#) are explained in the following table:

Column	Description
C	One-character line command input field. Possible values:
	C Copy symbol.
	D Delete symbol.
	L List symbol.
M Modify symbol.	
Symbol	Name of the symbol used as a variable.

Column	Description
F	Format of the variable. Possible values:
	A Alphanumeric (including special characters)
	D Date
	L Alphanumeric; conversion to lower-case letters
	N Numeric
	U Alphanumeric; conversion to upper-case letters
P	Prompting. The user is prompted for this symbol during manual activation of the job network. Possible values:
	A Prompted for symbol at each activation.
	E Prompted only if no value is specified in table.
	N Never prompted for symbol.
Value	Current value of the variable, i.e., the string or number substituted in dynamically generated JCL. Possible entries:
	*** empty *** No value is defined for the variable.
	*** <i>n</i> multiple values *** One or more values (<i>n</i>) are defined for the variable. Use the line command M and then PF9 (Mult) to view or change the values.
Modified by	User who last modified the symbol followed by date and time of last modification.

Special PF Keys: Symbol Table Master

The [Master Symbol Table screen](#) provides the following special PF key:

PF Key	Name	Function
PF9	Print	Print symbol table with all symbols on the printer assigned to PRINTER1. Exception for BS2000: An additional window opens in which you are asked to enter the printer name. If you wish to print via FORMS because RSO is not installed, enter the following in the field Print Device : <i>F=form-name</i> .

Maintaining the Usage of Symbol Table Versions

You can determine a period (from/to date range) during which a particular symbol table version is used as the current symbol table for all scheduled job activations.

The **Version Usage** function is used to define the maintenance of date ranges, in which symbol table versions are to be used for scheduled activations.



Note: You cannot use the **Version Usage** function to add a symbol table version. You can define a symbol table version when you create or copy a symbol table master.

This section covers the following topics:

- [Listing, Defining and Deleting Version Usage](#)
- [Fields and Columns: Version Usage Definition](#)
- [Commands: Version Usage Definition](#)

Related Topic:

- *Object Versioning* in the *Concepts and Facilities* documentation

Listing, Defining and Deleting Version Usage

This section provides instructions for listing, defining and deleting version usage definitions for symbol table versions.

➤ To list the version usage defined for a symbol table

- In the **Master Symbol Tables** window, enter the line command U (Version Usage) next to required symbol table, and press ENTER.

A **Symbol Table Version Usage** screen similar to the example below appears:

```

18.12.13          ***** Entire Operations *****          10:45:16
                   Symbol Table Version Usage
Owner SN          Symbol Table A-1
-----
Cmd from          to          Version      Description
-----
_ 18.12.13        31.12.13    v10.05      for the rest of 2013
_ 01.01.14        >>>>>>>>    v12.44      starting 2014

***** Bottom of Data *****
D Delete  M Modify
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Add   End                Up    Down      Left  Right
    
```

The fields and columns available on the screen are explained in *Fields and Columns: Version Usage Definition*. The commands available are explained in *Commands: Version Usage Definition*.

➤ **To add or modify a version usage definition**

- 1 On the **Symbol Table Version Usage** screen, press PF2 (Add) to create a definition.

Or:

On the **Symbol Table Version Usage** screen, enter the line command M next to the version you want to modify, and press ENTER.

The input fields that appear are shown in the following example:

```

18.12.13          ***** Entire Operations *****          10:48:42
                   Symbol Table Version Usage
Owner SN          Symbol Table A-1
-----
Symbol Table Version ==> v12.44____
is to be used as current Version      (current)
           from ==> 01.01.14      (empty: unlimited)
           to ==> _____      (empty: unlimited)

Version Usage Description
starting 2014_____

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

- 2 Enter the required values. The fields are explained in *Fields and Columns: Version Usage Definition*.
- 3 Press PF5 (Save) when you are finished.

Fields and Columns: Version Usage Definition

Field/Column	Description
Cmd	Input field for line commands: see <i>Commands: Version Usage Definition</i> .
from (column or input field)	<p>Start date of a default usage period of a network version.</p> <p>>>>>>>> denotes that no start date has been defined.</p> <p>Possible value for an input field:</p> <p>A start date in the format YY-MM-DD or no value (empty field) for no start date. See also <i>Date and Time Formats</i>.</p> <p>Default: start date is not defined.</p> <p>Note: Date ranges must not overlap.</p>
to (column or input field)	<p>End date of a default usage period of a network version.</p> <p>>>>>>>> denotes that no end date has been defined; the usage is unlimited (infinite).</p>

Field/Column	Description
	<p>Possible value for an input field:</p> <p>An end date in the format YY-MM-DD or no date (empty field) for an unlimited (infinite) time. See also Date and Time Formats.</p> <p>Note: Date ranges must not overlap.</p> <p>Default: end date is not defined.</p>
Version (column)	Symbol table version for which version usage is defined.
Description (column)	Description of the defined version usage.
Symbol Table Version	<p>Input field for the symbol table version to be used as the current version during the period specified in the from/to fields.</p> <p>This field contains (unnamed) by default.</p> <p>Enter the required version or use an asterisk (*) as a wildcard to select a version from a list.</p>
Version Usage Description	<p>Input field for a description of the version usage.</p> <p>Maximum input: 70 characters.</p>

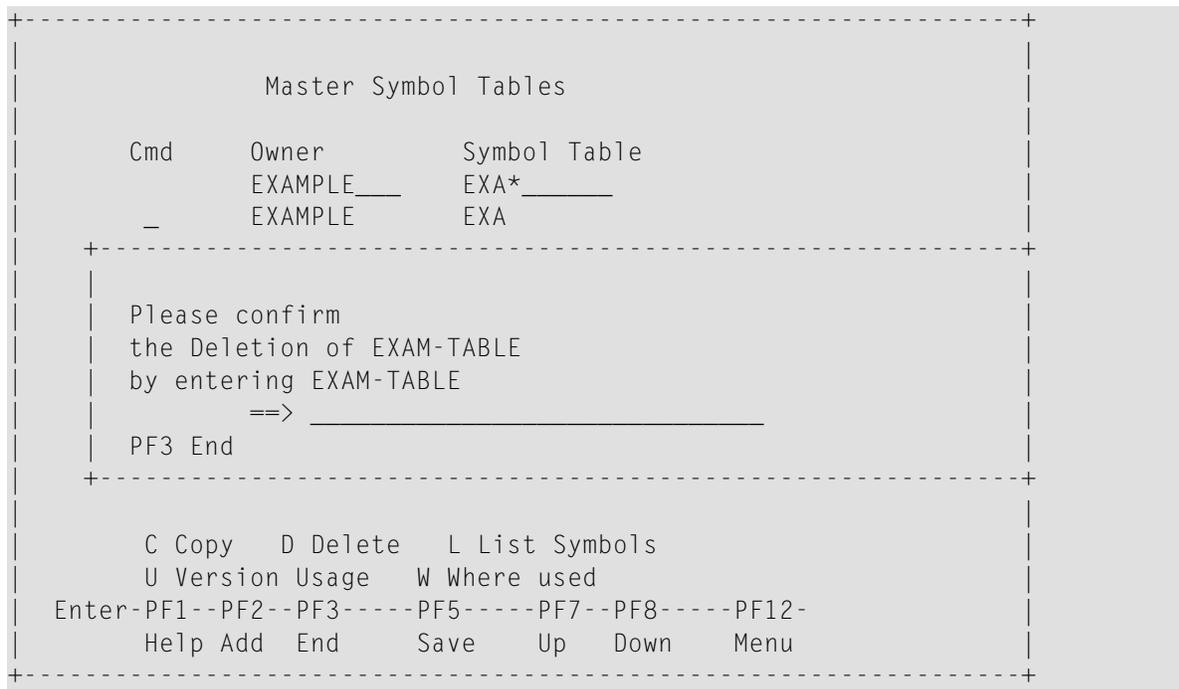
Commands: Version Usage Definition

The following line commands and special PF keys are available on the [Symbol Table Version Usage screen](#):

Line Command	Description
D	Deletes a version usage definition.
M	Modifies a version usage definition: see To add or modify a version usage definition .

Special PF Key	Description
PF2 (Add)	Adds a version usage definition: see To add or modify a version usage definition .

If the symbol table (and individual symbol table version) is no longer used or defined elsewhere, a window similar to the example below opens:



- 3 Enter the symbol table name and press ENTER to delete the table.

66 Finding Symbol Tables Associated with Jobs and Networks

- Listing Usable Symbol Tables 634
- Listing Jobs and Networks that use a Symbol Table Master 636

You can list all symbol tables available for use in a job or a network, and list all symbol tables already used by a job or a network.

Listing Usable Symbol Tables

You can list all symbol tables that can be used by a job or a network.

➤ **To list symbol tables used by a job or network**

- Open a **Job Definition (Master)** (or **Job Definition (active)**) or a **Network Modification** window and press PF7 (Symb).

A **Usable Symbol Tables** window similar to the example below opens:

Usable Symbol Tables						
Owner	Network	Version	Run	Job		
EXAMPLE	B60-FLOW		2223	JOB-04		
Cmd	Type	Owner	Sym.Table	Version	Network	Run
—	JA	EXAMPLE	EXAM-ST1		B60-FLOW	2223
—	JM	EXAMPLE	EXAM-ST1			
—	SD	SYSDBA	A			

L List

PF1 Help PF3 End PF7 Up PF8 Down

This window lists all symbol tables which can be used by the current object (network, job master, job active). The example above lists the symbol tables that can be used by the active job JOB-04. The columns are explained in [Columns: Usable Symbol Tables](#).

The symbol tables are listed in descending hierarchical order, that is, the topmost symbol table is used first during symbol replacement.

For information on all types of symbol tables available and the search order that applies when replacing symbols, see [Symbol Table Types and Symbol Search Order](#).

This section covers the following topics:

- [Columns: Usable Symbol Tables](#)
- [Line Commands: Usable Symbol Tables](#)

Columns: Usable Symbol Tables

The columns contained in the [Usable Symbol Tables window](#) are described in the following table:

Column	Description
Cmd	One-character line command input field. For possible commands, see Line Commands: Usable Symbol Tables .
Type	Type of the symbol table: see Symbol Table Types and Symbol Search Order .
Owner	Owner of the symbol table.
Sym.Table	Name of the symbol table.
Version	Version of the symbol table.
Network	For active symbol tables only Network which uses the active symbol table.
Run	For active symbol tables only. Run number of the active symbol table.
Job	Job that uses the symbol table. For an active job, the run number is also indicated.

Line Commands: Usable Symbol Tables

Use the following line command to perform the described function on any symbol table listed in the [Usable Symbol Tables](#) window:

Line Command	Description
L	List symbols in the selected master or active symbol table. Symbols from tables listed there can be modified only by an administrator, or a user when the symbol table owner is linked to a user whose profile setting Network Maintenance / Symbol Tables is at least W (read and write allowed). If several symbol table versions exist, the symbol table version selection window opens first. Select a symbol table version.

Listing Jobs and Networks that use a Symbol Table Master

You can list all jobs and networks that use a symbol table master. This is required, for example, if you cannot delete a symbol table because it still uses one or more jobs. You then need to find these jobs and change their symbol table settings or delete the jobs.

➤ **To list all jobs specified for a symbol table master**

- Type **W** in the line command input field of the selected symbol table in the **Master Symbol Tables window**. Press **ENTER**.

A **Symbol Table Usage** window similar to the example below opens:

```

03.05.16          ***** Entire Operations *****          11:48:00
Owner +-----+
--- + |
      | |                                     Symbol Table Usage
      | |
      | |   Owner EXAMPLE   Symbol Table EXAM-ST1   Version (all)_____
1     | |
2     | |   Network      Version      Run Job      Type  SymTab Version
3     | |   B60-FLOW
4     | |   B60-FLOW
5     | |   B60-FLOW
6     | |   B60-FLOW
7     | |   B60-FLOW
8     | |   B60-FLOW
9     | |   B60-FLOW
10    | |   B60-FLOW
11    | |
      | |   PF3 End   PF8 Down
      | |
      | |   +-----+
Com   | |   U Version Usage   W Where used
      | |   Enter-PF1--PF2--PF3-----PF5-----PF7--PF8-----PF12-
      | |   Help Add  End      Save    Up    Down    Menu
      | |   +-----+
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      Help           End                                     Owner Mail
  
```

The columns in the window are explained in *Fields and Columns: Symbol Table Usage*.

This section covers the following topics:

67

Defining Symbols and Symbol Values

▪ Listing Symbols of a Symbol Table	640
▪ Displaying a Symbol Definition	640
▪ Adding a Symbol Master	640
▪ Copying a Symbol within a Symbol Table or to another Symbol Table	641
▪ Modifying Symbols in a Symbol Table	642
▪ Defining Multiple Symbol Values	642
▪ Specifying a Range Check for Numeric Symbol Values	644
▪ Predefined Symbols	645
▪ Validating Symbol Values with a User Exit	653
▪ Global Symbol Modification Exit	654
▪ Symbol Setting triggered by the SYSOUT of a Job	654
▪ Deleting a Symbol from a Symbol Table	655

Listing Symbols of a Symbol Table

➤ To list symbols for a selected symbol table

- Type **L** in the line command input field of the selected symbol table in the **Master Symbol Tables** window. Press **ENTER**.

A **Master Symbol Table screen** appears.

The screen lists all symbol masters defined for the selected symbol table master.

The columns on the screen are explained in *Columns: Symbol Table Master*.

Displaying a Symbol Definition

➤ To display a symbol definition

- 1 Type **L** in the line command input field of the selected symbol on the **Master Symbol Table screen**. Press **ENTER**.

The **Master Symbol Browsing** window opens in the same format as the **Master Symbol Addition** window (see also *Adding a Symbol Table Master*).

- 2 Press **PF3 (End)** to return to the **Master Symbol Table** screen.

Adding a Symbol Master



Note: If this is the first symbol of a new symbol table, the table itself will be created implicitly by this action.

➤ To add a symbol to a symbol table

- 1 Press **PF2 (Add)** on the **Master Symbol Table screen**. The **Master Symbol Addition** window opens (see *Adding a Symbol Table Master*), with the name of the master symbol table already entered in the **Symbol Table** field. This field is not modifiable.
- 2 Enter the new symbol definition.
- 3 Press **PF5 (Save)** to save the new definition.
- 4 Press **PF3 (End)** to return to the **Master Symbol Table** screen.

Copying a Symbol within a Symbol Table or to another Symbol Table

> To copy a symbol definition within a symbol table or to another symbol table

- 1 Type C in the line command input field of the selected symbol on the **Master Symbol Table screen**. Press ENTER.

A **Symbol Copy** window similar to the example below opens:

```

19.12.13          ***** Entire Operations *****          14:10:24
Owner   SN          Master Symbol Table A-1          Version v12.44
Network +-----+
----- !
C Symbo !          Symbol Copy          !
 *---- !   From          !
_ ABCD !   Symbol ==> SHORT1_____          ! .13 15:25
_ FILE- !   Owner   ==> SN_____ Network ==> _____          ! .13 09:27
_ MULT- !   SymTab  ==> A-1_____ Run    ==> _____          ! .13 09:35
_ NUM-1 !   Version ==> v12.44_____          ! .13 10:19
c SHORT ! -----          ! .13 09:26
_ TEST !   To          !
_ Z1   !   Symbol ==> SHORT2_____          ! .13 16:44
_ Z2   !   Owner   ==> SN_____ Network ==> _____          ! .13 14:03
      !   Symtab  ==> A-1_____ Run    ==> _____          !
      !   Version ==> v12.44_____          !
      !          !
      ! PF1 Help  PF3 End          !
      +-----+
***** Bottom of Data *****
C Copy  D Delete  L List  M Modify
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Add   End   Copy  Save          Up   Down  Print

```

The upper half of the window shows the origin symbol. In the lower half of the window you must define the target symbol. With this function, you may copy to another symbol in the same table, or to a symbol in another symbol table and / or symbol table version. If you leave the window with PF3 without specifying a target symbol name, the copy will be aborted.

- 2 Enter the name of the symbol to copy to in the **To Symbol** field.
- 3 Press PF3 (End) to copy the symbol and return to the **Master Symbol Table** screen.

Modifying Symbols in a Symbol Table

> To modify a symbol definition

- 1 Type **M** in the line command input field of the selected symbol on the **Master Symbol Table screen**. Press **ENTER**.

The **Master Symbol Modification** window opens in the same format as the **Master Symbol Addition window** (see *Adding a Symbol Table Master*). The window contains the current values and the **Symbol Table** and **Symbol Name** fields are not modifiable.

- 2 Enter your modifications.
- 3 Press **PF5** (Save) to save the modified definition.
- 4 Press **PF3** (End) to return to the invoking location.

Defining Multiple Symbol Values

You can give a symbol several values which can be used in subsequent substitutions of the same symbol. These values can be used for the multiple parallel activation of a job. The number of parallel jobs is the same as the number of multiple values defined here.

> To view and define multiple symbol values

- 1 Open a **Master Symbol Modification** or **Active Symbol Modification** window.

If multiple values have already been defined for the selected symbol, the **Value** field indicates how many, for example: `*** 25 multiple values ***`.

- 2 Press **PF9** (Mult).

A **Multiple Symbol Values** window similar to the example below opens:

Symbol Replacement

You can use the **symbol function !MV or ?MV** (see *Functions for Symbol Replacement*) to get single occurrences of multiple-value symbols for replacement.

Specifying a Range Check for Numeric Symbol Values

You can guarantee that a numeric symbol value is within a specified number range.

> To specify a range check for numeric symbol values

- Open a **Master Symbol Addition/Modification window** for a symbol of the format N (numeric) and press PF8 (RgChk).

A **Symbol: Numeric Range Check** window similar to the example below opens:

```

19.12.13          ***** Entire Operations *****          10:12:03
Owner  SN          Master Symbol Table A-1          Version v12.44
+-----+-----+-----+-----+
!
!                Master Symbol Modification          !
!
! Symbol Table ==> A-1_____          Owner SN          ! .05.13 15:25
!      +-----+-----+-----+-----+          ! .12.13 09:27
! Symbo !          !          !          !          ! .12.13 09:35
!      !          Symbol: Numeric Range Check          !          ! .12.13 10:12
!      !          !          !          !          ! .12.13 09:26
!      ! Owner          ==> SN          !          ! .03.12 15:03
! Be car ! Symbol Table ==> A-1          !          ! .05.13 16:44
! The 2 ! Version          ==> v12.44          !          !
!      Pr ! Symbol          ==> NUM-1          ! _          !
!      Prom !          !          !          !          !
!      ! Value from          ==> 1_____          !          !
!      ! Value to          ==> 999_____          !          !
!      !          !          !          !          !
!      ! ---PF1---PF3-----PF5-----PF9-----          !          ! *****
! Enter- !      Help End          Save          Delete          ! --          !
!      +-----+-----+-----+-----+          ! F11---PF12---
+-----+-----+-----+-----+

```

In the following example, a value was entered outside the specified value range. In this case, an error message appears and Entire Operations does not accept the invalid value. This also applies to user exits that perform symbol checks (see also *User Exits for Validation Checks of Symbol Values*).

- [Predefined Symbols for Subnetworks and Recovery Jobs](#)

Predefined Symbols for Command Line Parameters

Symbol	Format	Description
CMDLINE- <i>job</i>	A80	<p>Command line parameter(s) for the following job types and/or JCL locations:</p> <ul style="list-style-type: none"> ■ Started tasks (job type STC) on z/OS, ■ UNIX shell scripts (job type JOB, ■ Windows BAT files (job type JOB) and PowerShell scripts (job type WPS), ■ Windows services (job type SRV), and ■ Programs which are to be executed directly under UNIX or Windows (JCL location EXE).

Passing Command Line Parameters

If you want to pass command line parameters for a job, use the predefined symbol `CMDLINE-job` in the job's symbol table, where *job* is the job name.

Several parameters, separated by blanks, can be contained in this symbol.

Example:

Job name:	JOB1
Member name in JCL definition:	STC001
Content of symbol CMDLINE-JOB1:	PARAM1=Y , PARAM2=N

The start command generated and issued by the Entire Operations Monitor is:

```
S STC001 , PARAM1=Y , PARAM2=N
```

Command Line Passing to Shell Scripts in Entire Operations

- If a job is to be submitted, Entire Operations checks whether a symbol `CMDLINE-job` exists.

If such a symbol exists, its content is appended to the invocation of the `job.B` script.

- The `job.sh` resp. `job.bat` script (the user's script, with some Entire Operations header files), is not invoked directly, but out of the wrapping script `job.nnnnnnnn.frame.sh` resp. `job.nnnnnnnn.frame.bat`.

`nnnnnnnn` denotes an 8-digit repeat count. For the first job submission, this is 00000000.

- The `job.nnnnnnnn.frame.sh` resp. `job.nnnnnnnn.frame.bat` script is generated by Entire Operations at submission time. On UNIX, this is always a Bourne shell script (starting with `#!/bin/sh`), regardless of the shell of the user's script.

- UNIX: Command line parameters must be passed in Bourne shell syntax. This means quoting and masking of special characters according to Bourne shell rules. Note that the Bourne shell script quotes from the parameters quoted in the invoked shell script.
- Use `c:\work\test.txt` to pass a windows file name, for example, `c:\work\test.txt`.
- The maximum total length of all parameters is 1000 bytes (including separating blanks).
- The maximum length of a single parameter is 240 bytes.

Predefined Symbols for Mainframe Platforms

Symbol	Format	Description
SYSOUT-NODE-GLOBAL	N5	(Only in table SYSDBA/A.) If this SYSOUT exists and contains a valid Entire System Server node number and if there is no other applicable SYSOUT node definition available, SYSOUT file copies for Entire Output Management are created on this node.
JCL-NODE	N5	If this symbol is found during job activation, this node is used as a JCL node for the active job.
SUBMIT-NODE	N5	If this symbol is found during job activation, this node is used as a job start node for the active job.
SYSOUT-NODE	N5	If this symbol is found during job activation, this node is used as a SYSOUT node for the active job. This symbol supersedes SYSOUT-NODE-GLOBAL.

The symbols JCL-NODE, SUBMIT-NODE and SYSOUT-NODE are searched for in the symbols tables as described in [Symbol Table Types and Symbol Search Order](#).

Predefined Symbols for FTP Jobs

There are some symbols which have a special meaning for jobs of the type **FTP**.

Wildcards mean that all symbols starting with the mentioned prefix are reserved.

Symbol	Meaning
FTP-JOBC*	FTP job cards.
FTP-SITE*	FTP site commands.
FTP-PARM*	FTP command line parameters.

Table of Predefined Symbols

The following symbols can be used without defining them in a symbol table:

For further explanations of date and time formats, see also [Date and Time Formats](#).

Symbol	Format	Description
*DATD	A8	Date in the format DD.MM.YY.
*DATE	A8	Date in the format DD/MM/YY.
*DATG	A15	Date in the format DD ^{month} YY.
*DATI	A8	Date in the format YY/MM/DD.
*DATJ	A5	Date in the format YYDDD (Julian).
*DATN	N8	Date in the format YYYYMMDD.
*DATU	A8	Date in the format MM/DD/YY.
*DAT4J	A7	Date in the format YYYYDDD (Julian).
*TIME	A10	Time of day in the format HH:MM:SS.T.
*TIME8	A8	Time of day in the format HH:MM:SS.
*TIMN	N7	Time of day in the format HHMMSS. See also the symbol function !TIMN. This function allows the usage of constant time values.
*TIMN6	N6	Time of day in the format HHMMSS.
*TIMA6	A6	Time of day in the format HHMMSS.
*TIMA7	A7	Time of day in the format HHMMSS.T.
§PMPA (escape character § or as defined)	A5	If an escape character (here: §) is defined in a job master JCL node , an execution node , or a SYSOUT node definition, the content of the suffix symbol of the multiple-value symbol is used. This allows the execution of multiple parallel jobs on different nodes. The multiple-value symbol must be an array of valid node numbers or mnemonic node short names.
P-ACT-USER	A8	For manual activations: user ID of the user who activated the network or job. For subnetwork activations: activation user ID of the calling (parent) job. For scheduled activations: default user ID of the Entire Operations Monitor.
P-ACT-ORIGIN	A1	Contains a letter that indicates the activation origin: A by API

Symbol	Format	Description
		E by EOJ action
		M manual activation
		R recovery
		S by schedule extraction
		U by API, as a subnetwork
P-DATE	A8	Execution date in the format YYYYMMDD. The date of the previous day is transmitted if the end time of the previous production day has not yet been reached.
P-EXEC-NODE	N5	Execution node.
P-EXECUTION-NODE	N5	Execution node.
P-JCL-FILE	A250	<p>JCL file.</p> <p>If not applicable, an empty string is returned. P-JCL-FILE is always resolved to a fully qualified file name.</p> <p>Recursive resolution of symbols is performed.</p> <p>If the JCL node is a Windows node, backslashes (\) in the file name can be returned by slashes.</p> <p>For further usage in Windows BAT and Powershell JCL, the string +F+&P-JCL-FILE is to be used (with & as escape character).</p> <p>Windows example, assuming the ampersand (&) is the escape character:</p> <pre>P-JCL-FILE = e:/sag/nop/jcl/job1.bat</pre> <pre>type +F+&P-JCL-FILE</pre> <p>is resolved to type</p> <pre>e:\sag\nop\jcl\job1.bat</pre>
P-JCL-MEMBER		<p>JCL member.</p> <p>If not applicable, an empty string is returned.</p>
P-JCL-NODE	N5	JCL node.
P-JOB	A10	Job.
P-JOB-ID	A10	Job ID (job number, BS2000 TSN, process ID).
P-JI	A10	Job ID (job number, BS2000 TSN, process ID).
P-JOBCOUNT	A8	<p>Job type SAP only:</p> <p>Job count in the SAP system.</p>

Symbol	Format	Description
		Note: The field is available after job termination.
P - MUL	A8	User who made last change (in lower case).
P - MUU	A8	User who made last change (in upper case).
P - NADIR	A250	Directory for temporary files for this network (UNIX and Windows only). Note: 1. On Windows nodes, the directory name contains backslashes (\). For the usage on mainframe (EBCDIC) environments, these backslashes will be returned encoded in trigraphs (see the relevant section in <i>Submission of Jobs by Entire Operations</i>). 2. The content of P - NADIR can be subject to change in different Entire Operations versions.
P - NETWORK	A10	Network.
P - NETWORK - VERSION	A10	Network version.
P - NODE	N3	Execution node.
P - OWNER	A10	Owner of the network.
P - REPEAT or P - REPEAT - COUNT	N10	Repeat count. Contains the number of the current repetition of the job (if the job was resubmitted, etc.).
P - RUN	N5	Run number of the network.
P - RUN5	N5	Run number of the network, always with 5 digits (with leading zeros). For example: The run number 7 will be returned as 00007. This symbol is also available in the macro JCL.
P - SUL	A8	Job start user ID (in lower case).
P - SUU	A8	Job start user ID (in lower case).
P - SUBMIT - ID	A20	User ID under which the job was started.
P - SUBMIT - GRP	A20	UNIX group or Windows domain under which the job was started.
P - SYSOUT	A54	Name of the SYSOUT file. This variable is dependent upon the operating system in which the job is executed: BS2000

Available any time

Symbol	Format	Description
		after the activation.
		z/OS and z/VSE Not available. The value is set to three consecutive periods (...).
		UNIX Available any time after the activation.
		Windows Available any time after the activation. The file name is returned in UNIX syntax.
P-SYMTAB	A10	Symbol table.
P-SYMBOL-TABLE	A10	Symbol table.
P-SYMTAB-VERSION	A10	Symbol table version.

Predefined Symbols for Multiple and Parallel Activations

Symbol	Format	Description
P-SUFFIX or P-X	A10	The job name suffix if multiple and parallel job activation is in use. You can find detailed information about multiple and parallel job activation in the section Job Maintenance . P-X is an abbreviation for P-SUFFIX.
P-MPA	A50	The full current value of the symbol containing the job name suffix if multiple and parallel job activation is in use. To redefine this field, code: <pre> \$ 1 P-C-MPA (A50) \$ 1 REDEFINE P-C-MPA ... #GET-SYMBOL P-C-MPA </pre>
P-MPI	N3	The numeric index of the multiple-value symbol used for a multiple parallel activation.

Predefined Symbols for Subnetworks and Recovery Jobs

The table below lists and describes the symbols a subnetwork or recovery job requires to communicate with its invoking network or the calling job, respectively.

These symbols have the value of a blank, unless they are used in a subnetwork or a recovery job.

Symbol	Format	Description	Rec. Job	Subnet
P-C-OWNER	A10	Owner of the calling job or invoking network.	X	X
P-C-NETWORK	A10	Network of the calling job or invoked subnetwork.	X	X
P-C-NETWORK-VERSION	A10	Network version of the calling job or invoked subnetwork.	X	X
P-C-RUN	N5	Run number of the calling job or invoking network.	X	X
P-C-JOB	A10	Name of the calling job.	X	X
P-C-SUFFIX	A10	Suffix value of the job which invoked the subnetwork where the symbol is used.	X	X
P-C-JCL-NODE	N5	JCL node of the calling job.	X	n/a
P-C-EXEC-NODE	N5	Execution node of the calling job.	X	n/a
P-C-SYMTAB	A10	Symbol table of the calling job or invoking network.	X	X
P-C-SYMTAB-VERSION	A10	Symbol table version of the calling job or invoking network.	X	X
P-C-MPA	A50	The full current value of the symbol, containing the suffix of the job which invoked the subnetwork if multiple and parallel job activation is in use in the calling network. To redefine this field, code: <pre> § 1 P-C-MPA (A50) § 1 REDEFINE P-C-MPA ... #GET-SYMBOL P-C-MPA </pre>	X	n/a
P-C-SUU	A20	Submit user ID of the calling job. It is possible to define this symbol as submit user ID in the jobs of the subnetwork, preceded by the activation escape character. (The subnetwork jobs must have a symbol table definition.)	X	n/a
P-C-SUG	A20	Submit group of the calling job. It is possible to define this symbol as submit group in the jobs of the subnetwork, preceded by the activation escape character. (The subnetwork jobs must have a symbol table definition.)	X	n/a

Validating Symbol Values with a User Exit

You can define a user exit to check the validity (plausibility) of symbol values during symbol modification or prompting.

For detailed information on writing a user exit, see *User Exits for Validation Checks of Symbol Values* in the section *User Exits*.

➤ **To check symbol values with a user exit**

- 1 Press PF11 (Exit) in a **Master Symbol Addition/Modification window**.

A **Definition of Symbol Check Exit** window similar to the example below opens:

```

19.12.13          ***** Entire Operations *****          10:19:58
Owner   SN          Master Symbol Table A-1          Version v12.44
+-----+
!      +-----+
!      !
!      !          Definition of Symbol Check Exit          !
!  Symbo !
!      !  Owner          ==> SN          !
!  Symbo !  Symbol Table ==> A-1          !
!      !  Version        ==> v12.44       !
!      !  Symbol         ==> FILE-1       !
!      !
!  Be car !  Library      ==> _____ !
!  The 2  !  Exit         ==> _____ !
!      Pr !
!  Prom  !  ---PF1-----PF3---PF4---PF5-----PF9----- !
!      !      Help      End   Edit  Save      Delete      !
!      +-----+
!
!          _____ !
!          _____ ! *****
! Enter-PF1---PF2---PF3-----PF5---PF8---PF9---PF10---PF11-- !
!      Help Add   End   Save  RgChk Mult Copy  Exit  ! F11--PF12---
+-----+

```

- 2 Enter or change the entries in the **Library** and **Exit** fields. The input fields are described in *Definition of Symbol Check Exit* in the section *Fields: Master and Active Symbols*.
- 3 Press PF5 (Save) when you are finished.

Special PF Keys: Symbol Check Exit

The PF keys available to edit or delete a user exit are described in the following table:

PF Key	Name	Function
PF4	Edit	Edit or create a user exit. Do not forget to STOW the routine so it can be used.
PF9	Delete	Removes the user exit from the selected symbol definition but keeps all related user exit objects in the library.

Global Symbol Modification Exit

For the active symbol modification, a global symbol modification exit (described in the *Administration* documentation) can be defined too. This global exit is only used if there is no specific exit defined on the network level. The global exit is called with the same parameter list as the exit on the network level.

Symbol Setting triggered by the SYSOUT of a Job

(Not applicable on mainframes)

It is possible to set symbols during a job's end-of-job checking, triggered by a meta command in the job's SYSOUT. The meta command is EOR-SYMBOL.

With this meta command it is possible to set an Entire Operations symbol during the end-of-job checking of a job.

The meta command is available for jobs running on UNIX or Windows.

The symbol will be set into the active job's active or master symbol table. If the active job has no active symbol table, the job result is set to `not ok`.

Syntax:

```
EOR-SYMBOL SA symbol=value
```

Sets an active symbol.

```
EOR-SYMBOL SM symbol=value
```

Sets a master symbol.

The meta command can be generated, for example, with an echo command:

```
echo "EOR-SYMBOL SA SYMBOL1=$HOSTNAME"
```

With `HOSTNAME=pcsn` this produces the `SYSOUT` line:

```
EOR-SYMBOL SA SYMBOL1=pcsn
```

During end-of-job check, the symbol `SYMBOL1` in the job's active symbol table will be set to `pcsn`.

The `echo` command can be generated dynamically by using symbol replacement. Therefore, this kind of symbol setting is very flexible.

Example:

```
echo "EOR-SYMBOL SA PID.&*TIMN=$$"
```

creates the `SYSOUT` line:

```
EOR-SYMBOL SA PID.1131185=4837
```

Deleting a Symbol from a Symbol Table

The options provided to delete a symbol master or symbol active from a symbol table are described in the following section.

➤ To delete a symbol from the master symbol table

- 1 On the **Master Symbol Table screen**, type `D` in the line command input field next to the symbol you want to delete. Press `ENTER`.

A confirmation window opens.

- 2 Type in the symbol name to confirm deletion and press `ENTER`.
- 3 Press `PF3 (End)` to return to the **Master Symbol Table** screen.



Note: You can delete individual symbols from a symbol table even if the table is specified for one or more MAC-type jobs. If the symbol table is specified for any job, the deleted symbol cannot be substituted when the JCL is dynamically generated. The Monitor records this event by writing a message to the log.

68

Symbol Replacement

- Symbol Escape Characters 658
- Symbol Prompting during Network or Job Activation 660
- Displaying Help Text for a Symbol 663
- Canceling Symbol Prompting 664
- Repeating Symbol Prompting 664
- Entering Long Symbol Values and Modifying Active Symbols 664
- Specifying User Exits for Symbol Modification 666
- Rules and Restrictions for Symbol Replacement 669

Symbols are replaced during network or job activation or during job submission:

- Symbols starting with the **activation escape character** are resolved during JCL loading, as part of the job activation.
- Symbols starting with the **submit escape character** are resolved during job submission.



Note: If a symbol replacement error occurs at submission time, the event `JNR Job not run - JCL error` will be set to occurred. If the event JNR is not defined, it will be added to the active job.

Note for BS2000:

BS2000 SYSJV references (for example, `&($SYSJV.TSN)`) are no longer treated as Entire Operations symbols, regardless of whether they are prefixed with an ampersand (&) denoting an activation escape or submit escape character.

Symbol Escape Characters

In JCL and Natural macro sources, the symbols to be substituted must be preceded by an escape character.

You can use different escape characters to determine whether symbol replacement is performed during JCL generation or job submission.

The following escape characters can be used in parallel in your Entire Operations environment:

■ Activation Escape Character

Symbols preceded by the activation escape character are replaced during activation of a network or job.

If a JCL file contains the activation escape character, a symbol replacement is performed:

- During JCL editing (temporary, using the master symbol table).
- During JCL loading (permanently, using the active symbol table).

■ Submit Escape Character

Symbols preceded by the submit escape character are replaced during submission of a job. They can be used to pass symbol values that cannot be specified earlier when the job is activated.

Both activation escape and submission escape characters can be defined as system-wide defaults in the Entire Operations default settings (see *Default Setting 1* in the *Administration* documentation). However, different escape characters can be used for each job if necessary.

You can also specify escape characters in your JCL or MACRO source. See also [Defining Escape Characters](#).

 **Caution:** If you change escape character definitions for existing JCL, symbols to be replaced can no longer be detected. In this case, consider defining escape characters in your JCL or MACRO source.

Recommended Escape Characters

We recommend that you use the following escape character:

Operating System	Escape Characters
z/OS	§ (section sign) and \$ (dollar sign)
z/VSE	§ (section sign) and # (number sign)
BS2000	^ (circumflex accent) and ` (grave accent) Do not use \$ (dollar sign) because it has a special meaning in BS2000.
UNIX, Windows	No recommendation, except for UNIX: Do not use \$ (dollar sign) because it has a special meaning in UNIX.

 **Caution:** We strongly recommend that you do not define escape characters already used by other (operating system specific) JCL features, for example, parameters for DO procedures and SUBDTA characters in BS2000.

Defining Escape Characters

This section describes the options provided to define your default activation and submit escape characters.

■ Default Settings

The **Default Setting (1)** screen of the **Entire Operations Defaults** function (see the *Administration* documentation) is used to define the default for all networks in your Entire Operations environment.

You can also define the default escape characters for each operating system class by using PF10 (OSpec) on the **Default Settings (1)** screen.

■ Network Definition

Defines the default escape characters to be used by all jobs of a network.

■ Job Definition

Defines the default escape characters to be used by an individual job.

■ Definitions in JCL and Macro Sources

In JCL and Natural macro sources, the symbols to be substituted must be preceded by an escape character.

You can define default escape characters in Entire Operations JCL or Natural macro sources by coding one or both of the following meta statements as the first source line(s):

```
#EOR-ESC-ACT = activation-character  
#EOR-ESC-SUB = submit-character
```

For example:

```
#EOR-ESC-ACT = §  
#EOR-ESC-SUB = ^
```

The `#EOR-ESC-ACT = activation-character` meta statement is evaluated when the JCL containing this statement is loaded. *activation-character* overrides any activation escape characters currently used in your active job networks. The meta statement is removed from the active JCL after loading.

The `#EOR-ESC-ACT = submit-character` meta statement is evaluated during job submission. *submit-character* overrides any submit escape character currently used in the active job. The meta statement is kept in the active JCL after loading.

Both meta statements are logged when they are evaluated during JCL loading or job submission.

If you add a meta statement to a Natural macro source, you have to recatalog the source with the `MACRO` command.

Symbol Prompting during Network or Job Activation

Entire Operations provides a standard method for symbol prompting. Standard symbol prompting is not used in NET-type (subnetworks) jobs. On a full-sized screen, all necessary symbols are displayed and can be modified by simply typing over them.

If any of the activated jobs are MAC-type jobs (dynamic JCL generation) or JOB-type jobs whose JCL contains symbols for variable values, these symbols are replaced by their current values during the manual activation, except those that are defined to be replaced at submission time.

In a job network definition and in the subsidiary jobs, several symbol tables can be defined. During a network or job activation, all symbol tables are prompted sequentially, which contain at least one symbol to be prompted.



Note: Up to 100 different symbol tables can be prompted during a network activation.

2. If a validation check is defined for the symbol, you can only enter a valid symbol value. Incorrect values are rejected and a user-defined error message is issued.

2 When you have made all desired modifications, press PF5 (Accpt) to confirm symbol prompting and accept all symbols for this activation.

If there are several screens of symbols, **** more **** appears in the bottom line.

Press PF8 (Down) to display more symbols until you get to the last screen. You can then accept all symbols by pressing PF5.

3 If defined in your user profile, you are then asked to print the symbols. A window similar to the example below opens in which you can enter the printer name:

```
+-----+
!      Printing of Prompted Symbols      !
!                                     !
!           Printer Selection           !
!                                     !
!   Please enter printer: _____   !
!                                     !
!   PF3 End                             !
!                                     !
+-----+
```

4 Enter the printer name in the field **Please enter printer**.

5 Press ENTER to output the symbols on the selected printer.

If you do not want to print the symbols, press PF3 (End) to return to the **Symbol Prompting for Table** screen.

If you generally do not want to print the symbols or display the above window, enter N for the function **Symbol Printing after Prompting** in the **Reporting Functions** window of your **User Definition and Profile** settings. For more information, see the section *User Definitions and Profile Settings* in the *Administration* documentation.

6 Press PF5 (Accpt) when you are finished.

Line Commands: Symbol Prompting

Line Command	Description
H	Display help text for the symbol.
L	Enter a value longer than 40 characters and modify other symbol parameters. See also Entering Long Symbol Values and Modifying Active Symbols .

Special PF Keys: Symbol Prompting

PF Key	Name	Function
PF3	End	Cancel symbol prompting and restore to former status.
PF5	Accpt	Accept all symbols for this activation and continue with the activation. If the list of prompted symbols exceeds one screen, scroll to the end of the list with PF8 first, before using PF5.
PF6	Cncl	Cancel this activation. If symbol prompting is done during a network or job activation, the corresponding activation is cancelled. Otherwise, only symbol prompting itself is cancelled.
PF7	Up	Scroll back to the top of the symbol list.
PF8	Down	If **** more **** appears in the bottom line, you can display more symbols by scrolling down with PF8. You must press this key to get to the last screen and accept all symbols with PF5.

Displaying Help Text for a Symbol

➤ To display a short help text for a symbol

- 1 Type H in the line command input field of the appropriate symbol on the [Symbol Prompting for Table](#) screen.
- 2 Press ENTER. A window opens with a short help text explaining the symbol.

Canceling Symbol Prompting

➤ To cancel symbol prompting

- Press PF6 (Cncl).

Repeating Symbol Prompting

➤ To repeat symbol prompting from the beginning

- Press PF6 (Cncl) to cancel the current network or job activation. Start the activation again.

Entering Long Symbol Values and Modifying Active Symbols

If a symbol value is longer than the 40 characters on the **Symbol Prompting for Table** screen, you can enter another 40 characters (up to a maximum of 80) in the **Active Symbol Modification** window. This window also allows you to modify other symbol parameters.

➤ To enter long symbol values and modify active symbols

- Enter L in the line command input field of the appropriate symbol on the **Symbol Prompting for Table** screen. Press ENTER.

An **Active Symbol Modification** window similar to the example below opens:

```

29.10.10          ***** Entire Operations *****                               10:15:11
Owner   REQUEST   Symbol Prompting for Table P185905
Network P185905   Run 21   on 29.10.10 at 10:10
+-----+-----+
!
!           Active Symbol Modification           ! nger-than-o
!
! Symbol Table ==> P185905___   Owner REQUEST   !
! Symbol Name  ==> LONG1_____ Network P185905  !
! Format       ==> A           Run 21           !
! Value       ==> This-is-a-very-long-value-which-is-much- !
!              longer-than-one-line_____       !
! Be careful with insertions or deletions.       !
! The 2 lines will be concatenated to one symbol value. !
! Prompting   ==> A           Update Mode ==> _   !
! Prompt Text ==> _____                       !
!            _____                           !
!            _____                           !
!            _____                           !
! Enter-PF1---PF2---PF3-----PF5----PF8----PF9---PF10---PF11-- ! *****
! Help Add End Save RgChk Mult Copy UsrRt ! F11--PF12---
+-----+-----+

```

This window allows you to modify the symbol value. The two lines of the **Value** field are concatenated for the long symbol value.



Caution: Be careful with insertions and deletions, since the two lines of **Value** are concatenated to one symbol value.

This section covers the following topics:

- [Fields: Active Symbol Modification](#)
- [Special PF Keys: Active Symbol Modification](#)

Fields: Active Symbol Modification

The fields in the [Active Symbol Modification window](#) are identical to the fields in the **Master Symbol Addition** window. For field descriptions, see [Fields: Master and Active Symbols](#).

Special PF Keys: Active Symbol Modification

For a description of special PF key functions, see [Special PF Keys: Master Symbol Addition](#).

Specifying User Exits for Symbol Modification

You can specify a customized user exit routine that defines your own validation checks and help information for symbol prompting. This can be used to modify active symbols depending on the values of other symbols, or of anything else.

You can then adapt a symbol table during each job run with or without using a window that prompts you for input. If you do not use a prompting window, the symbol table is adapted automatically in the background.

For details on defining user exit routines, see [User Exits for Validation Checks of Symbol Values](#) in the section *User Exits*.

➤ To specify a user exit for symbol modification

- 1 On the [Network Maintenance screen](#), type M in the line command field of the network for which symbol prompting is to be defined. Press ENTER.

A [Network Modification window](#) opens with the current values for the network.

- 2 Press PF8 (SP-UR).

A **Symbol Prompting Exit** window similar to the example below opens:

```

31.10.11          ***** Entire Operations *****          15:23:18
Owner SN          Network Maint +-----+
Selection OR_____ ! !
+-----+
!
!                      Network Modification                      !
! Network      ==> A-1_____ Owner ==> SN                      !
! Description  ==> several tests ..._____                      !
!                                                    Loop existing ==> N !
! Default Val +-----+
! Execution !
! JCL Node !                      Symbol Prompting Exit          ! _ !
! JCL Locat !
!                      Symbol Prompting for this network will be ! X !
! File == ! done by
! VolSer == !                      Natural Library ==> EOR-T531 ! _ !
!                      Exit ==> SYP00001                          ! !
! --PF1-----P !                      Modification                ! -- ! **
! Help E !                      without Prompting ==> N (Y/N/B) ! !
+-----+
Command => _____ ! PF1 Help PF3 End PF4 Edit PF5 Refr PF9 Delete ! !
Enter-PF1---PF2- +-----+ + F12---
      Help Add End Save Up Down NxtSt Menu
  
```

Enter the name of the user exit you want to use for symbol prompting. The fields and special PF keys available are described in [Fields: Symbol Prompting User Exit](#).

If no user exit yet exists, you can write a user exit routine as described in [User Exits for Validation Checks of Symbol Values](#) in the section *User Exits*.

This section covers the following topics:

- [Fields: Symbol Prompting User Exit](#)
- [Special PF Keys: Symbol Prompting Exit](#)
- [Example of an Active Symbol Table List](#)

Fields: Symbol Prompting User Exit

The input fields of the [Symbol Prompting Exit window](#) are described in the following table:

Field	Description
Natural Library	Name of the Natural library where the symbol prompting user exit resides.
User Exit	Name of the symbol prompting user exit.
Modification without Prompting	Symbol modification option(s). Possible input values:

Field	Description	
	N	A window prompts you to modify and/or confirm all symbols and output the active symbol table on a printer. See also Example of an Active Symbol Table List .
	Y	No prompt window opens. The symbol modification is managed by the user exit routine which runs in the background.
	B	Performs both modification options: A prompt window opens first and then symbol modification is managed by the specified user exit. This allows you to set other symbols which are dependent on prompting.

Special PF Keys: Symbol Prompting Exit

You can perform the following functions from the [Symbol Prompting Exit window](#) using these PF keys:

PF Key	Name	Function
PF4	Edit	Edit the defined symbol prompting user exit.
PF5	Refresh	Force an update of the user exit in the internal exit directory.
PF9	Delete	Delete the definition of the symbol prompting user exit. This does not delete the exit itself.

Example of an Active Symbol Table List

The following is an example of an output file generated for an active symbol table using the GUI client:

Active Symbol Table

Owner: SAGTEST Network: SAG-PROMPT Run: 26 Symbol Table: EXAM-TABLET-V3

Symbol	MV	Symbol Value	Modified by	Time
CLASS		K	IMPORT	2014-01-10 12:48:54
JOBLIB		NOP.EXAMPLE.LOAD	IMPORT	2014-01-10 12:48:54
MSGCLASS		X	IMPORT	2014-01-10 12:48:54
STEPLIB		NOP.EXAMPLE.LOAD	IMPORT	2014-01-10 12:48:54
TESTSYMBOL		a	NATQA	2016-05-12 15:28:03

The file lists the current status of the symbols defined for the symbol table EXAM-TABLET-V3 which is prompted when the network SAG-PROMPT is activated.

Rules and Restrictions for Symbol Replacement

**Notes:**

1. A symbol replacement is always attempted if at least one of the activation or submit escape characters currently in effect is found in a source line (see the section [Symbol Escape Characters](#)). It is not necessary for a symbol table to be defined on the job or network level. Symbols can also be replaced via [global symbol tables](#). [Predefined symbols](#) can be replaced without having to define a symbol table.
2. If an escape character is detected and no symbol table is defined, the error message `no symbol table defined` is issued.

This section covers the following topics:

- [Symbol Table Types and Symbol Search Order](#)
- [Nested Replacement](#)
- [Recursive Symbol Replacement](#)
- [Symbol Replacement in JCL](#)
- [Import/Export Restrictions](#)
- [Escape Character](#)
- [Indicator for End of a Symbol Name](#)
- [Symbol Replacement Algorithm](#)
- [Fixed Positions within a JCL Line](#)
- [Symbol Replacement with Multiple Symbol Values](#)
- [Symbol Replacement in Subnetworks](#)

Related Topic:

- [Predefined Symbols](#)

Symbol Table Types and Symbol Search Order

The types of symbol tables available and the search order used to find and replace symbols in symbol tables are described in the following section.

Entire Operations first searches for a symbol in the active user-specific symbol table. After this, it searches in the master user-defined symbol table, in the network default symbol table, and finally in the system-wide symbol table. If it searches for the symbol in a subnetwork, the symbol tables of the calling job of type NET as well as the symbol tables of the superior job networks are searched (in ascending order) if necessary.

If a symbol cannot be found anywhere, either the global exit “symbol not found” is called (if defined), which determines the further processing, or the job is set to an error status.

Symbol Table Type	Description	Search Order
JA Job active	Active symbol table referenced by the active job.	1
NA Network, active	Active symbol table referenced by the network.	2
CA Calling job	Active symbol table referenced by the calling job. This only applies if the active job is within a subnetwork . Depending on the subnetwork nesting level, several symbol tables of calling jobs can be searched. The search can repeat recursively ascending.	3
CN Calling network	Active symbol table referenced by the calling network. Depending on the subnetwork nesting level, several symbol tables of calling networks can be searched. The search can repeat recursively ascending.	4
JM Job master	Symbol table referenced by the master job.	5
NV Network version	Symbol table referenced in the network version.	6
OD Owner default	Master symbol table <i>owner/A</i> . <i>owner</i> is the current owner of the active job. The symbol table <i>owner/A</i> is searched even if there is no symbol table definition on the job level and/or network level.	7
SD System default	Master symbol table SYSDBA/A. The symbol table SYSDBA/A is searched even if there is no symbol table definition on the job level and/or network level.	8



Note: If you invoke the link to the symbol table(s) in the object network, master job, or active job (usually with PF7), you will get a list of all usable **symbol tables** for this object in the above hierarchical order.

Nested Replacement

Within nested replacements, the following rules apply:

- On the top level, only the “original” escape character is checked and replaced.
- If the activation escape is used on a nested level, only activation escapes are replaced on deeper levels. This is necessary to prevent too early replacements of submission escapes.
- If the submission escape is used on a nested level, submission and activation escapes are replaced on deeper levels.

Recursive Symbol Replacement

Symbols can be used within other symbols.

Example:

Symbol	Value
PREFIX	EOR
STEPLIB	\$PREFIX.EXAMPLE.LOAD

\$STEPLIB is replaced by EOR.EXAMPLE.LOAD.



Note: If a point marks the end of a symbol, it is deleted on replacement. With recursive replacement, the deletion takes place replacement by replacement. Thus, a sufficient number of points must be coded. If a point is to remain at the end of the symbol, two points must be coded.

Symbol Replacement in JCL

For symbol replacement in JCL (Job Control Language) submitted on mainframe operating systems (z/OS, z/VSE and BS2000), the following particularities apply:

Column 72 contains one blank and the columns 73 to 80 contain digits.	The columns 71 to 80 of the line are saved before the first replacement, and they are written back after the last replacement.
Column 72 contains a hyphen (-) and the columns 73 to 80 contain blanks.	The columns 71 to 80 of the line are saved before the first replacement, and they are written back after the last replacement.
Columns 71 to 80 contain a comma (,) (8 blanks behind the hyphen)	The columns 71 to 80 of the line are saved before the first replacement, and they are written back after the last replacement.
Intermediate results	Intermediate results can even be longer than 71 characters. They are only overwritten by possibly saved strings at the end of the replacement as a whole.

Import/Export Restrictions

Avoid the equal sign (=) within a symbol value. This causes problems during the import or export of symbol tables.

Escape Character

Symbols to be replaced are detected by a preceding escape character anywhere in the JCL. See [Symbol Escape Characters](#) for details.

Example: Recursive Symbol Replacement

```
/* IN $JJ. SOFTWARE AG
```

changes to the following when J = 20 and JJ = \$J.01:

```
/* IN 2001 SOFTWARE AG
```

The advantage of this method is that symbol replacement is performed within the fixed positions.

Indicator for End of a Symbol Name

Symbol names coded in a source end with one of the following delimiter characters:

blank (), comma (,), semicolon (;),
period (.), apostrophe ('), commercial at (@),
percent sign (%), logical not (¬), ampersand (&),
parenthesis ((or)), or equal sign (=).

Example: \$SYMBOL-1;.

A single period (.) after a symbol is removed during replacement. But two consecutive periods (..) are converted to one period (.). This is important for the dynamic creation of file names. Recursive symbol replacement (symbols within symbols) is also possible.

Symbol Replacement Algorithm

The following algorithm is used for symbol replacement:

1. Search for the symbol farthest to the left.

If none (or no more) available -> end.

2. Replacement

In case of error -> end.

3. Write updated line.

Continue at (1).

This makes recursive symbol replacement (symbols within symbols) possible.

Fixed Positions within a JCL Line



Note: This chapter applies to JCL submitted on mainframes only.

Multiple symbol replacement can cause a shift within a line. However, in the following cases, parts of the line are kept in a fixed position:

1. Symbols do not follow one another directly.
2. The columns 72 to 80 always remain fixed. If column 70 contains a blank and column 71 a comma, then these will also remain fixed.
3. If a symbol is followed by at least two blank spaces, the rest of the line is kept in its original position, unless the symbol value would overlap with it. For example:

```
/FILE ABC.$VAR          COMMENT
```

changes to the following when VAR = REPLACE:

```
/FILE ABC.REPLACE      COMMENT
```

4. If a symbol is followed directly by continuous text, then by at least two blank spaces, then by an exclamation mark (!) and continuous text: the symbol is replaced as in (2) and the exclamation mark (!) is replaced by blank spaces. For example:

```
/FILE ABC.$VAR..XYZ      !COMMENT
```

changes to the following when VAR = REPLACE:

```
/FILE ABC.REPLACE..XYZ    COMMENT
```

Symbol Replacement with Multiple Symbol Values

If you want to replace a symbol with multiple values when a job is activated numerous times concurrently, you must use P-SUFFIX to identify the variable to be replaced. This is the variable you entered in the **Suffix Symbol** field in the **Job Definition (Master)** window (see the field description in the section *Job Maintenance*).

For information on defining multiple symbol values, see the symbol functions **!MV or ?MV and !MM or ?MM** described in the section *Functions for Symbol Replacement*.



Note: If **Suffix Symbol** contains no values, then the multiple job is activated as a temporary dummy job.

You can obtain a specific occurrence of a multiple-value symbol by using the **symbol function !MV or ?MV** delivered with Entire Operations.

Symbol Replacement in Subnetworks

To be able to enter symbols in subnetworks as well, you can attach a symbol table to the main network; this table contains all symbols to be queried in the jobs of the subnetwork(s). At the same time, the symbol tables can be linked to the corresponding jobs of the subnetwork(s) by means of symbols whose values are not to be retrieved. After symbol entry, this creates active symbol tables attached to the main network.

On the job level, Entire Operations tries to load the symbols from the level(s) above if a symbol with the corresponding symbol table is not found, for example from the active symbol table of the main network. Entire Operations does not create active symbol tables on subnetwork job levels as the run numbers are not yet known at the time of the main network's activation and as symbol prompting is activated first.

The following example screens illustrate the procedure:

This screen shows the main network's jobs DUMMY and SUB-11 with symbol table.

```

+-----+-----+-----+-----+-----+-----+-----+-----+-----+ 0
!
!                               Job Definition (Master)                               ! --
!                               !                               ! er
! Job Name      ==> DUMMY_____ Mod ==> GFR      29.10.10 10:10 !
! Description   ==> dummy_____ !
! Job Type      ==> DUM !
! Execution Node ==> 146 MVS/ESA !
! ! !
! Special Type  ==> _ Symbol Table ==> GFR-ST3___ !
! Restartable   ==> N Suffix Symbol ==> _____ !
! ! Escape Characters: Activation ==> $ !
! ! Submit ==> $ !
! ! !
! Enter-PF1---PF2--PF3--PF4---PF5---PF6---PF7-----PF9--PF10---PF12- !
! Help Add End Edit Save Spec Symb JCL Copy Menu !
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
***** Bottom of Data *****
A Depend. C Copy D Delete E Edit G Pregen. I Input Cond. J JCL L Resources
M Modify O EOJ Chk + Act P Description R Activate S Scheduling Parms U Add.Log
Command => _____
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help Add End Save Up Down Menu

```

The symbol table containing symbols to be queried (GFR-ST3) is linked to the dummy job. No symbol tables are attached to the subnetworks themselves.

```

+-----+ 1
!                                     ! 0
!               Job Definition (Master) ! --
!                                     ! er
! Job Name      ==> TESTJOB11_         Mod ==> GFR      29.10.10 10:37 !
! Description   ==> _____         ! 4
! Job Type      ==> JOB                 !
! Execution Node ==> 148 MVS/ESA       !
!                                     !
! Special Type  ==> _                   Symbol Table ==> GFR-ST2___ !
! Restartable   ==> _                   Suffix Symbol ==> _____ !
!                                     Escape Characters:  Activation ==> $ !
!                                     Submit           ==> $ !
!                                     !
! Enter-PF1---PF2--PF3--PF4---PF5---PF6---PF7-----PF9--PF10---PF12- !
!           Help Add End Edit Save Spec Symb      JCL Copy Menu !
+-----+
***** Bottom of Data *****
A Depend. C Copy D Delete E Edit G Pregen. I Input Cond. J JCL L Resources
M Modify O EOJ Chk + Act P Description R Activate S Scheduling Parms U Add.Log
Command => _____
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
           Help Add End Save Up Down Menu

```

On the lowest job level, a symbol table is linked whose symbols are not queried (GFR-ST2). You can make the JCL definition through PF9.

Job definition (master, part 2 (JCL definition)).

```

+-----+ 1
!                               ! 0
+-----+ --
!                               ! er
!           Job: JCL Definition (Master)           !
!                               ! 4
! Job Name      ==> TESTJOB11      Mod ==> GFR      29.10.10 10:38 !
! Description   ==>                               !
! Job Type      ==> JOB                               !
! Execution Node ==> 148 MVS/ESA      Symbol Table ==> GFR-ST2 !
!                               !
! JCL Location  ==> PDS                               !
! JCL Node      ==> 148 MVS/ESA                       !
! File/NatLib   ==> SN.GFR.SOURCE_____ !
! Member        ==> IEFBR14_ _____ !
! VolSer        ==> _____ (if not catlgd) Password ==> !
!                               !
! Enter-PF1-----PF3--PF4---PF5---PF6---PF7---PF8-----PF12- ! **
!           Help      End Edit Save Spec Symb Impo           Menu !
+-----+ g
Command => _____
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
          Help Add End Save Up Down Menu

```

This screen shows the master symbol table GFR-ST2:

```

29.10.10          ***** Entire Operations *****          10:52:29
Owner   GFR          Master Symbol Table GFR-ST2
Network TESTNET10          Run
-----
C Symbol          F P Value          modified by
_ NUM            N A 100000          GFR      29.10.10 10:50
-
-
-
-
-
-
-
-
-
-
-
***** Bottom of Data *****
C Copy   D Delete   M Modify
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
          Help Add End Save Up Down Print

```



```

Edit SN.GFR.SOURCE(IEFBR14)----- ==> Member IEFBR14 saved
====>                                     BLAETTERN====> CSR
***** ***** top of data *****
00001 //IEFBR14 JOB ,GFR,MSGCLASS=$MSGCLASS,CLASS=$CLASS
00002 /* NUM=$NUM
00003 //STEP1 EXEC PGM=IEFBR14
***** ***** bottom of data *****

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

```

This screen shows the active JCL for TESTJOB11.

```

Edit SN.GFR.SOURCE(IEFBR14)----- ==> Member IEFBR14 saved
====>                                     BLAETTERN====> CSR
***** ***** top of data *****
00001 //IEFBR14 JOB ,GFR,MSGCLASS=X,CLASS=G
00002 /* NUM=$NUM
00003 //STEP1 EXEC PGM=IEFBR14
***** ***** bottom of data *****

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

```

69 Functions for Symbol Replacement

- Results Returned by a Symbol Function 682
- Function !D or ?D - Date from Today Calculation 683
- Function !E or ?E - Date from Date Calculation 689
- Functions !MV or ?MV and !MM or ?MM - Access to Multiple-Value Symbols 690
- Function !TIMN or ?TIMN - Constant Time Values 692
- Function !W or ?W - Date Calculation with Reference Calendar 693

A symbol function is used to replace symbols through the JCL of a job based on the conditions defined with the function. The same JCL can be used for different purposes, such as, changing the standard layout and output of symbols, job input conditions or end-of-job actions.

A symbol function is identified by either an exclamation mark (!) or a question mark (?) that is used as a start value. You can choose either character depending on the current code page used in your Natural environment. The exclamation mark (!) can conflict with code page conversion.

You can also perform your own symbol functions with a user exit routine. For details, see [User Exits for Symbol Functions](#) in the section *User Exits*.

The predefined symbol functions provided are explained in this section.

Related Topic:

- [User Exits for Symbol Functions](#)

Results Returned by a Symbol Function

Results of symbol functions are written back to the active job's symbol table after the first evaluation. The following invocations of the same symbol function with the same parameters do not invoke the symbol function again, but just return the function value.

Symbol functions with activation escape character are written to the active symbol table without escape character.

Symbol functions with submit escape character are written to the active symbol table with the submit escape character.



Notes:

1. The same symbol or symbol function can be used with both the activation escape character and the submit escape character. Both are handled separately and occupy different active symbol table entries. This is especially important if you use any time-dependent symbols or symbol functions.
2. Results of symbol functions are not kept in active symbol tables to enforce recalculation of symbol function values, for example, after the change of a multiple-value symbol.

Function !D or ?D - Date from Today Calculation

This symbol function calculates date values from the current date.

This section covers the following topics:

- [Syntax](#)
- [Examples](#)
- [Return Format](#)
- [Parameters](#)
- [Edit Masks for Date Functions](#)
- [Reference Objects](#)
- [Output Date Formats](#)

Syntax

```
$!D|?D<type|period|offset[|offset-unit][,edit-mask][,calendar]>
```

(for all **types** except T)

or

```
$!D|?D<T[,edit-mask]>
```

or

```
$!D|?D<yyyymmdd,edit-mask>
```

Examples

```
$!D<AC-1>
```

Get the date before the current date (yesterday). The date is returned in the default format YYYYMMDD.

```
$?D<CW+1,DDMMYY>
```

Get the first day (Monday) of the current week, in the format DDMMYY.

```
§!D<T,YYMMDD>
```

Get the current date in the format YYMMDD.

```
§!D<20070629,05>
```

Get the date 2007-06-29 in the format 05 (DD.MM.YY). See [Output Date Formats](#).

```
§!D<A1+1,MM>
```

Get the previous month in the format MM (month only).

```
§!D<A1+1,YYYYMMDD>
```

Get the first day of the previous month.

```
§!D<A1-1,YYYYMMDD>
```

Get the last day of the previous month.

Return Format

See [Output Date Formats](#).

Parameters

The parameters are written one after the other without spaces.

Syntax Element	Parameter	Explanation
<i>type</i>	T	Today's date. For this type, the parameters <i>period</i> , <i>offset</i> and <i>offset-unit</i> are not required or evaluated.
	A	All days (365 resp. 366 days a year).
	C	Calendar days (days defined as working days in the calendar).
	S	Schedule days.
<i>period</i>		Not evaluated for the type T.
	C	From current day.
	W	Week.
	M	Month.
	1 to 9	One month back...nine months back.
	Q	Quarter.
	Y	Year.
<i>offset</i>		Positive or negative number with prefix.

Syntax Element	Parameter	Explanation
<i>offset-unit</i>		Unit for the offset (optional); only evaluated for the type A .
	D	Days (default).
	W	Weeks.
	M	Months.
	Q	Quarters.
	Y	Years.
<i>edit-mask</i>		See Edit Masks for Date Functions .
<i>calendar</i>		See Calendar in <i>Reference Objects</i> .
<i>yyyymmdd</i>		Date in the format YYYYMMDD.

For further explanations of date and time formats, see also [Date and Time Formats](#).

Edit Masks for Date Functions

The edit mask can be entered either directly or with a format sign. The format sign can be used to shorten the function call.

Format Sign	Format	Example
F1, 1	DD	29
F2, 2	DDMM	2910
F3, 3	DDMMYY	291008
F4, 4	DDMMYYYY	29102008
F5, 5	DD.MM.YY	29.10.08
F6, 6	DD.MM.YYYY	29.10.2008
F7, 7	DD/MM/YY	29/10/08
F8, 8	DD/MM/YYYY	29/10/2008
F9, 9	DD-MM-YY	29-10-08
F10, 10	DD-MM-YYYY	29-10-2008
F11, 11	MM	10
F12, 12	MMDD	1029
F13, 13	MMYY	1008
F14, 14	MM.YY	10.08
F15, 15	MMYYYY	102008
F16, 16	MM.YYYY	10.2008
F17, 17	NN	We (Weekday; language-dependent)

Format Sign	Format	Example
F18, 18	O	4 (Number of the weekday)
F19, 19	WW	44 (Number of the week)
F20, 20	YY	08
F21, 21	YYJJJ	08245
F22, 22	YYMM	0810
F23, 23	YYMMDD	081029
F24, 24	YYYY	2008
F25, 25	YYYYJJJ	2008245
F26, 26	YYYYMM	200810
F27, 27	YYYYMMDD	20081029
F28, 28	YYYY.MM.DD	2008.10.29
F29, 29	YYYY/MM/DD	2008/10/29
F30, 30	YYYY-MM-DD	2008-10-29
F31, 31	YY.MM.DD	08.10.29
F32, 32	YY/MM/DD	08/10/29
F33, 33	YY-MM-DD	08-10-29
F34, 34	QQ	WE (Weekday; language-dependent; in upper case)

For further explanations of date and time formats, see also [Date and Time Formats](#).

Reference Objects

Syntax Element	Meaning
Owner	The owner of the network from which the function was called.
Calendar	<p>If a calendar was specified explicitly, this is used for calendar calculations. This calendar must exist under the current owner. If it is not found there, it is searched for under the owner SYSDBA. Otherwise, the calendar of the current schedule is used. If no calendar has been defined, all days are treated as working days.</p> <p>Subnetwork considerations</p> <ul style="list-style-type: none"> ■ If the symbol function is invoked from within a subnetwork, the schedule or calendar of the top level (main calling network) is used for schedule or calendar calculation. ■ Schedules or calendars defined in the subnetwork are ignored in such calculations.

Syntax Element	Meaning
	<ul style="list-style-type: none"> It is not necessary to define schedules or calendars in a network used only as a subnetwork, because they are not relevant.

If this parameter is missing or if an unknown format has been used, the date is returned in the format YYYYMMDD (example: 20070629).

Output Date Formats

The output date formats described here can be used in the symbol functions !D or ?D and !W or ?W.

The default date output format is YYYYMMDD (example: 20081029).

Format	Numeric Equivalent	Example
DD	01	29
DDMM	02	2910
DDMMYY	03	291008
DDMMYYYY	04	29102008
DD.MM.YY DD#MM#YY	05	29.10.08
DD.MM.YYYY DD#MM#YYYY	06	29.10.2008
DD/MM/YY	07	29/10/08
DD/MM/YYYY	08	29/10/2008
DD-MM-YY	09	29-10-08
DD-MM-YYYY	10	29-10-2008
JJJ	38	245
JJYY	39	24508
JJYYYY	40	2452008
MM	11	10
MMDD	12	1029
MMDDYY	35	102908
MMYY	13	1008
MM.YY MM#YY	14	10.08
MMYYYY	15	102008

Format	Numeric Equivalent	Example
MM.YYYY MM#YYYY	16	10.2008
MM/DD/YY	36	10/29/08
MM-DD-YY	37	10-29-08
NN	17	Sa (First 2 characters of name of weekday. This value depends on the current language setting of the Natural session.)
O	18	6 (Number of weekday. This value depends on the Natural profile parameter DTFORM of the Natural session. Refer to the Natural <i>Parameter Reference</i> documentation.)
QQ	34	SA (Like format NN, but in upper case)
WW	19	44 (Number of week)
WWYY	41	4408 (Week and year)
WWYYYY	42	442008 (Week and year)
YY	20	08
YYJJJ	21	08245
YYMM	22	0810
YYMMDD	23	081029
YYWW	43	0844 (Year and week)
YYYY	24	2008
YYYYJJJ	25	2008245
YYYYMM	26	200810 (Year and month)
YYYYMMDD	27	20081029
YYYYWW	44	200844 (Year and week)
YYYY.MM.DD YYYY#MM#DD	28	2008.10.29

Format	Numeric Equivalent	Example
YYYY/MM/DD	29	2008/10/29
YYYY-MM-DD	30	2008-10-29
YY.MM YY#MM	45	08.10
YY.MM.DD YY#MM#DD	31	08.10.29
YY/MM/DD	32	08/10/29
YY-MM-DD	33	08-10-29

For further explanations of date and time formats, see also [Date and Time Formats](#).

Function !E or ?E - Date from Date Calculation

This symbol function is used to add or subtract a number of days from a given date in the format YYYYMMDD. See also [Date and Time Formats](#).

This section covers the following topics:

- [Syntax](#)
- [Examples](#)
- [Return Format](#)
- [Parameters](#)

Syntax

```
$!E|?E<date,offset[,edit-mask]>
```

Examples

```
$!E<20081029,+28>
```

`§!E<20081029,-7,F10>`

Return Format

Date in the format YYYYMMDD. See also [Date and Time Formats](#).

Parameters

The parameters are separated by a comma.

Syntax Element	Description
<i>date</i>	Date in the format YYYYMMDD. See also Date and Time Formats .
<i>offset</i>	Positive or negative number with a plus (+) or minus (-) sign. Unit: days.
<i>edit-mask</i>	See Edit Masks for Date Functions .

Functions !MV or ?MV and !MM or ?MM - Access to Multiple-Value Symbols

These symbol functions allow you to use multiple symbol values.

You can get the maximum index of a multiple-value symbol.

This section covers the following topics:

- [Symbol Function !MV or ?MV](#)
- [Symbol Function !MM or ?MM](#)

Symbol Function !MV or ?MV

This symbol function allows you to use one single value of a multiple-value symbol.

Syntax

`§!MV | ?MV<symbol, index>`

Examples

```
$!MV<ARRAY,1>
```

```
$?MV<ARRAY,$INDEX>
```

Return Format

Integer value.

Parameters

The parameters are separated by a comma.

Syntax Element	Description
<i>symbol</i>	Name of the multiple-value symbol.
<i>index</i>	Index of the symbol value.

Errors

In the following cases, Symbol not found appears:

- Multiple symbol not found,
- Index missing,
- Index not numeric, and
- Index too high.

Reference Objects

Syntax Element	Description
Symbol table	The symbol table of the active job is always used.

Symbol Function !MM or ?MM

This symbol function allows you to get the maximum index of a multiple-value symbol.

This function accesses single occurrences of a multiple-value symbol.

Syntax

```
$!MM | ?MM<symbol>
```

Example

```
§!MM<ARRAY>
```

Return Format

Symbol value.

Parameters

This function has only one parameter.

Syntax Element	Description
<i>symbol</i>	Name of the multiple-value symbol.

Errors

In the following case, `Symbol not found` appears:

- Multiple symbol not found.

Reference Objects

Syntax Element	Description
Symbol table	The symbol table of the active job is always used.

Function !TIMN or ?TIMN - Constant Time Values

This symbol function allows you to use constant time values in multiple timestamp replacements.

This section covers the following topics:

- [Syntax](#)
- [Examples](#)
- [Return Format](#)
- [Parameters](#)

- [Reference Objects](#)

Syntax

```
$!TIMN|?TIMN<parm>
```

Examples

```
$!TIMN<A>
```

```
$?TIMN<B>
```

Return Format

A time stamp in the format of the Natural system variable *TIMN:

Format: A7,

Content: HHIISST where T = tenths of seconds.

Parameters

Parameter	Explanation
<i>parm</i>	The parameter is a dummy. But note that all invocations with the same parameter create the same time value (if the same active symbol table is used).

Reference Objects

None.

Function !W or ?W - Date Calculation with Reference Calendar

This symbol function allows you to use date calculations relative to a calendar.



Note: If the formula has an (even intermediate) result pointing to the 29th February of a non-leap-year, the resolution of the symbol function !W or ?W will be stopped with an error message.

This section covers the following topics:

- [Main Syntax](#)

- Parameter
- Return Format

Main Syntax

```

§!W|?W< formula>
[,<edit-mask>]
[,<owner>]
[,<calendar>]
    
```

Symbol containing four arguments as values. This symbol itself can be composed of symbols. The activation or submission escape character of the invoking active job is in effect.

Syntax of *formula*:

```

<date><period>
<point><just-in-period><offset>
<unit><point>
<just-in-unit>
    
```

Syntax of *edit-mask*:

```

<date-format>
<number-unit><period><supression><just>
<name-unit><language><character><length>
    
```

Parameter

Syntax Element	Possible Values
<i>formula</i>	
<i>date</i>	<p>Reference date in the valid Natural date format:</p> <p>YYYYMMDD, YYMMDD, DDMMYYYY or DDMMYY.</p> <p>MM, DD, [YY]YY can be separated by a slash (/), a dash (-) or a period (.). For further explanations of date formats, see also Date and Time Formats.</p> <p>Note: To avoid ambiguous date format interpretations, it is highly recommended to use a 4-digit year format. If a 6-digit date cannot be interpreted uniquely, the symbol function returns with an error message like: 040107A / MM is ambiguous.</p>

Syntax Element	Possible Values	
<i>period</i>	A	Days.
	C	Working days.
	W	Week.
	M	Month.
	Q	Quarter.
	S	Semester.
	Y	Year.
The limits of the date period are only left if $\langle offset \rangle * \langle unit \rangle$ is longer (or equal to) the period. If $\langle offset \rangle * \langle unit \rangle$ is shorter than the period, but would not match the beginning or end of the period, the first or the last (working) day of the period is returned.		
<i>point</i>	F	First day in period or unit.
	L	Last day in period or unit.
	S	Calculated date in period or unit.
<i>just-in-period</i>	B	If holiday, before.
	A	If holiday, after.
<i>offset</i>	Integer preceded by sign.	
<i>unit</i>	Like <i>period</i> .	
<i>just-in-unit</i>		
<i>edit-mask</i>		
<i>date-format</i>	Date in the format D. See also Output Date Formats described for the symbol function !D or ?D.	
<i>number-unit</i>	Number in the format N. Number of the <i>unit</i> in the <i>period</i> . This implies that the specified period must be larger than the unit. Same values as for <i>period</i> .	
<i>period</i>	Same values as for <i>period</i> .	
<i>suppression</i>	Z for suppression of leading zeroes.	
<i>just</i>	R	Right justified.
	L	Left justified.
<i>name-format</i>	Name in the format C.	
<i>name-unit</i>	D	Day.
	M	Month.
<i>language</i>	1	English.
	2	German.

Syntax Element	Possible Values	
	<i>nn</i>	Valid Natural language code.
<i>character</i>	U	Name only in upper-case letters.
	L	Only first character in upper case.
<i>length</i>	Display length.	
<i>owner</i>	Owner of the reference calendar.	
<i>calendar</i>	Reference calendar for date calculation. If the <i>formula</i> contains a calendar reference such as C for working day, B for working day before holiday or working day after holiday, the specification of a reference calendar is mandatory.	

Return Format

See [Output Date Formats](#) described for the symbol function !D or ?D.

XIV

Reporting

70 Reporting

▪ Report Types	700
▪ Generating Online Reports	702
▪ Fields and Columns: Reports	708
▪ Special PF Keys: Reports	711
▪ Determination Date for Report Data	712
▪ Report Output Options	712
▪ User Restrictions for Reports	712
▪ Examples of Reports	713
▪ Generating Batch Reports	734

This section describes the **Reports** function which is used to generate reports to overview your Entire Operations environment.

Related Topic:

- See also the cross reference reports described in the section [Cross References](#).

Report Types

The report types available you can select from the [Reports References menu](#) are described in the following table. Reports that require special user privileges are indicated in the table.

Report Type	Description
Log - reports (see below)	<p>The LOG - reports (Reports 1 through 5) provide job or network processing information for a given date/time range, extracted from the Entire Operations log. The following applies to all job logs:</p> <p>Only those networks are displayed for which the user has read access. The user has read access if:</p> <ul style="list-style-type: none"> ■ the user has profile type A (Administrator); ■ the owner SYSDBA is currently assigned to the user; ■ the network owner can be found on the user's owner list; ■ access to the network has been explicitly granted to the user (see Granting Definition: Authorizing Other Users or Owners to Access a Network in the section <i>Network Maintenance</i>).
Log - Terminated Jobs	<p>Lists all jobs that ended normally (OK).</p> <p>See also Example of Log - Terminated Jobs.</p>
Log - Abended Jobs	<p>Lists all jobs that ended abnormally (not OK).</p> <p>See also Example of Log - Abended Jobs.</p>
Log - Jobs not started	<p>Lists all jobs that were not started.</p> <p>A job cannot start, for example, if its latest start time exceeded, or if it is waiting for an input condition or a resource.</p> <p>See also Example of Log - Jobs not started.</p>
Log - Jobs with permanent errors	<p>Lists all jobs that fail to run because of permanent errors.</p> <p>See also Example of Log - Jobs with permanent errors.</p>

Report Type	Description
Log - Networks not activated	<p>Lists all networks that could not be activated, because an extraction or activation error occurred.</p> <p>See also Example of Log - Networks not activated.</p>
Accounting Data	<p>Information on accounting data (for example, job elapsed times and CPU times) of previous network and job executions.</p> <p>See also Example of Accounting Data.</p>
Network Description (short)	<p>Displays information on networks and jobs as defined on the master database, including scheduling information, prerequisites and end-of-job checking and actions.</p> <p>See also Example of Network Description (short).</p>
Network Description (detailed)	<p>Displays the same information as the Network Description (short), but includes all prose descriptions defined at the network, job or event level using the Editor facility.</p> <p>See also Example of Network Description (detailed and with JCL).</p>
Network Description (with JCL)	<p>Displays the same information as Network Description (detailed) and, additionally, outputs the JCL of each job with JCL location BS2, LMS, NAT, PDS, LIB or VSE.</p> <p>Note: For security reasons, no Entire System Server logons are performed. This means that the user must already have logged on to each node containing JCL (or the node must be started with the AUTOLOG=YES parameter).</p> <p>See also Example of Network Description (detailed and with JCL).</p>
Network Job Flow Display	<p>Provides an overview of the job flow within a network. The output can be sent to the screen or printer.</p> <p>See also Example of Network Job Flow Display.</p>
Schedule of Jobs	<p>Displays a jobs schedule for a specific date range.</p> <p>See also Example of Schedule of Jobs.</p>
Network Start Summary	<p>Status report of all network starts for a given day, regardless of whether they:</p> <ul style="list-style-type: none"> ■ are waiting for prerequisite resources; ■ have already been executed; ■ are currently being executed; ■ have already been completed. <p>See also Example of Network Start Summary.</p>

Report Type	Description
Network Schedule Overview	Overview of scheduled and/or not yet executed, system-wide network activations. See also Example of Network Start Summary .
Activation Overview	Overview of network activations. See also Example of Activation Overview .
Compare Symbol Tables	(Special user authorization required.) Displays the results of the comparison of one or more symbol tables. See also Example of Compare Symbol Tables .
Compare Networks	(Special user authorization required.) Displays the results of the comparison of one or more networks. The report shows whether jobs defined in one network exist also in another network. Additionally the attributes of the networks and of all jobs having the same name are compared. To generate a report in batch, see Compare Networks in Batch for further information. See also Example of Compare Networks .
Node Overview	(Special user authorization required.) Overview of available nodes. See also Example of Node Overview .
Network/Job Usage	(Special user authorization required.) Lists networks and related subnetworks (jobs of the type NET) as well as jobs activated by an end-of-job action or a recovery job. See also Example of Network/Job Usage .

Generating Online Reports

This section provides instructions for generating a report in online mode with the **Reports** function.

Report data is evaluated for the current day.



Note: The selection options provided to generate a report allow extremely flexible reporting. On the other hand, it allows you to formulate requests that will cause very many Adabas calls. For this reason, it is possible to disallow asterisk/wildcard report selection for non-administrator users: see [Range Specification and Usage Restrictions](#).

➤ **To generate a report**

- 1 From the **Main Menu**, choose **Reports** and press ENTER.

Or:

In the **Command** field of the **Main Menu**, type Option Code 8 or type REPORTS (see *Direct Commands*) and press ENTER.

Or:

You can skip this step if you know the number of the report type (for example, **14** for **Activation Overview**) you want to generate:

In the **Command** field of the **Main Menu**, type Option Code 8.14 and press ENTER.

A **Reports** menu similar to the example below appears:

```

15.11.16          ***** Entire Operations *****          11:49:51
Owner EXAMPLE          Reports          User ID SAG
-----
          Reports          Reports
1  Log - Terminated Jobs          15  Compare Symbol Tables
2  Log - Abended Jobs              16  Compare Networks
3  Log - Jobs not started          17  Node Overview
4  Log - Jobs with permanent errors 18  Network/Job Usage
5  Log - Networks not activated
6  Accounting Data
7  Network Description (short)
8  Network Description (detailed)
9  Network Description (with JCL)
10 Network Job Flow Display
11 Schedule of Jobs
12 Network Start Summary
13 Network Schedule Overview
14 Activation Overview
Command => _____

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
          Help          End          Menu

```

The menu displays a list of selectable **report types**.

- 2 Choose a report and press ENTER.

Depending on the report type selected, a selection window similar to the examples below opens for

Report types 1 through 12:

```

+-----+
!
! Log - Abended Jobs
!
! Owner.....: _____ (Blank for all, * for selection
! Network....: _____ or * and PF4 to process all
! Version....: _____ matching objects)
! Job.....: _____
!
! Destination: 1
! 1 - Screen 2 - Printer 3 - PC 4 - PC-CSV
!
! Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---
! Help End Apply Accpt
+-----+
    
```

Report type 13 (Network Schedule Overview):

```

+-----+
|
| Network Schedule Overview: Type
|
| A Extractions + Schedule, sorted by network, time
| B Extractions only, sorted by network, time
| C Extractions only, sorted by time
| D Extractions + Schedule, sorted by time, network
|
| Type ==> A
|
| Enter-----PF3-----
| End
+-----+
    
```

Reports type 14 (Activation Overview):

```

+-----+
|                                     |
|           Activation Overview: Type |
|                                     |
| Select one or more of these Activation Types: |
| * All types                A API |
| M Manual                  O EOJ |
| R Recovery                 S Scheduled |
|                                     |
| Type ==> * _____ |
|                                     |
| Enter-----PF3----- |
|                               End |
|                                     |
+-----+

```

Report type 15 (Compare Symbol Tables):

```

+-----+
|                                     |
| Compare Symbol Tables |
|                                     |
| 1                               2 |
| Owner.....: NOPALL____ | Owner.....: NOPALL____ |
| Table.....: _____ | Table.....: _____ |
| Version.....: _____ | Version.....: _____ |
|                                     |
| Show.....: D |
|   A - All    D - Differences  M - Message only |
| Destination: 1 |
|   1 - Screen 2 - Printer 3 - PC |
|                                     |
| Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9--- |
|   Help      End   Apply Acpt |
|                                     |
+-----+

```

Report type 16 (Compare Networks):

```
Compare Networks

1                               2
Owner.....: NOPALL_____  Owner.....: NOPALL_____
Network .... _____      Network .... _____
Version.....: _____      Version.....: _____

Show.....: D
  A - All      D - Differences  M - Message only
Destination: 1
  1 - Screen  2 - Printer  3 - PC

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---
      Help      End      Apply Accpt
```

Report type 17 (Node Overview):

```
Node Overview

Node Range
from..: ____1
to....: 99900

Destination: 1
  1 - Screen  2 - Printer  3 - PC  4 - PC-CSV

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8-
      Help      End                                  Down
```

Report type 18 (Network/Job Usage):

```

+-----+
|
| Network/Job Usage
|
| Used Job                               Using Job
| Owner.....: EXAMPLE____              Owner.....: EXAMPLE____
| Network.....: _____              Network.....: _____
| Version.....: _____              Version.....: _____
| Job.....: _____                  Job.....: _____
|
| Destination: 1
|   1 - Screen  2 - Printer  3 - PC
|
| Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---
|   Help       End   Apply Acpt
|
+-----+

```

- 3 Enter the required selection criteria and press PF4 or PF5, or ENTER for the report types **13**, **14** and **17**.

An additional selection window similar to the example below opens for

Report types **1** through **6**:

```

+-----+
|
| Log - Abended Jobs
|       Date / Run Number Selection
|
| From Date 09.06.15  00:00   to 09.06.15  23:59
| From Run  1_____         to 99999
|
| PF3  End
|
+-----+

```

Report types **11** and **12**:

```

+-----+
|
| Schedule of Jobs
|
| Start Date  ===> 09.06.15
|
| End Date    ===> 09.06.15
|
| Enter                               PF3 End
|
+-----+

```

Fields and Columns: Reports

The input fields for specifying selection criteria and output options for the report to be produced are explained in the following table. The input fields correspond to the columns contained on the report output screen.

Depending on the report type (to be) produced, the following fields and columns are available:

Field/Column	Description
Owner	Name of an owner or a range of names .
Network	Name of a network or a range of names .
Network Version	Name of a network version or a range of names .
Job	(Applies to Log - reports and Job Schedule only) Name of a job or a range of names . If no job name is defined, the corresponding output column on the report screen contains a dash (-).
From Date ... to or Start Date/End Date	Start date/time and end date/time of the reporting period. Valid number range: 1 through 31. Default: current date and time range 00:00 - 23:59. See also Date and Time Formats . Note: For Network Schedule Overview reports, the start and end dates can be in the past. In this case, only those network activations are logged, which have not yet been executed. These can be caused by symbol prompting, which has not been completed, or by an Entire Operations Monitor, that has not been active over an extended period of time.
From Run ... to	Start and end run numbers for a range of active job networks. Can be used only if owner and network have been selected. Default range: 1 - 9999 (all)
Destination	See Report Output Options .
Type	(Applies to Activation Overview only) Type of network activation. <hr/> Valid values: <hr/> * All types of activations.

Field/Column	Description	
M	Activated manually.	
R	Activated by End-of-Job recovery processing.	
A	Activated by the activation API. NOPUAC5N (see the section <i>API Routines</i>).	
O	Activated by an End-of-Job action.	
S	Activated by a schedule.	
Compare Networks/Symbol Tables reports only:		
Show	Determines the amount of information to be shown for the compared networks.	
	Valid values:	
	A	Shows all matching and differing objects and attributes.
	D	Shows only the differing objects with the differing attributes (default).
M	Shows only a message if the comparison detected any differences.	
Accounting Data/Schedule of Jobs reports only:		
Step	(Applies to z/OS only) Job step .	
JobID	Job identifier as assigned by the operating system.	
Start	Start date/time of the job. See also Date and Time Formats .	
Stop	End date/time of job. See also Date and Time Formats .	
Elapsed	Estimated run time (in minutes) of the job.	
CPU Tm	Estimated CPU time (in seconds) of the job.	
Averages from/to	Average values of all elapsed and CPU times for all jobs in the network.	
Network/Job Usage reports only:		

Field/Column	Description						
Used Job	Networks and jobs used by single or multiple jobs.						
Using Job	Networks and jobs using a job as a subnetwork (job type NET), an end-of-job (EOJ) action or a recovery job.						
	Usage indicates the job use: Subnet, EOJ Activation or Recovery.						
Node Overview reports only:							
Server Name	For mainframe nodes: a descriptive name For UNIX and Windows nodes: the name of the EntireX Broker service						
Node Range	A range of node numbers from 1 to 99900.						
Short	Short name of a node						
AM	Access mode used: <table border="1" data-bbox="397 751 1385 982"> <tr> <td>N</td> <td>Mainframe nodes accessed through Entire Net-Work</td> </tr> <tr> <td>B</td> <td>UNIX and Windows nodes accessed through EntireX Broker.</td> </tr> <tr> <td>L</td> <td>Applies to UNIX and Windows only Local node, invoked directly on the machine where Entire Operations is running.</td> </tr> </table>	N	Mainframe nodes accessed through Entire Net-Work	B	UNIX and Windows nodes accessed through EntireX Broker.	L	Applies to UNIX and Windows only Local node, invoked directly on the machine where Entire Operations is running.
N	Mainframe nodes accessed through Entire Net-Work						
B	UNIX and Windows nodes accessed through EntireX Broker.						
L	Applies to UNIX and Windows only Local node, invoked directly on the machine where Entire Operations is running.						
Op. Sys.	Operating system under which the node is running as received from the last SYSTEM-INFO call to Entire System Server or UNIX/Windows system information.						
Wait a. Error	Wait time after error. Time in minutes to wait until next node access after a temporary error. See also Date and Time Formats .						
VSE SysId	The SYSID defined for a z/VSE node is added to the job card of jobs submitted on this node.						
SSU	Submit Security User Type. If empty, the system-wide default is in effect for this node.						
Time Diff.	Time difference between local time and GMT in hours if node is in a different time zone.						
Valid	Indicates whether a node is available: <table border="1" data-bbox="397 1501 917 1633"> <tr> <td></td> <td></td> </tr> <tr> <td>yes</td> <td>Node can be used</td> </tr> <tr> <td>no</td> <td>Node has been disabled</td> </tr> </table>			yes	Node can be used	no	Node has been disabled
yes	Node can be used						
no	Node has been disabled						
NPR Version	Current Entire System Server version of the node						
OS Release	Information about the operating system (where available)						

Range Specification and Usage Restrictions

If indicated in the table above, you can use an asterisk (*) or a smaller/greater than (< or >) wildcard character to determine a start or an end value (see also *Valid Name Specifications* in *Direct Commands*) or open a selection list of names.

The use of asterisk/wildcard report selection can be disallowed for non-administrator users (see *User Definitions and Profile Settings* in the *Administration* documentation).

By default, users of type A (System-Administrator) and O (Operator) are allowed to use asterisk/wildcard selection; users of type G (regular user) are not. It is not possible to disallow it for users of type A.

Special PF Keys: Reports

The following PF keys can be used to execute a function in a report selection window or navigate in a report output screen:

PF Key	Name	Function
PF3	End	Return to Reports or Cross References menu, depending on the function performed.
PF4	Apply	Confirm selection criteria, including "*" wildcards. See also Confirming Selections .
PF5	Accpt	Confirm selection criteria. If any object name has an *, you will be prompted with a list of matching objects, from which you may select one and press PF5 again. See also Confirming Selections .
PF7	Up	Scroll up in a report screen.
PF8 (or ENTER key)	Down	Scroll down in a report screen.
PF10	Left	Scroll left in a report screen.
PF11	Right	Scroll right in a report screen.
PF12	Menu	Return to Main Menu .

Confirming Selections

When you are satisfied with your selection, press PF5 to move on to the date/time/run range selection window (where applicable) or to generate the report. However, if you press PF4 rather than PF5, you will not be prompted - instead the report will include all objects which match your selection criteria.

Determination Date for Report Data

Report data is evaluated for the current day. This evaluation date needs to be considered, in particular, when you maintain different versions of networks and symbol tables with either fix versions or versions dynamically loaded on this date.

Report Output Options

The **Destination** field specifies the format for the report and the location where the report is written.

Enter one of the following option codes to select the required destination:

Option	Description
1	The report will be written to the screen (default).
2	The report will be written to Printer 1 (which must be defined appropriately).
3	The report will be written to the PC (using Entire Connection and Printer 7).
4	The report will be written to a PC work file (using Entire Connection and Work File 7) in CSV (comma-separated value) file format, suitable for importing into a spreadsheet package. Note: This destination is not available for all reports.

User Restrictions for Reports

Administrator users (and reports run in batch) may report on any network in any owner. Non-administrator users, however, will only receive information on networks to which they have read access. A user is considered to have read access if:

1. The user is linked to owner SYSDBA.
2. The user is linked to the Network owner.
3. Access to the network has been explicitly granted to the user.

Examples of Reports

This section contains examples of all types of reports you can generate with the **Reports** function:

- Example of Log - Terminated Jobs
- Example of Log - Abended Jobs
- Example of Log - Jobs not started
- Example of Log - Jobs with permanent errors
- Example of Log – Networks not activated
- Example of Accounting Data
- Example of Network Description (short)
- Example of Network Description (detailed and with JCL)
- Example of Network Job Flow Display
- Example of Schedule of Jobs
- Example of Network Start Summary
- Example of Network Schedule Overview
- Example of Activation Overview
- Example of Compare Symbol Tables
- Example of Compare Networks
- Example of Node Overview
- Example of Network/Job Usage

Example of Log - Terminated Jobs

The following is an example of the report type **Log - Terminated Jobs**:

Entire Operations Log Report						
Owner	Network	Job	Run	Date	Time	Message
EXAMPLE	E60-FLOW	JOB-01	3581	31.05.12	13:15	Ended ok
EXAMPLE	E60-FLOW	J07	3581	31.05.12	13:18	Ended ok
EXAMPLE	E60-FLOW	JOB-012	3581	31.05.12	13:33	Ended ok
EXAMPLE	E60-FLOW	JOB-013	3581	31.05.12	13:50	Ended ok
EXAMPLE	E60-FLOW	JOB-014	3581	31.05.12	14:07	Ended ok
EXAMPLE	E60-FLOW	JOB-015	3581	31.05.12	14:08	Dummy Job terminated
EXAMPLE	E52-LOGRES	E52-J01	3974	01.06.12	01:01	Ended ok
EXAMPLE	E60-FLOW	JOB-019	3581	01.06.12	01:16	Ended ok
EXAMPLE	E60-FLOW	JOB-01	3582	01.06.12	13:16	Ended ok
EXAMPLE	E60-FLOW	J07	3582	01.06.12	13:18	Ended ok
EXAMPLE	E60-FLOW	JOB-012	3582	01.06.12	13:24	Ended ok
EXAMPLE	E60-FLOW	JOB-013	3582	01.06.12	13:32	Ended ok
EXAMPLE	E60-FLOW	JOB-014	3582	01.06.12	13:39	Ended ok
***** m o r e *****						
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---						
End			Down		Left Right Menu	

This report lists all terminated jobs for the network E60-FLOW in owner EXAMPLE.

The columns contained in the report are described in [Fields and Columns: Reporting](#).

The PF keys are explained in [Special PF Keys: Reports](#).

Example of Log - Abended Jobs

The following is an example of the report type **Log - Abended Jobs**:

Entire Operations Log Report						
Owner	Network	Job	Run	Date	Time	Message
EXAMPLE	E01-CONTI	E01-J02	58	05.06.12	16:38	Ended not ok
EXAMPLE	E01-CONTI	E01-J02	58	05.06.12	16:38	Ended not ok - STEP02 C
EXAMPLE	E01-CONTI	E01-J03	58	05.06.12	16:38	Ended not ok
EXAMPLE	E01-CONTI	E01-J03	58	05.06.12	16:38	Ended not ok - STEP03 S
EXAMPLE	E01-CONTI	E01-J04	58	05.06.12	16:38	Ended not ok
EXAMPLE	E01-CONTI	E01-J04	58	05.06.12	16:38	Ended not ok - STEP03 C
EXAMPLE	E01-CONTI	E01-J06	58	05.06.12	16:39	Ended not ok
EXAMPLE	E01-CONTI	E01-J06	58	05.06.12	16:39	Ended not ok - STEP06 U
EXAMPLE	E01-CONTI	E01-J02	59	05.06.12	16:58	Ended not ok
EXAMPLE	E01-CONTI	E01-J02	59	05.06.12	16:58	Ended not ok - STEP02 C
EXAMPLE	E01-CONTI	E01-J03	59	05.06.12	16:58	Ended not ok
EXAMPLE	E01-CONTI	E01-J03	59	05.06.12	16:58	Ended not ok - STEP03 S
EXAMPLE	E01-CONTI	E01-J04	59	05.06.12	16:58	Ended not ok
***** m o r e *****						
Enter	PF1	PF2	PF3	PF4	PF5	PF6
		End			Down	Left
						Right Menu

This report displays all abended jobs for all networks beginning with E01-CONTI of the owner EXAMPLE.

The columns contained in the report are described in [Fields and Columns: Reporting](#).

The PF keys are explained in [Special PF Keys: Reports](#).

Example of Log - Jobs not started

The following is an example of the report type **Log - Jobs not started**:

Entire Operations Log Report						
Owner	Network	Job	Run	Date	Time	Message
EXAMPLE	E60-FLOW	JOB-012	3596	21.06.12	00:00	E60-JOB1-0 - 3596 - RUN
EXAMPLE	E60-FLOW	JOB-013	3596	21.06.12	00:00	E60-J012-0 - 3596 - RUN
EXAMPLE	E60-FLOW	JOB-014	3596	21.06.12	00:00	E60-J013-0 - 3596 - RUN
EXAMPLE	E60-FLOW	JOB-015	3596	21.06.12	00:00	E60-J014-0 - 3596 - RUN
EXAMPLE	E60-FLOW	JOB-03	3596	21.06.12	00:00	E60-JOB2-01 - 3596 - RU
EXAMPLE	E60-FLOW	JOB-04	3596	21.06.12	00:00	E60-JOB3-0 - 3596 - RUN
EXAMPLE	E60-FLOW	JOB-05	3596	21.06.12	00:00	E60-JOB4-0 - 3596 - RUN
EXAMPLE	E60-FLOW	J07	3596	21.06.12	00:00	E60-JOB1-0 - 3596 - RUN
EXAMPLE	E52-LOGRES	E52-J21	3993	21.06.12	08:00	E52-J01-OK - 3993 - RUN
EXAMPLE	E52-LOGRES	E52-J22	3993	21.06.12	08:00	E52-J01-OK - 3993 - RUN
EXAMPLE	E52-LOGRES	E52-J23	3993	21.06.12	08:00	E52-J01-OK - 3993 - RUN
EXAMPLE	E52-LOGRES	E52-J24	3993	21.06.12	08:00	E52-J01-OK - 3993 - RUN
EXAMPLE	E52-LOGRES	E52-J25	3993	21.06.12	08:00	E52-J01-OK - 3993 - RUN
***** m o r e *****						
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---						
End			Down		Left Right Menu	

This report lists all jobs that were not started for all networks beginning from E60-FLOW of the owner EXAMPLE.

The columns contained in the report are described in [Fields and Columns: Reporting](#).

The PF keys are explained in [Special PF Keys: Reports](#).

Example of Log - Jobs with permanent errors

The following is an example of the report type **Log - Jobs with permanent errors**:

Entire Operations Log Report						
Owner	Network	Job	Run	Date	Time	Message
XSETIDZ	K1	KFZ-AA01	818	15.06.12	15:44	MACRO Programm RZF-0001
XSETIDZ	K1	KFZ-AA01	818	15.06.12	15:44	... NOP-JCL/RZF-0001
XSETIDZ	K1	KFZ-AA02	818	15.06.12	15:44	Time Frame Setting
XSETIDZ	K1	KFZ-AA03	818	15.06.12	15:44	Time Frame Setting
XSETIDZ	K1	KFZENDE	818	15.06.12	15:44	MACRO Programm RZFENDE
XSETIDZ	K1	KFZENDE	818	15.06.12	15:44	... NOP-JCL/RZFENDE
XSETIDZ	K1	KFZ20F15J	818	15.06.12	15:44	Subnetwork not found
XSETIDZ	K1	KFZ20F15J	818	15.06.12	15:44	... RZF/KFZ20F15J
XSETIDZ	K1	KFZ30F15H	818	15.06.12	15:44	Subnetwork not found
XSETIDZ	K1	KFZ30F15H	818	15.06.12	15:44	... RZF/KFZ30F15H
XSETIDZ	K1	KFZ40F15CV	818	15.06.12	15:44	Subnetwork not found
XSETIDZ	K1	KFZ40F15CV	818	15.06.12	15:44	... RZF/KFZ40F15CV
XSETIDZ	K1	KFZ50F15R	818	15.06.12	15:44	Subnetwork not found
***** m o r e *****						
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---						
End			Down		Left Right Menu	

This report lists all jobs not started for all networks beginning with K1 of the owner XSETIDZ, and indicates the reason why the jobs failed to start.

The columns contained in the report are described in [Fields and Columns: Reporting](#).

The PF keys are explained in [Special PF Keys: Reports](#).

Example of Log – Networks not activated

The following is an example of a **Log – Networks not activated** report:

Entire Operations Log Report									
Owner	Network	Job	Run	Date	Time	Message			
REQUEST	P230426	-	11065	15.06.12	15:42	Latest	Start	14.06	04:2
REQUEST	P230426	-	11066	15.06.12	15:42	Latest	Start	14.06	05:4
REQUEST	P230426	-	11067	15.06.12	15:42	Latest	Start	14.06	07:0
REQUEST	P230426	-	11068	15.06.12	15:42	Latest	Start	14.06	08:2
REQUEST	P230426	-	11069	15.06.12	15:42	Latest	Start	14.06	09:4
REQUEST	P230426	-	11070	15.06.12	15:42	Latest	Start	14.06	11:0
REQUEST	P230426	-	11071	15.06.12	15:42	Latest	Start	14.06	12:2
REQUEST	P230426	-	11072	15.06.12	15:42	Latest	Start	14.06	13:4
REQUEST	P230426	-	11073	15.06.12	15:42	Latest	Start	14.06	15:0
SN	KSM-001	-	48	15.06.12	15:42	Latest	Start	14.06	22:1
EXAMPLE	E52-LOGRES	-	3988	15.06.12	15:42	Latest	Start	14.06	12:0
NATQA	BS2411.2	-	1237	15.06.12	15:42	Latest	Start	14.06	00:3
NATQA	TEST4-9	-	1770	15.06.12	15:42	Latest	Start	14.06	03:2
***** m o r e *****									
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---									
End					Down		Left Right Menu		

This report lists all jobs not started for networks beginning from P230426 in owner REQUEST.

The columns contained in the report are described in [Fields and Columns: Reporting](#).

The PF keys are explained in [Special PF Keys: Reports](#).

Example of Accounting Data

The following is an example of the report type **Accounting Data**:

```

30.06.12          ***** Entire Operations *****          11:06:37
Owner  EXAMPLE          Job Accounting Data          Network E52-LOGRES
-----
Job      Run Step JobId Start          Stop          Elapsed min CPU Tm sec
E52-J01  3991      31342 20.06 01:00:41 20.06 01:00:51      0.16      0.01
(Network) 3991          01:00:41      01:00:51      0.16      0.01

E52-J01  3992 STEP01      21.06 01:00:00 21.06 01:00:00          0.01
E52-J01  3992      40659      01:00:30      01:00:41      0.18      0.01
(Network) 3992          01:00:30      01:00:41      0.18      0.01

E52-J01  3994 STEP01      23.06 01:00:00 23.06 01:00:00          0.01
E52-J01  3994      60427      01:00:45      01:00:56      0.18      0.01
(Network) 3994          01:00:45      01:00:56      0.18      0.01

E52-J01  3995 STEP01      24.06 01:00:00 24.06 01:00:00          0.01
E52-J01  3995      70076      01:00:22      01:00:32      0.16      0.01
(Network) 3995          01:00:22      01:00:32      0.16      0.01
-----
Averages from 20.06.12 01:00 to 24.06.12 01:00 are          0.00      0.00
***** more *****
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
          End Net          Up Down

```

The columns contained in the report are described under [Accounting Data/Schedule of Jobs reports only in Fields and Columns: Reporting](#).

The PF keys are explained in [Special PF Keys: Reports](#).

Example of Network Description (short)

The following is an example of the report type **Network Description (short)**:

1. First report screen:

```
25/06/15                               Entire Operations                               14:29:55
                                     Network Documentation
Owner EXAMPLE      Network B60-FLOW      Version                               Job
-----
Description   : Job Flow, BS2000
Execution Node: N0082
Symbol Table  : EXAM-ST1      Symbol Table Version :

Schedule Times                               Send Late Message to
-----
Earliest Start: 07:30:00      ASN

Latest Start   : 07:30:00
Deadline       : 07:30:00
****

... Job Description on the following Pages

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                               End                               Menu
```

2. The following screen appears after ENTER:

```
25/06/15                               Entire Operations                               14:35:56
                                      Network Documentation
Owner EXAMPLE      Network B60-FLOW      Version                               Job JOB-02
-----
Job: JOB-02       Type JOB Macro      Description: Dep. JOB-15, JOB-19
Special Type   :
Location       : MAC
DSN/Library:    EOR-T541                               Member: B60-M02
Node          : N0082
Symbol Table   : EXAM-ST1      Symbol Table Version :

Schedule Parameters          Send Late Message To
-----
Earliest Start: 10:05:00
Latest Start   :
Deadline       :
Elapsed Time   :

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                               End                               Menu
```

3. The following screen appears after ENTER if input conditions and End-of-Job actions have been defined for the job:

```

25/06/15                               Entire Operations                               14:34:03
                                     Network Documentation
Owner EXAMPLE      Network B60-FLOW    Version                               Job JOB-02
-----
Input Conditions
-----
E60-J015-0 RUN    must exist
E60-J019-0 RUN    must exist
VERSION CHECK EXIT RUN    must exist

End of Job Processing
-----
If Job ended ok then do
... Set Condition E60-JOB2-01 RUN
... Set Condition E60-JOB2-02 RUN
... Reset Condition E60-J015-0 RUN
... Reset Condition E60-J019-0 RUN
... Reset Condition E60-JOB1-0 RUN
If Job ended not ok then do

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                                     End                                         Menu

```

The columns contained in the report are described in [Fields and Columns: Reporting](#).

The PF keys are explained in [Special PF Keys: Reports](#).

Example of Network Description (detailed and with JCL)

The following is an example of the report types **Network Description (detailed)** and **Network Description (with JCL)**.

1. First report screen:

```

25/06/15                Entire Operations                15:30:29
                        Network Documentation
Owner EXAMPLE          Network B60-FLOW          Version          Job
-----
Description   : Job Flow, BS2000
Execution Node: N0082
Symbol Table  : EXAM-ST1      Symbol Table Version :

Schedule Times                Send Late Message to
-----
Earliest Start: 07:30:00      ASN
Latest Start  : 07:30:00
Deadline      : 07:30:00
****

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                        End                                Menu

```

The network parameters are displayed as defined in the network administration.

2. The following screen(s) appear after ENTER:

```

25/06/15                Entire Operations                14:03:12
                        Network Documentation
Owner EXAMPLE          Network B60-FLOW          Version          Job
-----

Description
-----
Network E60-FLOW
-----

This Network is just an example of 'standard' job flow for
a bigger amount of jobs.
?
The jobs are all defined with the Dynamic JCL Facility to allow
an easy migration to another environment.
No special end-of-job handling is defined, so that the NATURAL
OPERATIONS global defaults will be used.
?
Flow Diagram
-----
?

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                        End                                Menu

```

Displays the description of the network (here: E60-FLOW) if available.

3. The following screen appears after ENTER:

```

25/06/15                               Entire Operations                               15:26:03
                               Network Documentation
Owner EXAMPLE      Network B60-FLOW      Version                               Job JOB-02
-----
Job: JOB-02      Type JOB Macro      Description: Dep. JOB-15, JOB-19
Special Type      :
Location          : MAC
DSN/Library:      EOR-T541                               Member: B60-M02
Node              : N0082
Symbol Table     : EXAM-ST1      Symbol Table Version :

Schedule Parameters      Send Late Message To
-----
Earliest Start: 10:05:00
Latest Start   :
Deadline       :
Elapsed Time   :

JCL for JOB-02 follows

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                               End                               Menu
    
```

The screen shows information on the first job (here: JOB-02) in the network.

The job parameters are displayed as defined in the [job maintenance](#).

JCL for JOB-02 follows only appears in a **Network Description (JCL)** report.

4. The following screen appears after ENTER for a **Network Description (JCL)** report only:

```
25/06/15                               Entire Operations                               15:27:37
                                         Network Documentation
Owner EXAMPLE      Network B60-FLOW      Version                               Job JOB-02
-----
@ DEFINE DATA PARAMETER USING NOPXPL-A
@ LOCAL              /* LOCAL VARIABLES START HERE
@ 1 L-DUMMY          (A01)              /* LEAVE IT NOT EMPTY
@ 1 L-JOBNAME        (A08)
@ END-DEFINE
@ * -----
@ MOVE P-JOB TO L-JOBNAME
@ EXAMINE L-JOBNAME FOR '-' DELETE
./@L-JOBNAME LOGON
/REMARK
/REMARK  ENTIRE OPERATIONS EXAMPLE JOB ON @*DATD
/REMARK
/STA L
/LOGOFF

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                               End                               Menu
```

5. The following screen appears after ENTER if a long description of the job exists and if input conditions have been defined for the job:

```

25/06/15                Entire Operations                16:14:25
                        Network Documentation
Owner EXAMPLE          Network B60-FLOW          Version                Job JOB-02
-----
Description
-----
Job JOB-02
-----
This job forces a defined event at End-of-Job time and executes the
program EORCONTI which is contained in the installation library
EORnnn.LOAD. The necessary JCL with the name EORE* is contained in
the installation library EORnnn.SRCE.

Input Conditions
-----
E60-J015-0 RUN      must exist
E60-J019-0 RUN      must exist
VERSION CHECK EXIT RUN      must exist

End of Job Processing
-----
If Job ended ok then do
... Set Condition E60-JOB2-01 RUN
... Set Condition E60-JOB2-02 RUN
... Reset Condition E60-J015-0 RUN
... Reset Condition E60-J019-0 RUN
... Reset Condition E60-JOB1-0 RUN
If Job ended not ok then do

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                        End                                Menu

```

The PF keys are explained in [Special PF Keys: Reports](#).

Example of Network Job Flow Display

The following is an example of the report type **Network Job Flow Display**:

```

29.06.12                Entire Operations                15:15:23
                        Job Flow of Network E60-FLOW    Owner EXAMPLE                Page 1
-----
      Job      by Condition      from/to Job      Owner      Network
-----
(1)  JOB-01
      I        E60-JOB1-0
      +-----> E60-JOB1-0 -----> (11)  JOB-019
      +-----> E60-JOB1-0 -----> (12)  J07
(2)  JOB-012
      I        E60-J012-0
(3)  JOB-013
      I        E60-J013-0
(4)  JOB-014
      I        E60-J014-0
(5)  JOB-015
      I        E60-J015-0
      +<----- E60-J019-0 <----- (11)  JOB-019
(6)  JOB-02
      I        E60-JOB2-01

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                        End                                Menu

```

Explanation:

- Each job is identified by a sequence number.
- Direct successor jobs appear in vertical order.
- If the direct sequence is interrupted, a horizontal line is printed.
- If a successor cannot be shown indirectly, an arrow --> points from the left.
- If a predecessor cannot be shown directly, an arrow <-- points from the right.
- The linking conditions are shown on the right-hand side of the screen.
- Input conditions dependent on files and dependent on job variables (BS2000) are displayed.

The PF keys are explained in [Special PF Keys: Reports](#).

Example of Schedule of Jobs

The following is an example of the report type **Schedule of Jobs**:

22.06.15		Entire Operations			15:51:59	
22.06.15		Production Plan 22.06.15 thru 22.06.15			Page 1	
Owner	Network	Job	Description	Start	Elapsed	
EXAMPLE	B60-FLOW	ABC3A	Description of Job ABC3A			
EXAMPLE	B60-FLOW	JOB-012	Depending on Job-01			
EXAMPLE	B60-FLOW	JOB-013	Depending on JOB-012			
EXAMPLE	B60-FLOW	JOB-014	Depending on JOB-013			
EXAMPLE	B60-FLOW	JOB-015	Depending on JOB-014			
EXAMPLE	B60-FLOW	JOB-019	Depending on JOB-01			
EXAMPLE	B60-FLOW	JOB-02	Dep. JOB-15, JOB-19	10:05		
EXAMPLE	B60-FLOW	JOB-03	Depending on JOB-02			
EXAMPLE	B60-FLOW	JOB-04	Depending on JOB-03			
EXAMPLE	B60-FLOW	JOB-05	Depending on JOB-04			
EXAMPLE	B60-FLOW	JOB-06	Where it all ends	10:30		
EXAMPLE	B60-FLOW	MILE1-TEST	Milestone Job			
***** Bottom of data *****						
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---						
			End	Down	Menu	

The columns contained in the report are described under [Accounting Data/Schedule of Jobs reports only](#) in [Fields and Columns: Reporting](#).

The PF keys are explained in [Special PF Keys: Reports](#).

Example of Network Start Summary

The following is an example of the report type **Network Start Summary**:

```

29.06.12                Entire Operations                15:47:53
                        Network Start Summary for 29.06.12 thru 29.06.12      Page 1
-----
SN          A-1
-----
(5972) 28.06 09:07      Awaiting Symbol Prompting
        28.06 09:05      Activation with Default Version
        28.06 09:07      Symbol Table SN/A-1/5972/SN-01 activated
        28.06 09:07      EOR4510 - Symbol Table SN/A-1/5972/SN-05 emp
        28.06 09:07      Symbol Table SN/A-1/5972/ABC1A activated
        28.06 09:07      Symbol Prompt Request sent to SN
        28.06 09:08      EOR2509 - Message Sending to SN via Node 146

SN          A-2
-----
        28.06 09:08      Scheduled, but not extracted

SN          A-207
-----
        28.06 09:08      Scheduled, but not extracted

```

The columns contained in the report are described in [Fields and Columns: Reporting](#).

Example of Network Schedule Overview

The following is an example of the report type **Network Schedule Overview**:

```

MORE
30.06.12                Entire Operations                11:41:46
Owner  EXAMPLE
Network E52-LOGRES      Schedule from 20.04.10 thru 20.04.10      Page 1
-----
          Date      Time  Owner      Network      Run  Type
-----
Tuesday  20.04.10  08:00  EXAMPLE    E52-LOGRES    Schedule, periodic
-----
Tuesday  20.04.10  00:00  EXAMPLE    E60-FLOW      Schedule, periodic
-----

***** End of Report *****

```

The columns contained in the report are described in [Fields and Columns: Reporting](#).

Example of Activation Overview

The following is an example of the report type **Activation Overview**:

```

Entire Operations: Activation Ov
-----
Date: 10.06.15
Owner      Network      Version      Job      Run      Time      Type      Message
-----
EXAMPLE    B60-FLOW          -           1930 00:00 Scheduled Activation
EXAMPLE    E01-CONTI        -           428 00:00 Scheduled Activation
EXAMPLE    E52-LOGRES       -           5051 00:00 Scheduled Activation
EXAMPLE    E60-FLOW         -           4492 00:00 Scheduled Activation
EXAMPLE    Z60-FLOW         -           397 00:00 Scheduled Activation

***** Bottom of Data *****

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                End                                Left  Right Menu
    
```

Press ENTER to display the next screen if there is more data.

Press PF11 to scroll right to see additional information.

The columns contained in the report are described in [Fields and Columns: Reporting](#).

The PF keys are explained in [Special PF Keys: Reports](#).

Example of Compare Symbol Tables

The following is an example of the report type **Compare Symbol Tables**:

```

MORE
07.04.16                Entire Operations                15:00:43
                        Compare Symbol Tables            Page    4

Owner EXAMPLE           Table ADMIN           Version

Symbol                 F A Value
-----
Different attributes:
  Value
  Prompting
  Prompt Text
Missing      CLASS,N1319A ,NOPALL
Differs from CLASS,N1352B ,NOPALL
Different attributes:
  Value
  Prompting
  Prompt Text
Missing      CLASS,N1360A ,NOPALL
Missing      CLASS,N1378A ,NOPALL
Missing      CLASS,N1399A01 ,NOPALL ←

```

Columns: Compare Symbol Tables

The columns contained in the report are described under [Compare Networks/Symbol Tables reports only in Fields and Columns: Reporting](#).

Example of Compare Networks

The following are examples of the report type **Compare Networks**:

Example 1:

```

MORE
07.04.16                Entire Operations                15:39:15
                        Compare Networks            Page    1

Owner SMR               Network A-1B           Version vhugo

Network Attributes
-----
Differs from A-1B <V02>,SMR
Different attributes:
  Description
  Symbol Table Version ←

```

This screen shows that the description and the symbol table version differ between Network A-1B with version vhugo of Owner SMR and Network A-1B with version V02 of owner SMR.

Example 2:

```

MORE
07.04.16          Entire Operations          15:44:19
                  Compare Networks          Page    2

Owner HEB          Network AAA          Version v626

Job      Type Loc Description
-----
TRIGGERED1 DUM
Differs from TRIGGERED1,AAA <v6261>,HEB
Different attributes:
  Symbol Table
  Symbol Table Version
Additional   NV,AAA <v6261>,HEB
Additional   RCV,AAA <v6261>,HEB
Additional   SMRTEST,AAA <v6261>,HEB
Additional   1-A,AAA <v6261>,HEB
Additional   1-B,AAA <v6261>,HEB

```

The job TRIGGERED1 of Network AAA with version v626 of owner HEB has a different symbol table and a different symbol table version than the job with the same name in Network AAA with version v6261 of owner HEB. The jobs NV, RCV, SMRTEST, 1-A and 1-B exist additionally in Network AAA with version v6261 of owner HEB.

Columns: Compare Networks

The columns contained in the report are described under [Compare Networks/Symbol Tables reports only](#) in [Fields and Columns: Reporting](#).

Example of Node Overview

The following is an example of the report type **Node Overview**:

Entire Operations: Node Overview										
Server Name	Node	Short	AM	Op.Sys.	Wait a. Error	VSE SSU	Time SysId	Diff.	Valid	NPR Ver
Node 0001	1	N0001	N		4	U	3	-11.50	yes	
Node 0002	2	N0002	N	BS2000	5				yes	3.5
n4	4	N4	B		5			10.00	yes	
n5	5	N5	B		5				yes	
Adabas DB 9	9	N0009	N		5				yes	
TEST	12	HUGO	N		5				yes	
hannes	21	21	N		5				yes	
BS2000 SIH2	31	N0031	N	BS2000	1	U			yes	3.5
Broker 34	34	N0034	N	rted	5				yes	1.2
BS2 131	38	N0038	N	BS2000	5				yes	3.5
NPR 321	40	N0040	N	MVS/ESA	5				yes	3.5
Loc1 Nd DQA V134	42	N0042	N	MVS/ESA	5				yes	3.5
QE F-MC	53	N0053	N	MVS/ESA	5				yes	3.3
WOS-54	54	N0054	N	BS2000	5				yes	
ESM MVS	62	N0062	N	MVS/ESA	5				yes	2.2
Node 65	65	N0065	N	MVS/ESA	5				yes	3.3
***** m o r e *****										

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
 End Down Left Right Menu

Node Overview

Generation Date: 2016-03-23 - 17:03:12

Node Number	Node Short Name	Node Name	Access Mode	Operating System	Wait After Error	Submit Security User Type	VSE SystemID	Time Difference	Valid	NPR Version	OS Release
31	N0031	BS2000	N	BS2000	5		0		yes	3.5.4	
33	N0033	VSE	N	DOS/ESA	5		0		yes	3.5.4	
42	42	QANODE42	N	MVS/ESA	5		0		yes	3.6.1	
77	N0077	Test Node 77	B		5		0		yes	3.6.1	
148	N0148	XCEOR210	N	MVS/ESA	5		0		yes	3.6.1	
194	BS200	OSD-BS2	N	BS2000	5		0		yes	3.5.4	

The columns contained in the report are described under [Node Overview reports only](#) in [Fields and Columns: Reporting](#).

The PF keys are explained in [Special PF Keys: Reports](#).

Example of Network/Job Usage

The following is an example of the report type **Network/Job Usage**:

```

MORE
15.11.16                               Entire Operations                               15:25:13
                                         Network/Job Usage                               Page    6

```

Owner	Used Job Network	Job	Owner	Using Job Network	Version	Job
SAG	AAA	J-OHNE-VER	SAG	BBB	Current	TRIGGERED1
SAG	AAA	TRIGGERED1	SAG	EXTERN		JOB1
SAG	AAA	*	SAG	AAA	v6261	TRIGGERED1
Version: (void)			SAG	AAA	v6261a	TRIGGERED1
SAG	AAA	*	SAG	BBB	Current	TRIGGERED1
Version: (void)			SAG	SAG-MITSUB		SAG-NET1
SAG	AAA	*	SAG	SAG-MITSUB		SAG-NET1
Version: (void)			Usage: Subnet			

The columns contained in the report are described under [Network/Job Usage reports only](#) in [Fields and Columns: Reporting](#).

Generating Batch Reports

All reports can be generated in batch as well as online. In batch, there are no security or asterisk/wildcard restrictions.

Batch reports are printed on Natural printer 1, which must be allocated correctly in the job control. The Natural batch session must have the necessary LFILEs (131 and 216) correctly set and should run in delimiter input mode (IM=D). The examples below assume that input delimiter is set to “,” (comma).

The batch reporting programs input their parameters in several steps. This is described separately for each program.

This section covers the following topics:

- [Activation Overview](#)
- [Accounting Data](#)
- [Compare Networks in Batch](#)
- [Compare Symbol Tables in Batch](#)
- [Network/Job Usage in Batch](#)

- [Extended Log Report](#)
- [Log Reports](#)
- [Network Information](#)
- [Network Job Flow](#)
- [Network Start Summary](#)
- [Network Schedule Overview](#)
- [Node Overview Report](#)
- [Report of Used JCL](#)
- [Schedule of Jobs](#)

Activation Overview

The activation overview (equivalent to online option 14, see [Network and Job Accounting Data](#)) is produced by program RE-ACO-P. It does 3 INPUTs:

Syntax:

```
RE-ACO-P activation-types
owner-name, network-name
from-date, from-time, from-run, to-date, to-time, to-run
```

1. Activation Types to be included in report:

- - * - all network activations.
 - A - networks activated via NOPUAC5N [Entire Operations API-Routine](#) .
 - M - networks activated manually.
 - O - networks activated as a result of EOJ checking.
 - R - networks activated as part of recovery processing for an abended job.
 - S - networks activated according to schedule. You can either specify * or any combination of **A, M, O, R, S**

2. Owner/Network

- Owner name. *, > and < wildcards are allowed. Use * for all owners.
- Network name. *, > and < wildcards are allowed. Use * for all networks.

3. Date, time, run number ranges

- From date (format YYYYMMDD)
- From time (format HH:II:SS)
- From run (maximum 999999999)
- To date (format YYYYMMDD)
- To time (format HH:II:SS)

- To run (maximum 999999999)

See also [Date and Time Formats](#).

Examples

Below are examples of Natural CMSYNIN data for producing various activation overview reports:

Example 1:

```
RE-ACO-P AM
EXAMPLE,*
20081101,,,20081130 (equivalent to ←
20081101,00:00:00,1,20081130,23:59:59,999999999)
```

will produce an overview of all networks under owner EXAMPLE, activated manually or by API between 1st and 30th November 2008.

Example 2:

```
RE-ACO-P *
*,*
20081115,14:00:00,71,20081115,22:00:00,99
```

will produce an overview of all activations between 14:00:00 and 22:00:00 on 15th November 2008, with run numbers between 71 and 99.

Accounting Data

Accounting data (equivalent to online option 6, see [Network and Job Accounting Data](#)) is produced by program RE-ACC-P. It does 2 INPUTs:

Syntax:

```
RE-ACC-P
owner-name,network-name,job-name
from-date,from-time,from-run,to-date,to-time,to-run
```

1. Owner/Network/Job

- Owner name. *, > and < wildcards are allowed. Use * for all owners.
- Network name. *, > and < wildcards are allowed. Use * for all networks.
- Job name. *, > and < wildcards are allowed. Use * for all jobs.

2. Date, time, run number ranges

- From date (format YYYYMMDD)

- From time (format HH:II:SS)
- From run (maximum 999999999)
- To date (format YYYYMMDD)
- To time (format HH:II:SS)
- To run (maximum 999999999)

See also [Date and Time Formats](#).

Examples

Below are examples of Natural CMSYNIN data for producing various accounting reports:

Example 1:

```
RE-ACC-P
EXAMPLE,E01-CONTI
20081101,,20081130,23:59:59,999999999
```

will produce accounting data for all jobs in network E01-CONTI under owner EXAMPLE, starting at 00:00:00 on 1st November 2008 and ending at 23:59:59 on 30th November 2008, for run numbers in the range 0 to 999999999.

Example 2:

```
RE-ACC-P
EXAMPLE*,E10*
20081101,09:00:00,51,20081231,23:59:59,60
```

will produce accounting data for all jobs beginning E10 in all networks under owner EXAMPLE, starting at 09:00:00 on 1st November 2008 ending at 23:59:59 on 31st December 2008, with run numbers between 51 and 60.

Compare Networks in Batch

This report displays the results of the comparison of one or more networks.

Syntax:

```
RE-CNV-P
owner1,network1,version1,
owner2,network2,version2,
show-content
```

1. *owner1/network1/version1*:

- Owner name. *, > and < wildcards are allowed. Use * for all owners.

- Network. *, > and < wildcards are allowed. Use * for all networks.
- Version. *, > and < wildcards are allowed. Use * for all versions.

2. *owner2/network2/version2*:

- Owner name. *, > and < wildcards are allowed. Use * for all owners.
- Network. *, > and < wildcards are allowed. Use * for all networks.
- Version. *, > and < wildcards are allowed. Use * for all versions.

3. *show-content*:

- A = All objects.
- D = Different objects.
- M= Give message about result.

Compare Symbol Tables in Batch

This report displays the results of the comparison of one or more symbol tables.

Syntax:

```
RE-CNV-P
owner1,symbol-table1,symbol-table-version1,
owner2,symbol-table2,symbol-table-version2,
show-content
```

1. *owner1/symbol-table1/symbol-table-version1*:

- Owner name. *, > and < wildcards are allowed. Use * for all owners.
- Symbol Table. *, > and < wildcards are allowed. Use * for all symbol tables.
- Symbol Table Version name. *, > and < wildcards are allowed. Use * for all symbol table versions.

2. *owner2/symbol-table2/symbol-table-version2*:

- Owner name. *, > and < wildcards are allowed. Use * for all owners.
- Symbol Table. *, > and < wildcards are allowed. Use * for all symbol tables.
- Symbol Table Version name. *, > and < wildcards are allowed. Use * for all symbol table versions.

3. *show-content*:

- A = All objects.
- D = Different objects.
- M= Give message about result.

Network/Job Usage in Batch

The report lists networks and related subnetworks (jobs of the type NET) as well as jobs activated by an end-of-job action or a recovery job. See also *Example of Network/Job Usage*.

Syntax:

```
RE-XRF-P
owner1,network1,version1
owner2,network2,version2
job-1,job-2
```

1. *owner1/network1/version1* (used job):
 - Owner name. *, > and < wildcards are allowed. Use * for all owners.
 - Network. *, > and < wildcards are allowed. Use * for all networks.
 - Version. *, > and < wildcards are allowed. Use * for all versions.
2. *owner2/network2/version2* (using job):
 - Owner name. *, > and < wildcards are allowed. Use * for all owners.
 - Network. *, > and < wildcards are allowed. Use * for all networks.
 - Version. *, > and < wildcards are allowed. Use * for all versions.
3. *job-1* (used job) and *job-2* (using job):
 - Job name. *, > and < wildcards are allowed. Use * for all jobs.

Extended Log Report

This report is only available in batch and can be used to print the following:

- log records
- job output written to the log
- active JCL changes written to the log
- active JCL

The report program is NOPLP01P and it does one input to receive these parameters:

Because of the large number of parameters, you are recommended to use keyword mode for specifying them. NOPLP01P can be executed in a standard batch Natural, with LFILEs 131 and 216 set.

The output is written to the Natural system print data set (CMPRINT).

Syntax:

```
NOPLP01P
parameter=value%
parameter=value%
parameter=value%
```

Parameter	Explanation
FROM-DATE-TIME	Format YYYYMMDDHHIISS, starting time for log report.
THRU-DATE-TIME	Format YYYYMMDDHHIISS, ending time for log report.
LANG	Language in which log records will be printed, either 1 for English (the default) or 2 for German.
OWNER	Owner(s) to be included in the report. The wildcards *, > and < are allowed.
NETWORK	Network(s) to be included in the report. The wildcards *, > and < are allowed.
DBENV	Reserved for future use. Do not specify this parameter.
RUN-FROM	Starting run number.
RUN-THRU	Ending run number.
JOB	Job(s) to be included in the report. The wildcards *, > and < are allowed.
ACTJCL	Y or N. If Y, any active JCL for jobs matching the owner/network/job/run criteria is printed after the log information.
SYSOUT	Y or N. If Y, any logged SYSOUT for jobs matching the owner/network/run/job/date/time criteria is printed.
CHGJCL	Y or N. If Y, any active JCL modifications for jobs matching the owner/network/run/job/date/time criteria are printed.
EXTLOG	Y or N. Specify a default value for ACTJCL, SYSOUT and CHGJCL.

The following example assumes the Natural parameters IM=D, IA==, CF=%.

Example:

```
NOPLP01P
FROM-DATE-TIME=20081101000000%
THRU-DATE-TIME=20081115163000%
RUN-FROM=1%
RUN-THRU=999%
OWNER=UKSJ*%
NETWORK=TEST-2>%
```

```
JOB=L<%
LANG=2%
EXTLOG=Y
```

This example will print all log records as well as logged SYSOUT, logged active JCL modifications and active JCL for jobs with names less than L in networks with names greater than TEST-2 under owners beginning with UKSJ, with run numbers in the range 1 to 999 between 00:00 on 1st November 2008 and 16:30 on 15th November 2008. The log records will be printed in German.



Note: The date/time range has no influence on printing of active JCL.

Log Reports

Log reports (equivalent to online options 1, 2, 3, 4 and 5) are produced by program RE-LOG-P.

Syntax:

```
RE-LOG-P report-type,count-jobs
owner-name,network-name,job-name
from-date,from-time,from-run,to-date,to-time,to-run
```

1. Options:



A All events in log.

B Abended jobs.

T Terminated jobs.

X Jobs not started due to start time, input conditions, submission permanent errors and extraction + activation * permanent errors.

Y Jobs not started due to submission permanent errors and extraction + activation permanent errors.

N Networks not activated due to extraction or activation errors.

■ Count Jobs

Y Produce a count of jobs.



Caution: This requires a rereading of the whole log and therefore will greatly increase the elapsed and CPU times and the number of Adabas calls for the batch job

N Do not produce a count of jobs.

2. Owner / Network / Network Version / Job

- Owner name. *, > and < wildcards are allowed. Use * for all owners.
- Network name. *, > and < wildcards are allowed. Use * for all networks.
- Job name. *, > and < wildcards are allowed. Use * for all jobs.

3. Date, time, run number ranges

- From date (format YYYYMMDD)
- From time (format HH:II:SS)
- From run (maximum 999999999)
- To date (format YYYYMMDD)
- To time (format HH:II:SS)
- To run (maximum 999999999)

See also [Date and Time Formats](#).

Examples

Below are examples of Natural CMSYNIN data for producing various log reports:

Example 1:

```
RE-LOG-P T,Y  
EX*,E10>  
20081101,, ,20081130,23:59:59,999999999
```

will produce a report of all terminated jobs in networks with names greater than E10 under owners beginning EX, starting at 00:00:00 on 1st November 2008 and ending at 23:59:59 on 30th November 2008, for run numbers in the range 0 to 999999999. A count of jobs will be produced.

Example 1:

```
RE-LOG-P A,N  
*,*,  
20081114,09:00:00,51,20081114,17:30:00,60
```

will produce a report of all events between 09:00:00 and 17:30:00 on 14th November 2008 for all owners, networks and jobs with run numbers between 51 and 60. A count of jobs will not be produced.

Network Information

Network reports (equivalent to online options 7, 8 and 9, see [Network and Job Accounting Data](#)) are produced by program RE-NET-P. It does 2 INPUTs:

Syntax:

```
RE-NET-P report-type
         owner-name, network-name
```

1. Report type

- S Report Type (short).
- D Report Type (detailed).
- J Report Type (detailed, with JCL).

2. Owner/Network

- Owner name. *, > and < wildcards are allowed. Use * for all owners.
- Network name. *, > and < wildcards are allowed. Use * for all networks.

Examples

Below are examples of Natural CMSYNIN data for producing various network reports:

Example 1:

```
RE-NET-P S
*,*
```

will produce a summary report for all networks.

Example 2:

```
RE-NET-P J
SYS*,W>
```

will produce a detailed report, including JCL for all networks with names greater than W in owners beginning SYS.

Network Job Flow

Network job flow (equivalent to online option 14, see [Network and Job Accounting Data](#)) is produced by program RE-FLW-P. It does 1 INPUT:

Syntax:

```
RE-FLW-P  
owner-name, network-name
```

1. Owner/Network

- Owner name. *, > and < wildcards are allowed. Use * for all owners.
- Network name. *, > and < wildcards are allowed. Use * for all networks.

Examples

Below are examples of Natural CMSYNIN data for producing various network job flow reports:

Example 1:

```
RE-FLW-P  
EXAMPLE, E60-FLOW
```

will produce a report for E60-FLOW in owner EXAMPLE.

Example 2:

```
RE-FLW-P  
SYS*, W>
```

will produce a report for all networks with names greater than W in owners beginning SYS.

Network Start Summary

The network start summary (equivalent to online option 12, see [Network and Job Accounting Data](#)) is produced by program RE-ACT-P. It does 2 INPUTS:

1. Owner/Network

- Owner name. *, > and < wildcards are allowed. Use * for all owners.
- Network name. *, > and < wildcards are allowed. Use * for all networks.

2. Date range

- From date (format YYYYMMDD)
- To date (format YYYYMMDD)

See also [Date and Time Formats](#).

Examples

Below are examples of Natural CMSYNIN data for producing various network start summary reports:

Example 1:

```
RE-ACT-P
EXAMPLE,*
20081101,20081130
```

will produce a summary of all networks under owner EXAMPLE, between 1st and 30th November 2008.

Example 2:

```
RE-ACT-P
E*,M<
20081101,20081231
```

will produce a summary of all networks with names less than M under owners beginning with E between 1st November and 31st December 2008.

Network Schedule Overview

The network schedule overview (equivalent to online option 13, see [Network and Job Accounting Data](#)) is produced by program RE-NSC-P. It does 3 INPUTs:

Syntax:

```
RE-NSC-P report-type
         owner-name,network-name
         from-date,to-date
```

1. Report Type

- A - with schedule.
- B - to activate, sorted by network and time.
- C - to activate, sorted by time.

2. Owner/Network

- Owner name. *, > and < wildcards are allowed. Use * for all owners.
- Network name. *, > and < wildcards are allowed. Use * for all networks.

3. Date range

- From date (format YYYYMMDD).

- To date (format YYYYMMDD).

See also [Date and Time Formats](#).

Examples

Below are examples of Natural CMSYNIN data for producing various schedule overview reports:

Example 1:

```
RE-NSC-P A
EXAMPLE,*
20081101,20081130
```

will produce an overview of all networks under owner EXAMPLE, scheduled to run between 1st and 30th November 2008.

Example 2:

```
RE-NSC-P B
E*,M<
20081101,20081231
```

will produce an overview of schedule extractions (activations) of all networks with names less than M under owners beginning with E between 1st November and 31st December 2008.

Node Overview Report

Node Overview Reports are generated with the program RE-NOD-P.

Syntax:

```
RE-NOD-P from-node to-node
```

Report of Used JCL

The report of used JCL is produced by the program RJCLOC-P. It does 2 INPUTs:

Syntax:

```
RJCLOC-P owner-name, network-name
```

- **Owner** Wildcard may be used.
- **Network** Wildcard may be used.

Below are examples of Natural CMSYNIN data for producing JCL usage reports:

Example 1:

```
RJCLOC-P EXAMPLE NET-01
```

List used JCL of network EXAMPLE / NET-01.

Example 2:

```
RJCLOC-P TEST* *
```

List used JCL of all networks of owners with have TEST as begin of their names.

Schedule of Jobs

Job schedules (equivalent to online option 11, see [Network and Job Accounting Data](#)) are produced by program RE-JSC-P. It does 2 INPUTs:

Syntax:

```
RE-JSC-P
owner-name, network-name
from-date, to-date
```

1. Owner/Network

- Owner name. *, > and < wildcards are allowed. Use * for all owners.
- Network name. *, > and < wildcards are allowed. Use * for all networks.

2. Date range

- From date (format YYYYMMDD)
- To date (format YYYYMMDD)

See also [Date and Time Formats](#).

Examples

Below are examples of Natural CMSYNIN data for producing various job schedule reports:

Example 1:

```
RE-JSC-P  
EXAMPLE,E01-CONTI  
20081101,20081130
```

will produce a schedule of jobs in network E01-CONTI under owner EXAMPLE, between 1st and 30th November 2008.

Example 2:

```
RE-JSC-P  
*,*  
20081101,20081231
```

will produce a schedule of all jobs, between 1st and 30th November 2008.

XV

Cross References

71 Cross References

- Types of Cross Reference Reports 752
- Generating Cross Reference Reports Online 753
- Fields and Columns: Cross References 762
- Examples of Cross Reference Reports 766
- Generating Cross Reference Reports in Batch 771

The **Cross References** function is used to cross check Entire Operations objects and produce a report from the data found.

This section describes how to generate cross reference reports in online and batch mode.

Related Topic:

- For general information on using reports, see [Generating Online Reports](#) in the section [Reporting](#).

Types of Cross Reference Reports

The types of cross references you can select from the **Cross References menu** are described in the following table.

Report Type	Description
Exit Usage	Lists user exits defined in your environment. See also Example of User Exit Usage .
Symbol Table Usage	Lists symbol tables used in networks and jobs. Note: The cross reference for symbol tables includes the symbol table usage for input conditions and for end-of-job symbol settings. See also Example of Symbol Table Usage .
JCL Usage	Lists JCL files used in networks and jobs. See also Example of JCL Usage .
Node Usage	Lists JCL nodes and execution nodes used in networks. See also Example of Node Usage .
Symbol Search by Value	Lists symbols that match a specified value. See also Example of Symbol Search by Value .
List Undefined Objects	Lists objects that are referenced by other objects which are not defined in your environment. The missing definition of these objects might lead to errors during network activation. Note: JCL is not inspected for undefined symbols. See also Example of Undefined Objects .
Resource Usage	Lists resources used by active jobs. See also Example of Resource Usage .

Generating Cross Reference Reports Online

This section describes how to generate cross reference reports online.

Report data is evaluated for the current day.

» **To generate a cross reference report**

- 1 From the **Main Menu**, choose **Cross References** and press ENTER.

Or:

In the **Command** field of the **Main Menu**, type Option Code 9 or type XREF (see *Direct Commands*) and press ENTER.

Or:

You can skip this step if you know the number of the report type (for example, 7 for **Resource Usage**) you want to generate:

In the **Command** field of the **Main Menu**, type Option Code 9.7 and press ENTER.

A **Cross References** menu similar to the example below appears:

```
16.04.15          ***** Entire Operations *****          09:59:32
Owner EXAMPLE          Cross References          User ID SAG
-----
          Cross References

1  Exit Usage
2  Symbol Table Usage
3  JCL Usage
4  Node Usage
5  Symbol Search by Value
6  List Undefined Objects
7  Resource Usage

Command => _____

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

The menu displays a list of selectable **report types**.

- 2 Select the type of cross reference for which you want to generate a report and press ENTER.

Depending on the report type selected, proceed as described in the relevant step for:

Exit Usage

Symbol Table Usage

JCL Usage

Node Usage

Symbol Search by Value

List Undefined Objects

Resource Usage

- 3 **Exit Usage:** If you select **Exit Usage** from the **Cross References** menu and press ENTER, a window similar to the example below opens:

```

+-----+
!           !
!   Send Output to Printer ?  N (Y/N)   !
!           !
+-----+

```

Enter Y (Yes) or N (No) and press ENTER. N is the default setting.

A screen similar to the example below appears:

```

08.04.16          ***** Entire Operations *****          17:58:11
                   Usage of Exits
-----
Please mark one or several Types.

Cmd  Type
--   -
_    Input Condition Exits
_    Resource Master Determination Exits
_    Master JCL and Natural Programs
_    End-of-Job Check Exits
_    End-of-Job Action Exits
_    Symbol Check Exits
_    Symbol Prompt Exits

Command => _____

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                               End

```

Mark one of the fields to select the cross reference you want to display and press ENTER.

A report output screen with a list of used user exits appears as shown in [Example of User Exit Usage](#).

If MORE appears at the top of the report screen, press ENTER to display additional pages of the report.

Press PF3 to exit the report screen.

- 4 **Symbol Table Usage:** If you select **Symbol Table Usage** from the [Cross References](#) menu and press ENTER, a screen similar to the example below appears:

An additional window similar to the example below opens:

```

+-----+
| JCL Usage                                     |
| Owner.....: EXAMPLE___ (Blank for all, * for selection |
| Network....: B60-FLOW___ or * and PF4 to process all  |
| Version....: _____ matching objects)           |
|                                               |
| Destination: 1                               |
|   1 - Screen  2 - Printer  3 - PC  4 - PC-CSV      |
| Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9--- |
|           Help       End   Apply Acpt           |
+-----+

```

Use the input fields to specify selection criteria for the report to be produced as described in [Fields and Columns: Cross References](#).

When you are finished, press PF4 or PF5 (see also [Confirming Selections](#) in the section *Reporting*).

A report output screen with a list of used JCL appears as shown in [Example of JCL Usage](#).

If MORE appears at the top of the report screen, press ENTER to display additional pages of the report.

Press PF3 to exit the report screen.

- 6 **Node Usage:** If you select **Node Usage** from the [Cross References](#) menu and press ENTER, a window similar to the example below opens:

```

+-----+
| !                                     ! |
| !                               Node Usage ! |
| !                                     ! |
| ! JCL Node      ==> _____ (Blank for all ! |
| ! Execution Node ==> _____ or * for selection) ! |
| !                                     ! |
| ! Usage in master or active jobs ==> M ('M' - Master; 'A' - Active) ! |
| !                                     ! |
| ! Enter-----PF3----- ! |
| !           End           ! |
| !                                     ! |
| !                                     ! |
+-----+

```

Enter the selection criteria required for the report to be produced. For valid input values, see [Fields and Columns: Cross References](#).

When you are finished, press ENTER.

An additional window similar to the example below opens:

```

+-----+
| Node Usage                                     |
| Owner.....: EXAMPLE__ (Blank for all, * for selection |
| Network....: B60-FLOW__ or * and PF4 to process all |
| Version....: _____ matching objects)          |
|                                                |
| Destination: 1                                |
|   1 - Screen  2 - Printer  3 - PC  4 - PC-CSV      |
| Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9--- |
|   Help       End   Apply Acpt                    |
+-----+
    
```

Enter the required selection criteria. For valid input values, see [Fields and Columns: Cross References](#).

When you are finished, press PF4 or PF5 (see also [Confirming Selections](#) in the section *Reporting*).

A report output screen with a list of used nodes appears as shown in [Example of Node Usage](#).

If MORE appears at the top of the report screen, press ENTER to display additional pages of the report.

Press PF3 to exit the report screen.

- 7 **Symbol Search by Value:** If you select **Symbol Search by Value** from the [Cross References](#) menu and press ENTER, a screen similar to the example below appears:

```

                                Symbol Search by Value

Owner          ==> EXAMPLE____      (Blank for all, * for selection
Symbol Table ==> _____          or * and PF4 to process all
Version        ==> _____          matching objects)

Symbol         ==> _____

Mult.value index: From ==> ____   To ==> *__
Usage in: Master ==> _   Active ==> _

Case sensitive ==> _           At position ==> *_ (*,nn,E)

Scan for ==> _____
                1...+...10....+...20....+...30....+...40....+...50....+...60

Destination   ==> 1
    1 - Screen  2 - Printer  3 - PC  4 - PC-CSV
Enter-----PF3-----PF4-----
                End      Apply

```

Enter the selection criteria required for the report to be produced. For valid input values, see [Fields and Columns: Cross References](#).

When you are finished, press PF4 (see also [Confirming Selections](#) in the section *Reporting*).

A report output screen with a list of used symbols appears as shown in [Example of Symbol Search by Value](#).

If MORE appears at the top of the report screen, press ENTER to display additional pages of the report.

Press PF3 to exit the report screen.

- 8 **List Undefined Objects:** If you select **List Undefined Objects** from the [Cross References](#) menu and press ENTER, a window similar to the example below opens:

```

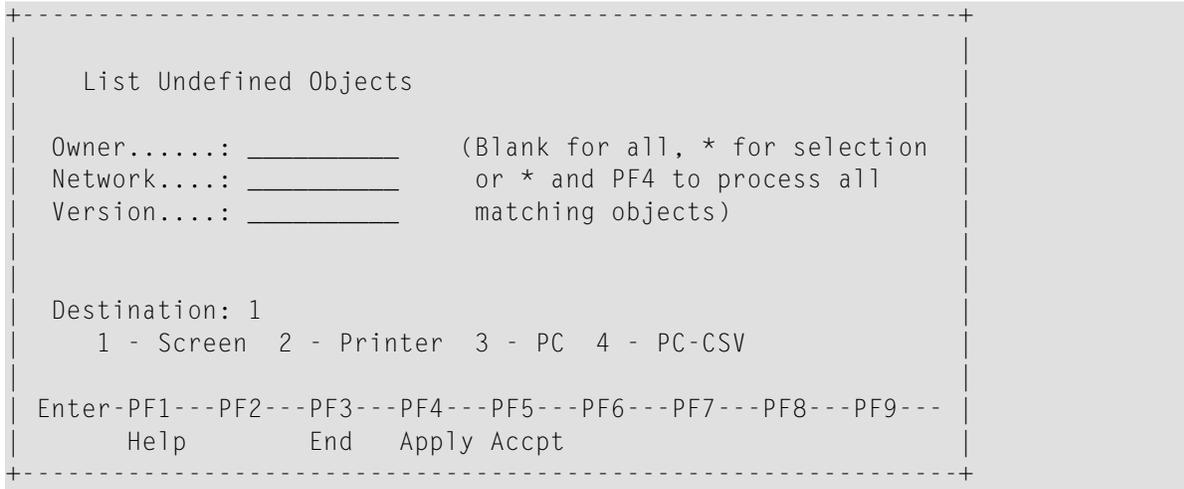
+-----+
|                                     |
|                               List Undefined Objects                       |
|                                     |
| Mark the referenced object types to be inspected                         |
|                                     |
| Symbol          : X                Network          : X                   |
| Symbol Table    : X                Job              : X                   |
| Resource        : X                Schedule         : X                   |
| Calendar        : X                |
|                                     |
| Enter-----PF3-----|
|                               End                                         |
+-----+

```

All object types for which you can search are selected by default. Deselect the object type(s) you do not require by entering a blank character in the corresponding input field.

When you are finished, press ENTER.

An additional window similar to the example below opens:



Enter the selection criteria required for the report to be produced. For valid input values, see [Fields and Columns: Cross References](#).

When you are finished, press PF4 or PF5 (see also [Confirming Selections](#) in the section *Reporting*).

A report output screen with a list of used undefined objects appears as shown in [Example of Undefined Objects](#).

Press ENTER to display additional pages of the report (if available), or PF3 to exit the report screen.

- 9 **Resource Usage:** If you select **Resource Usage** from the [Cross References](#) menu and press ENTER, a window similar to the example below opens:

```

Resource Usage

Resource....: _____
(Blank for all, * for selection
or * and PF4 to process all
matching objects)

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8-
                End   Apply Accpt

```

Specify the resource(s) to be checked: for valid input values, see the **Resource** field in *Fields: Cross References*.

When you are finished, press PF4 or PF5 (see also *Confirming Selections* in the section *Reporting*).

An additional window similar to the example below opens:

```

Resource Usage

Owner.....: _____ (Blank for all, * for selection
Network....: _____ or * and PF4 to process all
Version....: _____ matching objects)

Destination: 1
             1 - Screen 2 - Printer 3 - PC 4 - PC-CSV

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---
             Help      End   Apply Accpt

```

Enter the selection criteria required for the report to be produced. For valid input values, see *Fields and Columns: Cross References*.

When you are finished, press PF4 or PF5 (see also *Confirming Selections* in the section *Reporting*).

A report output screen with a list of used resources appears as shown in *Example of Resource Usage*.

Press ENTER to display additional pages of the report (if available). Press PF11 or PF10 to scroll right or left in the screen.

Press PF3 to exit the report screen.

Fields and Columns: Cross References

The input fields for specifying selection criteria and output options for the cross reference report to be produced are explained in the following table. The input fields correspond to the columns contained on the report output screen.

Depending on the report type (to be) produced, the following fields and columns are available:

Field/Column	Explanation
Owner	Name of an owner or a range of names: use an asterisk (*) or a smaller/greater than (< or >) wildcard character to determine a start or an end value (see also <i>Valid Name Specifications</i> in <i>Direct Commands</i>) or open a selection list of names. Leave the field blank to select all names.
Network	Name of a network or a range of names: use an asterisk (*) or a smaller/greater than (< or >) wildcard character to determine a start or an end value (see also <i>Valid Name Specifications</i> in <i>Direct Commands</i>) or open a selection list of names. Leave the field blank to select all names.
Network Version	Name of a network version or a range of names: use an asterisk (*) or a smaller/greater than (< or >) wildcard character to determine a start or an end value (see also <i>Valid Name Specifications</i> in <i>Direct Commands</i>) or open a selection list of names. Leave the field blank to select all names.
Symbol Table	Name of a symbol table or a range of names: use an asterisk (*) or a smaller/greater than (< or >) wildcard character to determine a start or an end value (see also <i>Valid Name Specifications</i> in <i>Direct Commands</i>) or open a selection list of names. Leave the field blank to select all names.
Symbol Table Version	Version of the specified symbol table.
Destination	See Report Output Options in the section <i>Reporting</i> .
Resource	Name of a resource. This can reflect real resources or can describe a fictitious resource. Enter a resource name or a range of names: use an asterisk (*) or a wildcard character (< or >) to determine a start or an end value (see also <i>Valid Name Specifications</i> in <i>Direct Commands</i>) or open a selection list of names.

Field/Column	Explanation	
	Leave the field blank to select all names.	
User Exit Usage reports only:		
Job	Name of the job using the user exit.	
ExitLib	Name of the library that contains the user exit.	
Exit	Name of the user exit.	
Enabled	Depending on the type of user exit selected, indicates whether the user exit is enabled (Y) or disabled (N).	
BG	This column is only shown for symbol prompt exits. It indicates whether symbol prompting runs in the background:	
	Y	Symbol prompting runs in the background.
	N	A window prompts the user for input.
	B	Performs both opens a prompt window and then runs prompting in the background.
	See also Specifying User Exits for Symbol Modification .	
Type	Type of user exit(s) to be selected: Input condition exits Resource master determination exits Master JCL and Natural programs End-of-Job (EOJ) check exits End-of-Job (EOJ) action exits Symbol check exits Symbol prompt exits	
Symbol Table Usage reports only:		
Usage in active jobs	Jobs to be searched.	
	Possible values:	
	N	Selects master jobs only (default).
	Y	Selects both master jobs and active jobs.
JCL Usage reports only:		
JCL Location	A supported JCL location. All possible locations are described in List of JCL Locations in the section <i>Job Maintenance</i> . Leave the field blank to select all locations.	

Field/Column	Explanation	
File/Natlib	Name of a file or a Natural library, or a range of names: use an asterisk (*) to determine a start value (see also <i>Valid Name Specifications</i> in <i>Direct Commands</i>) or open a selection list of names. Note: The search for names is case-insensitive. Example: ab, aB, Ab, AB are all assumed to be the same.	
Member	Name of a member or a range of names: use an asterisk (*) to determine a start value (see also <i>Valid Name Specifications</i> in <i>Direct Commands</i>) or open a selection list of names. Note: The search for names is case-insensitive. Example: ab, aB, Ab, AB are all assumed to be the same.	
Node Usage reports only:		
JCL Node	Name of a JCL node. Enter an asterisk (*) to select a name from a list.	
Execution Node	Name of an execution node or a range of names. Leave the field blank to select all names.	
Usage in master or active jobs	Jobs to be selected	
	Possible values:	
	M	Selects master jobs only (default).
	A	Selects active jobs only.
Symbol Search by Value reports only:		
Symbol	Symbol for which to search. Leave the field blank to select all symbols.	
Usage in: Master/Active	Symbols in which to search. Mark Master (default) and/or Active with any character to search in master symbols, active symbols, or in both.	
Mult. value index: From/To	Index range for symbols to which multiple values are assigned. From: Minimum index value in the range from 1 through 100 To: Maximum index value in the range from 1 through 100 or * for all multiple values available Default range: 1 to *	
At position	Position within the symbol value to be searched.	
	Possible input values:	
	*	Anywhere within the symbol value (default)

Field/Column	Explanation	
	<i>nn</i>	At the exact position: <i>nn</i> is any value in the range from 1 through 80
	at beginning	At the beginning of the symbol value
	E	At the end of the symbol value
Scan for	Scan value. Searches in a symbol value for a string of up to 60 characters	
Case sensitive	Mark this option with any character to perform a case-sensitive search.	
Undefined Objects reports only:		
This report inspects specified objects (symbol, symbol, table, resource, calendar, network, job and/or schedule) to determine which objects are referenced but not defined in your environment.		
Note: We recommend that you view this report in Entire Operations GUI Client which provides a graphical structure that helps interpret the output more easily.		
The columns in the GUI report are arranged in hierarchical order of object types from left (superior objects) to right. The corresponding items in the CUI report are arranged in a reverse order (subordinate object first).		
Resource Usage reports only:		
Type	Type of resource	
	Possible values:	
	U	Not reusable, quantitative
	R	Reusable, quantitative
	N	Not quantitative (absolute)
Initial Qty	Initial quantity Total amount of the resource defined to the system	
Used Qty	Used quantity Amount of resource currently used by running job	
Used by Owner	The active job, used by owner	
Used by Network	The active job, used by network	
Run	Run job	
Job	Active job	
Start usage	Date and time of the allocation (If this column is not displayed, use PF11 to scroll to the right of the screen.) See also <i>Date and Time Formats</i> .	

Examples of Cross Reference Reports

This section contains examples of all types of reports you can generate with the **Cross References** function:

- [Example of User Exit Usage](#)
- [Example of Symbol Table Usage](#)
- [Example of JCL Usage](#)
- [Example of Node Usage](#)
- [Example of Symbol Search by Value](#)
- [Example of Undefined Objects](#)
- [Example of Resource Usage](#)

Example of User Exit Usage

The following is an example of a cross reference report for **Exit Usage**:

Owner	Network	Version	Job	ExitLib	Exit
EXAMPLE	B60-FLOW		JOB-1-TEST	SYSEXT	EXT-CXML
EXAMPLE	E50-USRT		E50-J1-IC	SYSEORU	URD-UIC
INCIDENT	I5045008		I50450081	EOR-T531	I5045008
NATQA5	A		WILLI1	NOPNET	AAEVIT1
NATQA5	NET1432C		JOB-01	NOPNET	N1432X01
NATQA5	NET2240A		D002	NOPNET	N2240X01
NATQA5	NET2552A		J001	EOR-T543	ICDUM001
NATQA5	NET2552B		J001	EOR-T543	ICDUM002
NATQA5	NET2552C		J001	EOR-T543	ICDUM002
NOPALL	N2240A		D002	EOR-T543	N2240X01

All user exits of the specified type (here: input condition) and the jobs that reference these exists are listed in the report.

The columns contained in the report are described in [Fields and Columns: Cross References](#).

Example of Symbol Table Usage

The following is an example of a cross reference report for **Symbol Table Usage**:

```

MORE
30.06.12   Entire Operations   14:33:11
           Symbol Table Usage Page    1

           Owner      TESTBED
           Symbol Table NET01

Owner      Network    Version    Run      Job
-----
TESTBED    NET01
TESTBED    NET01                COLLECT1
TESTBED    NET01                EXE1
TESTBED    NET01                EXIT1
TESTBED    NET01                EXIT1-CHK
TESTBED    NET01                EXIT2
TESTBED    NET01                EXIT2-CHK
TESTBED    NET01                FIRST
TESTBED    NET01                LAST
TESTBED    NET01                PRQ-CHK
TESTBED    NET01                USR01-DEF
TESTBED    NET01                USR02-DEF
TESTBED    NET01                USR02-GINV
TESTBED    NET01                USR03-DEF

```

All networks and jobs that use the specified symbol table are listed in the report.

The columns contained in the report are described in [Fields and Columns: Cross References](#).

Example of JCL Usage

The following is an example of a cross reference report for **JCL Usage**:

```

MORE
06.05.15          Entire Operations          Page: 1
Owner: EXAMPLE    JCL Usage
Network   Version   Job       JCL Location/File/Member
-----
B60-FLOW           JOB-01    MAC/EOR-T541/B60-M02
B60-FLOW           JOB-012   MAC/EOR-T541/B60-M01
B60-FLOW           JOB-013   MAC/EOR-T541/B60-M01
B60-FLOW           JOB-014   MAC/EOR-T541/B60-M01
B60-FLOW           JOB-019   MAC/EOR-T541/B60-M01
B60-FLOW           JOB-02    MAC/EOR-T541/B60-M02
B60-FLOW           JOB-03    NAT/EOR-T531/B60-P01
B60-FLOW           JOB-04    MAC/EOR-T541/B60-M01
B60-FLOW           JOB-06    MAC/EOR-T541/B60-M02
    
```

All JCL files and referencing jobs are listed for the specified selection.

The columns contained in the report are described in [Fields and Columns: Cross References](#).

Example of Node Usage

The following is an example of a cross reference report for **Node Usage**:

```

MORE
06.05.15          Entire Operations          13:38:46
                   JCL node and/or Execution node usage      Page 1
Owner   Network   Version   Job       Job Type   JCL Node  Exec.Node
-----
EXAMPLE B60-FLOW           ABC3A     JOB                N0194
EXAMPLE B60-FLOW           JOB-01    JOB                N0194  N0082
EXAMPLE B60-FLOW           JOB-012   JOB                N0194  N0194
EXAMPLE B60-FLOW           JOB-013   JOB                N0194  N0194
EXAMPLE B60-FLOW           JOB-014   JOB                N0194  N0194
EXAMPLE B60-FLOW           JOB-015   DUM                N0194  N0194
EXAMPLE B60-FLOW           JOB-019   JOB                N0194  N0194
EXAMPLE B60-FLOW           JOB-02    JOB                N0194  N0082
EXAMPLE B60-FLOW           JOB-03    NAT                N0194  N0194
EXAMPLE B60-FLOW           JOB-04    JOB                N0194  N0194
EXAMPLE B60-FLOW           JOB-05    DUM                N0194  N0194
EXAMPLE B60-FLOW           JOB-06    JOB                N0194  N0194
EXAMPLE B60-FLOW           MILE1-TEST JOB                N0194
    
```

All nodes and referencing jobs are listed for the specified selection.

The columns contained in the report are described in [Fields and Columns: Cross References](#).

Example of Symbol Search by Value

The following is an example of a cross reference report for **Symbol Search by Value**:

```

MORE
09.05.15          ***** Entire Operations *****          11:38:26
                   Search Symbol by Value                      Page    1

```

Owner	Symbol Table	Symbol	MV F Network	Run Symbol Valu
EXAMPLE	ADMIN	CLASS	A	G
EXAMPLE	ADMIN	DATE	D	19970207
EXAMPLE	ADMIN	LOWER	L	
EXAMPLE	ADMIN	MSGCLASS	A	X
EXAMPLE	ADMIN	MULT	1 A	1
EXAMPLE	ADMIN	MULT	2 A	2
EXAMPLE	ADMIN	MULT	3 A	3
EXAMPLE	ADMIN	MULT	4 A	4
EXAMPLE	ADMIN	MULT	5 A	5
EXAMPLE	ADMIN	MULT	6 A	6
EXAMPLE	ADMIN	MULT	7 A	7
EXAMPLE	ADMIN	MULT	8 A	8
EXAMPLE	ADMIN	MULT	9 A	9
EXAMPLE	ADMIN	MULT	10 A	10
EXAMPLE	ADMIN	NUMBER	N	7
EXAMPLE	ADMIN	UPPER	U	
EXAMPLE	EX-ST-COMN	CLASS	A	G
EXAMPLE	EX-ST-COMN	CLASS-K	A	K

All symbols defined in the specified symbol table are listed in the report.

The columns contained in the report are described in [Fields and Columns: Cross References](#).

Example of Undefined Objects

The following is an example of a cross reference report for **List Undefined Objects**:

```

26.04.16          Entire Operations          11:58:20
                   List Undefined Objects

```

Network : NM-2, OW-SAG referenced by
Master Job : JM-1B NM-1B, OW-SAG

Condition : EXT-1

Symbol : TEST1 ,OW-SAG referenced by
Master Job : JM-1B NM-1B, OW-SAG

E0J Checking : --

All objects that are referenced but not defined in your environment are listed in the report.

In the example above, the following undefined objects are found for the owner OW-SAG:

- The network NM-2 is referenced in the input condition EXT-1 of the job JM-1B in the network NM-1B.
- The symbol TEST1 is referenced in the EOJ checking of the job JM-1B in the network NM-1B.

The fields contained in the report are described in *Fields and Columns: Cross References*.

Example of Resource Usage

The following is an example of a cross reference report for **Resource Usage**:

----- Entire Operations: Resource Usa -----							
Resource	Type	Initial Qty	Used Qty	used by Owner	Network	Run	Job
BOA-RES	R	20.00	5.00	XSETAB01	IMPXP2	2635	JOB2
BOA-RES	R	20.00	5.00	XSETAB01	IMPXP2	2636	JOB2
BOA-RES	R	20.00	5.00	XSETAB01	IMPXP2	2638	JOB2
BOA-RES	R	20.00	5.00	XSETAB01	IMPXP2	2639	JOB2
DEA1-KEEP-01	R	100.00	3.00	SN	RES-DEA1	21864	J03-K
DEA1-KEEP-01	R	100.00	3.00	SN	RES-DEA1	21865	J03-K
DEA1-KEEP-01	R	100.00	3.00	SN	RES-DEA1	21866	J03-K
DEA1-KEEP-01	R	100.00	3.00	SN	RES-DEA1	21867	J03-K
DEA1-KEEP-01	R	100.00	3.00	SN	RES-DEA1	21868	J03-K
DEA1-KEEP-01	R	100.00	3.00	SN	RES-DEA1	21869	J03-K
DEA1-KEEP-01	R	100.00	3.00	SN	RES-DEA1	21870	J03-K
DEA1-KEEP-01	R	100.00	3.00	SN	RES-DEA1	21871	J03-K
DEA1-KEEP-01	R	100.00	3.00	SN	RES-DEA1	21872	J03-K
DEA1-KEEP-01	R	100.00	3.00	SN	RES-DEA1	21873	J03-K
DEA1-KEEP-01	R	100.00	3.00	SN	RES-DEA1	21874	J03-K
NPR-INSTALL	R	1.00	1.00	NOPALL	N2029D	85	J00-D
***** m o r e *****							
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---							
End				Down		Left Right Menu	

All resources and referencing active jobs are listed for the specified selection.

The columns contained in the report are described in *Fields and Columns: Cross References*.

Generating Cross Reference Reports in Batch

This section describes how to generate cross reference reports in batch mode.

- [User Exit Cross Reference Reports in Batch](#)
- [Symbol Table Cross Reference Reports in Batch](#)
- [JCL Cross Reference Reports in Batch](#)
- [Node Cross Reference Reports in Batch](#)
- [Undefined Objects Reports in Batch](#)
- [Resource Usage Reports in Batch](#)

User Exit Cross Reference Reports in Batch

➤ **To produce the user cross reference report in batch**

- Run `XRUR - - P` in a standard batch Natural job with LFILES 131 and 216 assigned and printer file 1 correctly defined.

The program has no input parameters.

Symbol Table Cross Reference Reports in Batch

➤ **To produce the symbol table cross reference report in batch**

- Run `XRSYT - - P` in a standard batch Natural job with LFILES 131 and 216 assigned and printer file 1 correctly defined.

Syntax:

```
XRSYT--P
owner-name,symbol-table-name,symbol-table-version,active-job
```

The program does 1 input:

- Owner, Symbol Table, Symbol Table Version, Active Jobs

The following examples assume the Natural parameters `IM=D, ID=, :`

```
XRSYT - - P  
PROD, ABC*, *, M
```

This will produce a list of master symbol tables of owner PROD with names starting with ABC.

JCL Cross Reference Reports in Batch

➤ To produce the JCL cross reference report in batch

- Run `XRJCL - - P` in a standard batch Natural job with LFILEs 131 and 216 assigned and printer file 1 correctly defined.

Syntax:

```
XRJCL--P  
file/natlib,member,jcl-location  
owner-name,network-name,network-version
```

The program does 2 inputs:

1. File/Natlib, Member, JCL Location
2. Owner, Network, Network Version (for future use)

The following examples assume the Natural parameters `IM=D, ID=, :`

Example 1:

```
XRJCL - - P  
PROD1.JOBLIB, IEB*  
SYS*, W, *
```

This will produce a list of all jobs in networks with names less than W under owners beginning with SYS that use JCL members beginning with IEB in data set PROD1.JOBLIB.

Example 2:

```
XRJCL - - P  
SYS*, *, NAT  
*, *, *
```

This will produce a list of all jobs which use JCL in Natural libraries beginning with SYS.

Node Cross Reference Reports in Batch

➤ To produce the node cross reference report in batch

- Run `XRNDU--P` in a standard batch Natural job with LFILES 131 and 216 assigned and printer file 1 correctly defined.

Syntax:

```
XRNDU--P
jcl-node,execution-node,master/active-jobs
owner-name,network-name,network-version
```

The program does 2 inputs:

1. JCL Node, Execution Node, active Jobs Master/Active
2. Owner, Network, Network Version

The following examples assume the Natural parameters `IM=D, ID=, :`

Example:

```
XRNDU--P
517,517,M
SYS*,W,*
```

This will produce a list of master jobs in networks with names less than `W` under owners beginning with `SYS` that use JCL node 517 and execution node 517.

Undefined Objects Reports in Batch

Note that a version name (`void`) might occur for an object, if no version usage is defined for the object that is valid on the determination date. The default value of the determination date is the start of the users Entire Operations session.

➤ To list undefined objects in batch

- Run `XRNDF--P` in a standard batch Natural job

Syntax:

XRNDF—P

Specification of the object types to be inspected:

P-CHECK-SYMBOL	'X' = search for undefined symbols.
P-CHECK-SYMTAB	'X' = search for undefined symbol tables.
P-CHECK-NETWORK	'X' = search for undefined networks.
P-CHECK-JOB	'X' = search for undefined jobs.
P-CHECK-RESOURCE	'X' = search for undefined resources.
P-CHECK-SCHEDULE	'X' = search for undefined schedules.

Field	Explanation
Owner	Owner name. *, > and < wildcards are allowed. Use * for all owners.
Network	Network name. *, > and < wildcards are allowed. Use * for all networks.
Network Version	Network Version name. *, > and < wildcards are allowed. Use * for all network versions.

Resource Usage Reports in Batch

➤ To produce the resource usage cross reference listing in batch

- Run `XRRES - -P` in a standard batch Natural job with LFILEs 131 and 216 assigned and printer file 1 correctly defined.

Syntax:

```
XRRES - -P  
Resource-name  
owner-name, network-name, network-version
```

The program does two inputs:

1. Resource
2. Owner, Network, Network Version

The following examples assume the Natural parameters `IM=D, ID=;`:

```
XRRES - - P  
RES*  
PROD,ABC*,*
```

This will produce a list of all jobs of owner PROD with names starting with ABC using a resource beginning with RES.

XVI

API Routines

This section describes the application programming interface (API) routines provided by Entire Operations for batch and online processing.

[Using an API Routine](#)

[API Routines for Entire Operations](#)

72

Using an API Routine

- API Usage Rules and Restrictions 780
- Invoking the API Routines 781
- Example of Use of API Routines 781

An Entire Operations API routine is supplied as Natural subprogram with the name `NOPxxxxN` in the Natural SYSEOR system library.

The naming convention is:

`NOPUxxxy`

where `NOPU` means Entire Operations User API Routine, `xxx` is the program name (sometimes containing a version number) and `y` is the program type (`N` - subprogram, `P` - program).

To use the Entire Operations API routines, define at least the SYSEOR and SYSSAT libraries as Steplibs.

You execute an API subprogram with the Natural `CALLNAT` statement. The `CALLNAT` statement is described in detail in the Natural *Statements* documentation.

API Usage Rules and Restrictions

The following general rules apply when using an API:

- We recommend that you do not rename API routines because they are also used internally by Entire Operations.
- The routines delivered with the previous Entire Operations version are still supported (except for special notice, and only if they are delivered again with the current SYSEOR library). Convert their calls to the newer version when you modify the calling programs. Change all relevant `CALLNAT` statements and adapt the API parameter list. The complete range of functions and functionality is only available with the latest API version.
- Entire Operations internal data structures are subject to change between versions. The usage of an outdated API may cause data inconsistencies, unexpected Monitor failures or abnormal termination of Entire Operations.
- With newer API versions, new fields might have been added. You may leave these fields blank or zero if their description says “for future use”. These fields are intended to become meaningful with a future Entire Operations version.
- No `END TRANSACTION` statement will be executed inside of Entire Operations APIs.
- If you invoke APIs from within an Entire Operations user exit, you must not code any `END TRANSACTION` statement within the exit. Refer to the user exit coding rules.
- If you invoke APIs from anywhere else than an Entire Operations user exit, you must code an `END TRANSACTION` statement in an appropriate location after the `CALLNAT` statement to close Entire Operations transaction logic.
- You can change the parameter names, but not the parameter format, number and order.

API Access Restrictions

If Natural Security is installed, you can restrict Natural RPC access to particular APIs. For more information, see the **Service Protection** option of an RPC server profile described in the section *Allowing/Disallowing Services* in the *Natural Security* documentation.

Invoking the API Routines

First, the user application must establish a link to the Entire Operations system files. This can be done in any of the following ways:

The link is already established if the user application startup is extended with either of the following:

```
NTLFILE ID=216,DBID=eor-sysf1-dbid,FNR=eor-sysf1-fnr
NTLFILE ID=131,DBID=sat-log-dbid,FNR=sat-log-fnr (in parameter module)
```

or

```
LFILE (216,eor-sysf1-dbid,eor-sysf1-fnr)
LFILE (131,sat-log-dbid,sat-log-fnr) (as dynamic parameter)
```

Example of Use of API Routines

The Natural program `NOPUT1-P` can be invoked to find out how API routines react to real data. It shows how to code the calls of the supplied routines, and how to check the return codes.

The source and the cataloged object of `NOPUT1-P` are supplied in the SYSEOR system library.

When you execute the program `NOPUT1-P`, you can select and test the required API routine from a screen similar to the example below:

Entire Operations - API Routines

Please Select:

Network / Job Status	ST	NOPUST3N
Conditions	CO	NOPUCN3N
Resource Master	RM	NOPURS1N
Resource Usage	RU	NOPURE2N
Symbols	SY	NOPUSY6N
Calendars/Schedules	CS	NOPUCS1N
Activation	AC	NOPUAC5N
Milestone	MI	NOPUMI1N
Subnetwork	SN	NOPUSN2N
Job Import	JI	NOPUJI4N
Version Info	VE	NOPUVI2N
Symbol Prompt Texts	SP	NOPUSP3N
Job Schedule	SC	NOPUJS2N
Log Monitor Activity	LM	NOPMLA1N
Active Job Statistics	AS	NOPUAS1N
Message Text	MT	NOPUMT3N

====> _ (blank or . or PF3 ==> End)

73

List of Available Entire Operations API Routines

▪ EORUCB1N: Check Use of BS2000 User IDs	784
▪ NOPFB2-N: Generate SYSOUT File Names for BS2000	784
▪ NOPMLA1N: Start and Stop the Monitor Activity Log	787
▪ NOPU--1N: Read Network Available to a Specific User	788
▪ NOPU--2N: Return all Usable Symbol Tables for a Network	789
▪ NOPU--3N: Get Correlation ID for an Activated Network	790
▪ NOPU--4N: Store New Event in Entire Operations System File	790
▪ NOPUAC5N: Activate Job Networks or Jobs	791
▪ NOPUAS1N: Retrieve Numbers of Active Jobs in Defined Status Ranges	794
▪ NOPUCN3N: Access Entire Operations Conditions	796
▪ NOPUCS1N: Access Calendars and Schedules	797
▪ NOPUJ4N: Import Operating System Jobs into the Active Queue	801
▪ NOPUJS2N: Job Schedule Inquiry and Modification	802
▪ NOPULW9N: Write Messages to System Automation Tools Log	804
▪ NOPUMI1N: Set/Reset Text Milestones in Master and Active Jobs	805
▪ NOPUMT3N: Expand Message Texts	806
▪ NOPUNI1N: Invalidate Entire System Server Node Table Entries	808
▪ NOPUNX1N: Entire System Server Calls to Access UNIX and Windows Files	808
▪ NOPURE2N: Handle Resource Allocations	816
▪ NOPURS1N: Access Entire Operations Master Resources	819
▪ NOPUSN2N: Inquire Calling Job or Called Network for Subnetworks	821
▪ NOPUSP3N: Display Long Texts for Symbol Prompting	822
▪ NOPUST3N: Inquire Network and Job Status, Symbol Table	824
▪ NOPUSY6N: Access Entire Operations Symbols	828
▪ NOPUVI2N: Obtain Entire Operations Version Information	833
▪ NOPUXD1N: Maintain End-of-Job User Exits for a Network	834
▪ NOPUXI0N: Add Input Condition to an Activated Job	835

EORUCB1N: Check Use of BS2000 User IDs

You can use the exit `EORUCB1N` contained in the library `SYSEORU` for a user-defined check, if Natural users are allowed to use BS2000 user IDs.

Meaning of the input and output fields:

Parameter Description

Parameter	Format	Use	
P-RC	N4	out	Return code (not evaluated, should be 0).
P-RT	A70	out	Return text (not evaluated).
P-USER	A8	in	*USER in Natural.
P-BS2000-USERID	A8	in	BS2000 user ID.
P-ACCOUNT	A8	in	BS2000 account number.
P-SUBMIT-PSWD	A8	in	BS2000 submit password.

In the following you find the example of the exit `EORUCB1N` delivered with Entire Operations which you can adapt according to your need. The exit always returns `P-RC=0` (authorized).

Example

```
* NOPFB2-N  
END
```

NOPFB2-N: Generate SYSOUT File Names for BS2000

The exit `NOPFB2-N` may be copied into the library `SYSEOR` to enable the user-defined generation of `SYSOUT` file names in BS2000. Only one instance of this exit exists in the system and it is executed only if it is present. Otherwise, standard name generation is used.

This exit is called with the parameter list `NOPXPL-A`, so the first line of the exit must be:

DEFINE DATA PARAMETER USING NOPXPL-A

This section covers the following topics:

- [Parameter Description](#)
- [Rules for Exit Coding](#)
- [Use with Work File Deletion](#)
- [Example](#)

Parameter Description

Meaning of the parameters in [NOPXPL-A](#):

Parameter	Format	Use	
P-CALL-PLACE	A3	in	Constant FSB (file name SYSOUT BS2000).
P-RC	N4	out	Return code:
			0
			Function OK (name generation OK)
			1
			A name could not be generated. The default name routine of Entire Operations is to be called.
			2
			A name could not be generated. The action (activation) is to be cancelled.
			If the activation is cancelled by means of the name routine, a corresponding log entry will be written.
P-RT	A66	out	Return code (not evaluated).
P-OWNER	A10	in	Network owner.
P-NETWORK	A10	in	Job network.
P-JOB	A10	in	Job.
P-RUN	P13	in	Run number.
P-ACTIVATION-TIME	T	in	Activation time of the network.
P-EXECUTION-NODE	N3	in	Entire System Server execution node.
P-EXECUTION-OPSYS	A8	in	Operating system of Entire System Server node.
P-SYMBOL-TABLE	A10	in	Name of the defined symbol table.
P-FSB-OBJECT-TYPE	A5	in	File type:
			SO-C
			Current SYSOUT file.
			SO-P
			Previous SYSOUT file.
			SO-P1
			Previous SYSOUT file without user ID.
			SO-A
			All SYSOUT files of a job.
P-FSB-USERID	A8	in	BS2000 user ID.
P-FSB-CATID	A4	in	BS2000 Cat ID.
P-FSB-SUFFIX	A2	in	BS2000 suffix.

Parameter	Format	Use
P-SYSOUT-FILE	A54	out Generated file name.

Rules for Exit Coding

- File names must be created which are unique system-wide. Otherwise, the proper running of Entire Operations is no longer possible.
- File names which are unique system-wide can be produced by using the fields P-OWNER, P-NETWORK, P-RUN, P-JOB and P-FSB-SUFFIX simultaneously.
- You must be able to process all of the object types defined above.
- Except for the object type SO-P1, a fully-qualified file name must be created which includes the BS2000 user ID.
- A wildcard file entry valid for all SYSOUT files of an active job must be created for the type SO-A. This type is used for deleting files.
- The field P-FSB-SUFFIX is transferred and must be used to differentiate SYSOUT files when jobs are repeated. The caller ensures that the suffix for the currently active job is unique. For the object type SO-C, this field is blank.



Note: If no file name is returned, then standard name generation is used.

Use with Work File Deletion

Even in case of work file deletion (in connection with deactivation or cleanup) this exit is called, if it exists at all. The following will apply in this context:

- At any rate, an attempt is made to also delete work files, which correspond to the default name conventions of Entire Operations, independent of the existence of the exit.
- It is presumed that the exit will generate the run number with exactly 5 digits, which are enclosed by dots:
 - aaa.00000.bbb (example: aaa.01234.bbb)

For the Entire Operations file search, this pattern is replaced by a wildcard:

- aaa.00000.bbb --> aaa.*.bbb
- If the name syntax of the exit is modified, files cannot be found any longer, the names of which have been generated before the modification. It is not recommended to exchange the exit, while active jobs are running in the operating system.

Example

```

* NOPFB2-N
* EXAMPLE EXIT FOR SYSOUT FILE NAME
*
* MODIFICATIONS:
* 29.10.08 (139240) SYSOUT FILE NAME EXIT SN210
* -----
DEFINE DATA PARAMETER USING NOPXPL-A
END-DEFINE
* -----
COMPRESS P-OWNER P-NETWORK P-RUN P-JOB
  INTO P-SYSOUT-FILE LEAVING NO SPACE
IF P-FSB-OBJECT-TYPE NE 'SO-P1'
* -- ALL OBJECT TYPES EXCEPT SO-P1 M U S T HAVE A BS2000 USERID --
  COMPRESS '$' P-FSB-USERID '.' P-SYSOUT-FILE
  INTO P-SYSOUT-FILE LEAVING NO SPACE
END-IF
IF P-FSB-OBJECT-TYPE EQ 'SO-A'
* -- ALL SYSOUT FILES FOR A JOB --
  COMPRESS P-SYSOUT-FILE '*'
  INTO P-SYSOUT-FILE LEAVING NO SPACE
ELSE
  COMPRESS P-SYSOUT-FILE '.S' P-FSB-SUFFIX
  INTO P-SYSOUT-FILE LEAVING NO SPACE
END-IF
RESET P-RC /* OK
END

```

NOPMLA1N: Start and Stop the Monitor Activity Log

You can start and stop the monitor activity log using the following calls from your Natural application.

```

CALLNAT 'NOPMLA1N'
P-FUNCTION

```

Parameter Description

Parameter	Format	Use
P-FUNCTION	A1	in Function code:
		Y Starting the monitor activity log.
		N Stopping the monitor activity log.

The call of this API has the same effect as setting the field Log Monitor Activity in the Monitor Defaults.

This enables you to automatically start and stop the monitor activity log at particular times or on account of specific conditions.

NOPU--1N: Read Network Available to a Specific User

You can get a list of networks owned or granted to an user using the following call from your Natural application.

```
CALLNAT 'NOPU--1N'
P-USER
P-SELECTION
P-RC
P-RESULT-GRANGED-NETWORKS (*)
```

The parameters have the following meaning:

Parameter	Format	Use
P-USER	A8	in User name.
P-SELECTION	A10	in Selection Criteria.
		Select by P-SELECTION (combination of values possible!):
		O Networks of Owner.
		G Owner Granted Networks.
		A Active Networks only. Note: only usable together with O,G,U.
		U User Granted Networks.
P-RC	N3	1 User not defined.
		99 Natural runtime error.
P-RESULT-GRANGED-NETWORKS		

Parameter	Format	Use	
Note: This is a dynamic structure consisting of :			
P-OWNER	A10	out	Owner of Network.
P-NETWORK	A10	out	Network Name.
P-NETWORK-VERSION	A10	out	At least "" and "(current)".
P-DESCRIPTION	A50	out	Short description of Network .

NOPU--2N: Return all Usable Symbol Tables for a Network

You can get a list of usable symbol tables for a given Network using the following call from your Natural application.

```
CALLNAT 'NOPU--2N'
  P-OWNER
  P-NETWORK
  P-NETWORK-VERSION
  P-RC
  P-RESULT-SYMBOL-TABLES (*)
```

The parameters have the following meaning:

Parameter	Format	Use	
P-OWNER	A10	in	Owner Name.
P-NETWORK	A10	in	Network Name.
P-NETWORK-VERSION	A10	in	Network Version.
P-RC	N3	1	Error when determining tables.
		99	Natural runtime error.
P-RESULT-SYMBOL-TABLES			
Note: This is a dynamic structure consisting of:			
P-SYMBOL-TABLE_TYPE	A2	out	Symbol Table Type.
P-SYMBOL-TABLE_OWNER	A10	out	Symbol Table Owner.
P-SYMBOL-TABLE	A10	out	Table Name.
P-SYMBOL-TABLE_VERSION	A10	out	Symbol Table Version.

NOPU--3N: Get Correlation ID for an Activated Network

You can retrieve the correlation ID for a specific Network using the following call from your Natural application.

```
CALLNAT 'NOPU--3N'
P-OWNER
P-NETWORK
P-RUN-NUMBER
P-RC
P-CORRELATION-SUBSYS
P-CORRELATION-ID
P-CORRELATION-TYPE
```

The parameters have the following meaning:

Parameter	Format	Use
P-OWNER	A10	in Owner Name.
P-NETWORK	A10	in Network Name.
P-RUN-NUMBER	I4	in Run Number.
P-RC	N3	1 Activated Network does not exist.
		99 Natural runtime error.
P-CORRELATION-SUBSYS	A32	out Correlation subsys.
P-CORRELATION-ID	A36	out Correlation subsys.
P-CORRELATION-TYPE	A32	out Correlation type.

NOPU--4N: Store New Event in Entire Operations System File

You can store an event record for a given network run using the following call from your Natural application.

```
CALLNAT 'NOPU--4N'
P-EVENT-TYPE
P-EVENT-VERSION
P-EVENT-DATA (*)
P-OWNER
P-NETWORK
P-NETWORK-RUN
P-RC
P-ERROR-NR
```

The parameters have the following meaning:

Parameter	Format	Use		
P-EVENT-TYPE	A32	in		
P-EVENT-VERSION	A5	in		
P-EVENT-DATA	A80/1:19			
P-OWNER	A10	in	Owner of Network .	
P-NETWORK	A10	in	Network.	
P-NETWORK-RUN	I4	in	Run Number.	
P-RC	N4	out	1	Network run does not exist.
			99	Natural run time error.
P-ERROR-NR	N7	out	Only filled in case of run time error with contents *ERROR-NR.	

NOPUAC5N: Activate Job Networks or Jobs

You can activate job networks or jobs from your Natural application. The network of job definitions must exist in Entire Operations. Use:

```
CALLNAT 'NOPUAC5N'
      FUNCTION RC OWNER NETWORK NETWORK-VERSION JOB SYMTAB SYMTAB-VERSION START-TIME ↵
RUN
```

This section covers the following topics:

- [Parameter Description](#)
- [Activation with Symbol Modification](#)
- [Repetition of an Active Network](#)

Parameter Description

Parameter	Format	Use		
FUNCTION	A1	in	Function code:	
			A	Activate , no hold for symbols.
			1	Like A, but time frame as defined in schedule is used.
			C	Change start time.
			D	RUN > 0: Deactivate network, RUN = -1: Delete master-network.
			H	Activate , but hold task until released.

Parameter	Format	Use
		<p>I Set run number initial value.</p> <p>The next network run will use the next run number available after the number entered.</p> <p>The run number must be between 0 and the highest run number permitted.</p>
		J Modify the earliest start time, latest start time and deadline time of an active job. The three time stamps are shifted with the same offset.
		K Modify the earliest start time of an active job only. The latest start time and deadline time will not be modified.
		R Release activation.
		2 Like R, but time frame as defined in schedule is used.
		S Repetition of an active job.
		T Repetition of an active network.
		X Network or job existence test only.
RC	N03	<p>out Return code:</p> <p>0 OK</p> <p>1 Owner, network, job not found.</p> <p>2 Activation entry not found.</p> <p>3 Network not active.</p> <p>4 Job cannot not be restarted. For the function S (repetition), this return code is issued if there is a SYSOUT copy in progress.</p> <p>5 Run number has already been used.</p> <p>6 Run number not in valid range.</p> <p>7 No free run number.</p> <p>8 Active objects existing.</p> <p>9 Current version can not be determined.</p> <p>10 Version default usage found. This return may be issued at a deletion attempt.</p> <p>101 Invalid function code.</p> <p>102 Parameters missing.</p> <p>111 Symbol table name contained reserved prefix =EOR=.</p> <p>121 Owner does not exist.</p> <p>122 Network version is required.</p>

Parameter	Format	Use
		901 OK; subnetwork symbol prompting.
DBENV	A10	in Database environment (for future use). OPTIONAL(*) parameter.
OWNER	A10	in Owner of the job network. If a job is placed in Hold status by means of the function H, this will be written to the log.
NETWORK	A10	in The job network.
NETWORK-VERSION	A10	in Version name of the job network or (current)
JOB	A10	in Job. If empty, whole network is activated, deactivated or tested.
SYMTAB	A10	in The symbol table.
SYMTAB-VERSION	A10	in Symbol table version.
START-TIME	T	in If zero (0), the network is activated immediately. Otherwise, the network is activated at the designated time. The field must contain both date and time.
		out Start time actually set.
RUN	I4	in Run number used by Entire Operations. Functions A, H: <ul style="list-style-type: none"> When activating a single job, a existing run number can be entered. This should only be used for post activations of an active network. Functions C, D, R, S: <ul style="list-style-type: none"> must be an existing run number of the job network. Function X: <ul style="list-style-type: none"> If 0, the master network is tested. Otherwise, an active network with this run number is tested.
		out Functions A, H: <ul style="list-style-type: none"> If 0, the next free run number is assigned. If not 0, this number is used; if this number is still active, the next free number in ascending order is used.



Note: You can use this function from outside Entire Operations, as well as from user exits within Entire Operations. This function is comparable to the manual activation of networks or jobs in the Entire Operations online system.

Activation with Symbol Modification

➤ To activate a job or network and pass specific symbol values to this activation, proceed as follows

- 1 Activate the network and keep it in hold, with function H.
- 2 Use the returned run number to set symbols in the active symbol table(s) for this run. You must call the routine **NOPUSY6N** (see the subsection *Accessing Entire Operations Symbols*).
- 3 After setting the symbol(s), release this activation with the R function.



Note: Active symbols, which are set here, will not be overwritten by a subsequent complete symbol table activation.

Repetition of an Active Network

The field P-JOB may remain empty.

- If P-JOB is empty, all executed jobs of the active network will be repeated.
- If P-JOB contains a valid job name, the active network will be repeated, but the given job will be repeated as “dummy due to repetition”.

The repetition also works correctly for jobs of NET type: subnetworks are also repeated, with their original run number.

The conditions NET-BEGIN, NET-END, NET-END-NOTOK will be reset before the repetition.

This API function enables you to repeat whole active networks, including subnetworks. Active conditions which are already set or reset will be backed out.



Caution: You should exercise caution, if the API NOPUAC5N function T is invoked for a network, in which jobs are currently executing. *These will not be repeated.* We recommend that you use this API function only for active networks, in which no job is currently executing.

NOPUAS1N: Retrieve Numbers of Active Jobs in Defined Status Ranges

This section covers the following topics:

NOUCN3N: Access Entire Operations Conditions

You can handle conditions using the following statement in your Natural application:

```
CALLNAT 'NOUCN3N'
  P-FUNCTION P-RC P-OWNER P-NETWORK P-CONDITION P-RUN-FROM P-RUN-TO
  P-TIME-FROM P-TIME-TO P-CONFIRM P-JCL-CHECK
```

This section covers the following topics:

- [Parameter Description](#)
- [Description of the Function Codes](#)

Parameter Description

Parameter	Format	Use	
P-FUNCTION	A1	in	Function code:
		R	Reset a condition.
		S	Set a condition.
		T	Test a condition.
P-RC	N03	out	Return code:
		0	Function OK.
		1	Condition not found.
		2	Time or run required for reset.
		3	Invalid run number range.
		101	Invalid function code.
		102	Condition name missing.
		104	Network name missing.
121	Owner does not exist.		
P-OWNER	A10	in	Owner of the job network / condition.
P-NETWORK	A10	in	Job network.
P-CONDITION	A20	in	Condition name.
P-RUN-FROM	I4		Corresponding run number.
		in	Absolute condition: -1
		out	Function T and field empty: Last run number found is returned.
P-RUN-TO	I4	in	Run number to (end of interval). If zero, only P-RUN-FROM will be handled.
P-TIME-FROM	T		Starting time of condition.
		in	

Parameter	Format	Use
		out Function T and field empty: Time belonging to run number is returned.
P-TIME-TO	T	in End time of condition.
P-CONFIRM	A1	in Confirm deletions: Y yes, N no
P-JCL-CHECK	A1	in If this field contains C, the condition is treated like one which is used during a JCL check.

Description of the Function Codes

Meaning of the function codes:

Code	Description
R	Reset a condition. The date or the run number must be specified. If a run number is specified, the date is not considered. If a date is specified, the deletion of several conditions with the same date, but different run numbers might occur.
S	Set a condition. The date and/or run number are optional.
T	Test a condition. If a run number is specified, the time frame is not considered. If no time frame and no run number are specified, each condition with the specified name matches. If no run number is specified, the last run number for the time frame up to the current time is returned. If no time frame was given, the time belonging to this run number is returned in P-TIME-FROM.

NOPUCS1N: Access Calendars and Schedules

This section covers the following topics:

- [Accessing Dates within Calendars or Schedules](#)

▪ [Example for Using NOPUCS1N](#)

Accessing Dates within Calendars or Schedules

You can access dates within calendars or schedules using the following statement in your Natural application:

```
CALLNAT 'NOPUCS1N'
      P-FUNCTION P-RC P-OBJECT-TYPE P-OWNER P-OBJECT P-DATE-A8
```

Meaning of the parameters:

Parameter	Format	Use	
P-FUNCTION	A1	in	Function code:
		A	Delete a calendar or schedule.
		B	Add calendar.
		C	Determine the calendar used.
		D	Determine the schedule used.
		N	Determine the next set date, from a start date.
		P	Determine the previous set date, from a start date.
		R	Reset a date. Date will be removed from calendar or schedule.
		S	Set a date. Date will be added to calendar or schedule.
		T	Test a date. If date exists within calendar or schedule: RC contains 0, otherwise 1.
		1	Set all dates in the whole table (calendar or schedule).
		2	Reset all dates in the whole table (calendar or schedule).
P-RC	N03	out	Return code:
		0	Function OK, or date is set. Note: For calendars, this means "working day".
		1	Date is not set. Note: For calendars, this means "holiday".
		2	Object not found.
		3	Object used in definitions.
		4	Object already existing.
		101	Invalid function code.

Parameter	Format	Use
		102 Parameter(s) missing.
		103 Invalid parameter combination.
		104 Invalid object type.
		105 Invalid date.
		106 Table of explicit dates full.
		121 Owner does not exist.
P-OBJECT-TYPE	A1	in Object type:
		N Network
		C Calendar
		H Schedule history
		S Schedule
P-OWNER	A10	in Owner of object.
		out Functions C and D: owner of determined object.
P-OBJECT	A10	in Name of object: owner name resp. schedule name If P - FUNCTION is N, P, or T, and if a calendar is expected as object: ■ You may use the reserved calendar name =EOR=WORK7. This pseudo-calendar is valid for any year and has all days defined as working days. ■ The value of P - OWNER is meaningless in this case.
		out Function codes C and D: name of determined object.
P-DATE-A8	A8	in Date in format: YYYYMMDD. See also Date and Time Formats . ALL in this field means "all years". This value applies for deletions.
		out Function codes N and P: the date found.

**Notes:**

1. Only function T (test a date) is allowed for object type H (schedule history)
2. Schedule history is usually available for the current year and previous two years.
3. Schedule dates can be set for the current year and the next year.
4. In calendars, a set date means "working day", and a non-set date means "holiday". The functions N and P work for a range of years. Limitations:
For schedules, only the current and the previous year are available.
5. Calendars must be defined for all years in which a search is to be performed.

6. Function A deletes calendar/schedule master when P-OBJECT-TYPE='C' or P-OBJECT-TYPE='S' respectively. In case calendar/schedule is for the current year, and if is referenced in schedule/network the P-RC value 3 (Object used in definitions) is returned. Calendars for past years will be deleted without reference check.
7. When a calendar is deleted (P-FUNCTION='A' and P-OBJECT-TYPE='C'), the value of P-DATE-A8 shall be either in form YYYY or YYYYMMDD or constant ALL. Value ALL means: the calendar will be deleted for all defined years.

Example for Using NOPUCS1N

The program calculates the first Monday of each month in 2008 and subsequently enters the network REQUEST/BFA001 for the calculated days in the schedule.

```

* BFA001
* DETECT FIRST MONDAY IN MONTHS
*
DEFINE DATA
LOCAL
1 #YYYYMMDD (A8)
1 REDEFINE #YYYYMMDD
  2 #YYYY (N4)
  2 #MM (N2)
  2 #DD (N2)
1 #D1 (D)
1 #WD (A1)
1 #OWNER (A10) INIT <'REQUEST'> /* Owner
1 #OBJECT (A10) INIT <'NET001'> /* Network
1 #RC3 (N3)
END-DEFINE
*
#YYYY := 2010 /* Year
F1. FOR #MM = 01 TO 12
  RESET #DD
  R1. REPEAT
    ADD 1 TO #DD
    MOVE EDITED #YYYYMMDD TO #D1 (EM=YYYYMMDD)
    MOVE EDITED #D1 (EM=0) TO #WD
    WRITE #D1 (EM=YYYYMMDD) #WD
    UNTIL #WD = '1' /* R1.
  END-REPEAT /* R1.
  WRITE #D1 (EM=YYYYMMDD' 'N(10))
*
* CALL NOP SCHEDULE API TO SET THESE DAYS
*
CALLNAT 'NOPUCS1N'
'S' #RC3 'S' #OWNER #OBJECT #YYYYMMDD
WRITE #YYYYMMDD #RC3
END-FOR /* F1.

```

```
END TRANSACTION
END
```

NOPUJI4N: Import Operating System Jobs into the Active Queue

You can activate a job in Entire Operations together with an existing operating system job number. This is possible if the job was submitted by any application and set to HOLD status.

Use the following statement:

```
CALLNAT 'NOPUJI4N'
  P-FUNCTION P-RC P-OWNER P-NETWORK P-JOB P-RUN P-EXECUTION-NODE P-SYSOUT-NODE
  P-SYSOUT-FILE P-BS2000-MONJV P-JOB-ID P-OS-JOB-NAME
```

Parameter Description

Parameter	Format	Use		
P-FUNCTION	A1	in	Function code:	
			I	Import job, without prerequisite check.
			J	Import job, with prerequisite check. The new active job is created "with context", i.e., with input and output condition definitions.
P-RC	N3	out	Return code:	
			0	Function OK.
			101	Invalid function code.
			102	Parameter(s) missing.
			103	P-JOB-ID value invalid
			121	Owner does not exist.
The parameters P-OWNER, P-NETWORK, P-JOB and P-RUN are optional. If any of these are empty, IMPORT is used instead as owner, network and/or job name. Entire Operations will try to find a job definition for these values. If a definition is found, the new active job entry is supplied from it. Otherwise, an activation is performed.				
P-OWNER	A10	in	Optional.	
P-NETWORK	A10	in	Optional.	
P-JOB	A10	in	Optional.	
P-RUN	I4	in	Optional. If P-RUN is set to zero (0), Entire Operations assigns a new run number; otherwise, the job is activated under the run number given. Master jobs for imports with	

Parameter	Format	Use	
			run number must be defined with the special type R to prevent automatic activation.
P-EXECUTION-NODE	I4	in	Required. Note: Format was changed. Previous API version used was N3 in this case.
P-SYSOUT-NODE	I4	in	Optional. Necessary only if P-SYSOUT-FILE is specified, and if P-SYSOUT-NODE is different from P-EXECUTION-NODE.
P-SYSOUT-FILE	A250	in	Required for BS2000, UNIX, Windows. File which contains the SYSOUT.
P-BS2000-MONJV	A250	in	Optional. BS2000 monitor job variable. Specify this parameter only for BS2000, and only if the job was started in conjunction with a monitor job variable.
P-JOB-ID	A10	in	Required. The Job Id can be passed leftbound in this field, regardless whether it is numeric or alphanumeric. Note: Format was changed. Previous API version used was A5 in this case.
P-OS-JOB-NAME	A8	in	Optional; from operating system.

NOPUJS2N: Job Schedule Inquiry and Modification

You can inquire and modify a job schedule by using the following statement in your Natural application:

```
CALLNAT 'NOPUJS2N'
  P-FUNCTION P-RC P-DBENV P-OWNER P-NETWORK P-NETWORK-VERSION
  P-RUN P-JOB P-AVERAGE-TIME P-EARLIEST-START
  P-LATEST-START P-DEADLINE
```

Parameter Description

Parameter	Format	Use		
P-FUNCTION	A1	in	Function code	
			N	Inquire job schedule for the next job (in alphabetical order). At the end of the list, RC = 1 is returned.
			R	Inquire job schedule.
			S	Change job schedule.

Parameter	Format	Use		
P-RC	N3	out	Return code	
			0	Function OK;
			1	Owner, network, run number or job not found.
			101	Invalid function code.
			102	Parameter missing.
			103	Earliest start time invalid.
			104	Latest start time invalid.
			105	End time invalid.
			106	Latest start must be greater than earliest start.
			107	End time must be greater than latest start.
			108	Job has already been started.
			109	Earliest start must be greater than current time.
			110	Job is in HOLD status.
121	Owner does not exist.			
P-DBENV	A10	in	Database environment (for future use). OPTIONAL(*) parameter.	
P-OWNER	A10	in	Owner of the network.	
P-NETWORK	A10	in	Job network.	
P-NETWORK-VERSION	A10	in	Network version.	
P-RUN	I4	in	Run number. 0 master data are inquired or modified. else inquiry or modification is valid for one activation.	
P-JOB	A10	in	Job.	
P-AVERAGE-TIME	I4	in/out	Expected run time tenths of a second.	
P-EARLIEST-START	A14	in/out	Earliest possible start time. Format: YYYYMMDDHHISS	
P-LATEST-START	A14	in/out	Latest possible start time. Format: YYYYMMDDHHISS	
P-DEADLINE	A14	in/out	Deadline. Format: YYYYMMDDHHISS	

NOPULW9N: Write Messages to System Automation Tools Log

You can write messages to the System Automation Tools log file. The messages will be written with the product identifier of Entire Operations so that they will appear in the Entire Operations log display.

Use the following statement:

```
CALLNAT 'NOPULW9N'
      P-TASK-ID P-MESSAGE-CODE P-MESSAGE P-DBENV P-OWNER P-NETWORK P-NETWORK-VERSION
      P-RUN P-JOB P-JOB-ID P-LOG-TARGET P-USE-NAT-MSG-CACHE
```

Parameter Description

Parameter	Format	Use				
P-TASK-ID	A8	in User ID of person initiating the Entire Operations log messages. This can remain blank, in which case it is supplied with the content of the Natural *USER system variable. Entire Operations-internal use: name of the current Monitor task.				
P-MESSAGE-CODE	N4	in Internal message code (not displayed); used for later message selection. Use message codes in the range 8000-8999 to avoid conflicts with internal messages.				
P-MESSAGE	A70/1:V	in The message text line(s). Note: This parameter is a dynamic array. One or several text lines may be passed.				
P-DBENV	A10	in Database environment (for future use). OPTIONAL(*) parameter.				
P-OWNER	A10	in Owner; used for later message selection; can remain blank.				
P-NETWORK	A10	in Network; used for later message selection; can remain blank.				
P-NETWORK-VERSION	A10	in Network version; used for later message selection; can remain blank.				
P-RUN	I4	in Run number assigned to the message; can remain blank.				
P-JOB	A10	in Job; used for later message selection; can remain blank.				
P-JOB-ID	A10	in Operating system job identifier; can remain blank. Numeric values must be right-justified with leading zeros, for example: 0000004711.				
P-LOG-TARGET	A3	in Defines where to write the log message: <table border="1" data-bbox="613 1780 1380 1871"> <tr> <td></td> <td></td> </tr> <tr> <td>NOP</td> <td>Writes to Entire Operations log (default).</td> </tr> </table>			NOP	Writes to Entire Operations log (default).
NOP	Writes to Entire Operations log (default).					

Parameter	Format	Use				
		<table border="1"> <tr> <td>S0</td> <td>Writes to the SYSOUT of the current monitor task or of the batch job.</td> </tr> <tr> <td>ALL</td> <td>Writes to both the Entire Operations log and the SYSOUT.</td> </tr> </table>	S0	Writes to the SYSOUT of the current monitor task or of the batch job.	ALL	Writes to both the Entire Operations log and the SYSOUT.
S0	Writes to the SYSOUT of the current monitor task or of the batch job.					
ALL	Writes to both the Entire Operations log and the SYSOUT.					
P-USE-NAT-MSG-CACHE	L	<table border="1"> <tr> <td>in</td> <td> Use the Natural internal message cache. If this routine is called from outside the Entire Operations Monitor, this parameter must be set to FALSE, because the Natural message cache is not initialised. If this routine is called from within the Entire Operations Monitor, you may set this parameter to TRUE. In case of any problems with this setting, use FALSE. </td> </tr> </table>	in	Use the Natural internal message cache. If this routine is called from outside the Entire Operations Monitor, this parameter must be set to FALSE, because the Natural message cache is not initialised. If this routine is called from within the Entire Operations Monitor, you may set this parameter to TRUE. In case of any problems with this setting, use FALSE.		
in	Use the Natural internal message cache. If this routine is called from outside the Entire Operations Monitor, this parameter must be set to FALSE, because the Natural message cache is not initialised. If this routine is called from within the Entire Operations Monitor, you may set this parameter to TRUE. In case of any problems with this setting, use FALSE.					

The time stamp and the originating Natural user ID are inserted automatically.

This routine is also invoked internally by the other user API routines.

NOPUMI1N: Set/Reset Text Milestones in Master and Active Jobs

You can inquire on or modify milestone definitions in master and active jobs (see also [Milestone types](#) in the section *Job Maintenance*) by using the following statement in your Natural application:

```
CALLNAT 'NOPUMI1N'
      P-FUNCTION P-MILESTONE-TYPE P-DBENV P-OWNER P-NETWORK P-NETWORK-VERSION
      P-RUN P-JOB P-RC
```

The parameters have the following meaning:

Parameter	Format	Use		
P-FUNCTION	A1	in	Function code:	
			S	Set milestone.
			R	Reset (delete) milestone.
			T	Test milestone.
P-MILESTONE-TYPE	A1	in	Milestone type:	
			B	Milestone set at network begin.
			E	Milestone set at network end.
			I	Milestone set for any job (any network position except the begin and end).
			A	All milestone types (not for function code S).

Parameter	Format	Use	
P-DBENV	A10	in Database environment (for future use). OPTIONAL(*) parameter.	
P-OWNER	A10	in Owner of the network.	
P-NETWORK	A10	in Name of the job network.	
P-NETWORK-VERSION	A10	in Version of the job network. OPTIONAL(*) parameter.	
P-RUN	I4 BY VALUE RESULT	in Run number of the job network. OPTIONAL(*) parameter.	
P-JOB	A10	in Name of the job.	
P-RC	I4 BY VALUE RESULT	out Return code:	
		0	Function OK. Or: Milestone was set.
		1	Owner, network, version, run and/or job not found.
		2	Milestone was not set.
		9	Current version cannot be determined.
		101	Invalid function code.
		102	Parameter missing
		103	Invalid milestone type.
		121	Owner does not exist.
122	Network version is required.		

NOPUMT3N: Expand Message Texts

Entire Operations stores its message texts in the Log file in a language-independent, compressed format.

To read the proper message text, you must use the following routine:

CALLNAT 'NOPUMT3N'

P-APPLIC-ID P-MESSAGE P-USE_NAT_MSG_CACHE P-LANGUAGE P-PREFIX-HANDLING

**Note:** Expansion is only necessary if the message text begins with a semicolon (;).

Parameter Description

Parameter	Format	Use	
P-APPLIC-ID	A8	in Name of the application from which the message is to be taken.	
P-MESSAGE	A70	in The message text. Input: compressed. Output: readable text.	
P-USE-NAT-MSG-CACHE	L	in Use the Natural internal message cache. If this routine is called from outside the Entire Operations Monitor, this parameter must be set to FALSE, because the Natural message cache is not initialized. If this routine is called from within the Entire Operations Monitor, you may set this parameter to TRUE. In case of any problems with this setting, use FALSE.	
P-LANGUAGE	I1	in Language code for the message. Values:	
		0	Use current setting of the Natural *LANGUAGE system variable.
		1	English
		2	German
		other	English
P-PREFIX-HANDLING	A1	in A (or blank)	Do not remove message prefix.
		,	
		B	If prefix AAA9999 - , return from position 9. Example: AAAA999 - text --> - text
		C	If prefix AAA9999 - , return from position 11. Example: AAAA999 - text --> text
D	If prefix is missing in SYSERR, insert a prefix. The resulting message will be EOR1234 - Just an example. Example: The message in internal format is ;EOR;1234, and the SYSERR message text is Just an example.		

Parameter	Format	Use
		The resulting message will be EOR1234 - Just an example.

NOPUNI1N: Invalidate Entire System Server Node Table Entries

You can use this API to invalidate session node table entries.

This API must be invoked before exiting a user exit if logons to Entire System Server nodes were made. The invalidation forces new Entire System Server logons to the nodes after termination of the calling user exit.

Use the following statement:

```
CALLNAT 'NOPUNI1N'
      P-FUNCTION P-NODE(*) P-RC
```

Parameter Description

Parameter	Format	Use	
P-FUNCTION	A1	in	Function code
		I	Invalidate nodes.
P-NODE	I4/1:V	in	Node number(s) of nodes to be invalidated. The array may have an arbitrary size.
P-RC	I4	out	Return code
		0	Function OK.
		101	Invalid function code.

NOPUNX1N: Entire System Server Calls to Access UNIX and Windows Files

You can use this API to access UNIX and Windows files out of Natural programs.

CALLNAT 'NOPUNX1N' NOPUNX1A

If NOPUNX1N is called, the parameter data area of NOPUNX1A will be issued.

This section covers the following topics:

- [Parameter Description](#)
- [Runtime Environment](#)
- [Programming Hints](#)
- [Opening and Closing of Files](#)
- [Error Checking](#)
- [Reading a File](#)
- [Searching for specific Data Sets in a File](#)
- [Writing a File](#)
- [Deleting a File](#)
- [Copying a File](#)
- [Scrolling or Renaming a File](#)
- [Verifying the Existence of a File](#)
- [Creating a List of Files of a Directory](#)

Parameter Description

Parameter	Format	Use		
OBJECT	A16	in	Possible values: USER FILE	
FUNCTION	A8	in	Possible values for USER:	
			LOGON	Logon to a node.
			LOGOFF	Logoff from a node.
			Possible values for FILE:	
			COPY	Copy file.
			CLOSE	Close file.
			DELETE	Delete file.
			DIR	List files.
			MOVE	Move file or rename.
			READ	Read file.
SCAN	Search for strings in a file.			
WRITE	Write file.			
RETURN-CODE	N8	out	If 0, then function was OK. If a different value was returned, then check the data content of RETURN-TEXT.	
RETURN-TEXT	A80	out		

List of Available Entire Operations API Routines

Parameter	Format	Use		
NODE	N5	in	Node, as defined in Entire Operations.	
NODE-NAME	A16	in	Node name <ul style="list-style-type: none"> ■ can be passed as an alternative to NODE. ■ will be analyzed if NODE=0. 	
NODE-ACCESS-MODE	A1	can only be changed by the API	<i>empty</i>	has not yet been checked.
			N	Node is not local.
			Y	Node is local.
			This field is set by API at USER / LOGON. It should not be changed by the application.	
USERID	A20	in	Userid for Logon.	
GROUP	A20	in	Optional:	
			UNIX	Group for Logon.
			Windows	Domain for Logon.
PASSWORD	A16	in	Password for Logon. We recommended that you clear this field after a successful USER / LOGON.	
SID	N10	can only be changed by the API	Session ID from Entire System Server. This field is set by the API at USER / LOGON. It should not be changed by the application until USER / LOGOFF.	
UID	A10	can only be changed by the API	Internal user ID of the operating system. This field is set by the API at USER / LOGON. It should not be changed by the application until USER / LOGOFF.	
GID	A10	can only be changed by the API	Internal user ID of the operating system. This field is set by the API at USER / LOGON. It should not be changed by the application until USER / LOGOFF.	
ESC-TRIGRAPH	A1	in	Trigraph usage:	
			<i>empty</i> or N	No Trigraph converted.
			Y	Use question mark (?) as Trigraph escape character.
			other	Use this character as Trigraph escape character.
		For information on trigraph encoding , see the relevant section in <i>Submission of Jobs by Entire Operations</i> .		
TRACE-LEVEL	N1	in	0	No trace.
			> 0	Trace activated.
TRACE-TARGET	A1	in	Destination where trace will be stored.	

Parameter	Format	Use		
			S	SYSOUT
			L	Entire Operations Log
			B	Both: SYSOUT and Entire Operations Log. (only relevant if TRACE - LEVEL > 0)
OBJECT-AREA	A1200	-	Object specific fields.	
			Object: FILE (redefinition of OBJECT - AREA)	
F-FILE	A250	in	File name. The file name has to be in a qualified expression. That means in a complete path starting at the root of the file system. The file name may contain environment variables.	
F-HANDLE	N10	can only be changed by the API	File handle. Is allocated by Entire System Server at the first access of a file. Is active until FILE / CLOSE.	
F-LINE	N10	in/out	In	Line number where reading begins. (0 or 1: start reading at the beginning of the file)
			Out	(last) read line number.
F-NUMBER-RECORDS	N10	out	Number of read data sets.	
F-SEARCH-TEXT	A100	in	Only FILE / SCAN: Text to be searched for in the file.	
F-RECORD-BUFFER	A600	in/out	Whole record buffer (redefinition of F-RECORD BUFFER)	
F-RECORD-250	A250	in/out	FILE / READ, FILE / WRITE: Effective record.	

Runtime Environment

- The API can be run under Entire Operations on mainframes, UNIX or Windows.
- With this API you can access only UNIX and Windows nodes.
- In the selected Natural environment you must define the steplib SYSEOR and the steplibs used by SYSEOR.
- The system files (System File 1, System Automation Tools Log File) used by Entire Operations must be defined by LFILE assignments.

Programming Hints

Preface

In general the call OBJECT / FUNCTION (e.g. USER / LOGON), starts NOPUNX1N with the requested object and function.

Windows File Names

Windows file names can also be specified with a slash instead of a backslash.

Example:

```
c:/work/file1.txt
```

is equivalent to

```
c:\work\date1.txt
```

Session

All file operations are functional only during a valid session.

The session is started with USER / LOGON, and ends with USER / LOGOFF.

During a session the SESSION-ID, UID and GID are not to be changed.

Course of Session

```
USER / LOGON  
FILE / ...  
FILE / ...  
...  
USER / LOGOFF
```

Opening and Closing of Files

- Files must not be explicitly opened. This will be done automatically with the first call of FILE / READ or FILE / WRITE.
- After a read oder write session the file must be closed with the function FILE / CLOSE.

Error Checking

After each call of `NOPUNX1N`, the error code is to be checked in `NOPUNX1A.RETURN-CODE`. In `NOPUNX1A.RETURN-TEXT` you can access the information.

Reading a File

With the function `FILE / READ`, a file is read in sequential mode.

In `NOPUNX1A.F-LINE` you can name a line number to start.

If `NOPUNX1A.F-LINE` contains the value 0 (nil) or 1, the read sequence starts at the beginning of the file.

Scheme

```

USER / LOGON

R1. repeat
  FILE / READ
  if (NOPUNX1A.RETURN-CODE ne 0)
    escape bottom (R1.)
  end-if
  ... handle record
end-repeat      /* R1.

FILE / CLOSE

USER / LOGOFF

```

Searching for specific Data Sets in a File

With the function `FILE / SCAN`, it is possible to search for a specific string in a file. The search string must be present in `NOPUNX1A.F-SEARCH-TEXT`.

Scheme

```

USER / LOGON

R1. repeat
  FILE / SCAN
  if (NOPUNX1A.RETURN-CODE ne 0)
    escape bottom (R1.)
  end-if
  ... handle record
  add 1 to NOPUNX1A.F-LINE
*      -- line start for follow-up scan
end-repeat      /* R1.

```

```
FILE / CLOSE
```

```
USER / LOGOFF
```

Writing a File

With the function `FILE / WRITE`, a file is written in sequential mode.

Scheme

```
USER / LOGON

R1. repeat
  ... allocate record in NOPUNX1A.F-RECORD-250
FILE / WRITE
  if (...)
    escape bottom (R1.)
  end-if
end-repeat /* R1.

FILE / CLOSE

USER / LOGOFF
```

Deleting a File

With the function `FILE / DELETE`, you can delete one or several files. If `F-FILE` contains a wildcard (*), all files will be deleted that match the wildcard.

Copying a File

With the function `FILE / COPY`, the target file is passed on to `F-RECORD-250`.

Scrolling or Renaming a File

With the function `FILE / MOVE`, the target file is passed on to `F-RECORD-250`.

Verifying the Existence of a File

To verify the existence of a file, you can use the function `FILE / DIR` with a defined file name. If a 0 (nil) is returned to `NOPUNX1A.RETURN-CODE`, the files exists.

Creating a List of Files of a Directory

With the function `FILE / DIR`, you can create a list of files of a directory. With the wildcard expression contained in `NOPUNX1A.F-FILE` you can keep the list small.

Example:

<code>/tmp/*</code>	All files in the directory <code>/tmp</code> .
<code>/tmp/*.txt</code>	All files in the directory <code>/tmp</code> that end with <code>.txt</code> .

The file names are stored in `F-RECORD-BUFFER`. The file names are separated by a semicolon (;).

You can see an example of a separation in `SHOW-DIR-BUFFER` below.

If the buffer is not large enough for the file names, then the value 19 (more objects) is passed on to `NOPUNX1A.RETURN-CODE`.

Scheme

```

USER / LOGON

... allocate file name (with wildcard) in NOPUNX1A.F-FILE

R1. repeat
FILE / DIR
  decide on first value of NOPUNX1A.RETURN-CODE
    value 0, 19      /* 0: ok, 19: more objects
      perform SHOW-DIR-BUFFER
    none value
      ignore
  end-decide
  if NOPUNX1A.RETURN-CODE ne 19
    escape bottom (R1.)
  end-if
  add 1 to NOPUNX1A.F-LINE
end-repeat      /* R1.

FILE / CLOSE

USER / LOGOFF

* -----
DEFINE SUBROUTINE SHOW-DIR-BUFFER
* -----
* -- Function FILE / DIR:
* -- The record buffer contains a file list, with delimiter ';'
* -- This routine separates the single file names out of the buffer.
* -- #DIR-FILE and #DIR-REST should be declared as (A600)
*
SDB1. REPEAT

```

```

    EXAMINE  NOPUNX1A.F-RECORD-BUFFER FOR ';' REPLACE FIRST WITH ':'
SEPARATE NOPUNX1A.F-RECORD-BUFFER INTO #DIR-FILE #DIR-REST
        WITH DELIMITER ':'
*  -- now #DIR-FILE contains a single file name, without qualifiers

        IF #DIR-REST = ' '
            ESCAPE BOTTOM (SDB1.)
        END-IF
    NOPUNX1A.F-RECORD-BUFFER := #DIR-REST
END-REPEAT      /* SDB1.
END-SUBROUTINE

```

NOPURE2N: Handle Resource Allocations

You can handle resource allocations and deallocations with the following statement:

```

CALLNAT 'NOPURE2N'
    P-FUNCTION P-RC P-DBENV P-OWNER P-NETWORK P-NETWORK-VERSION P-RUN
    P-JOB P-RESOURCE P-QUANTITY P-DEALLOCATION P-JOB-EXECUTED

```

Parameter Description

Parameter	Format	Use	
P-FUNCTION	A1	in	Function code:
		A	Allocate resource.
		R	Release one or several resources. <ul style="list-style-type: none"> ■ If job and resource name given: release only one allocated resource of one job. ■ If job given, but no resource: release all allocated resources of one job. ■ If run number, but no job and no resource given: release all allocated resources of one network run. ■ If no run number and no job and no resource given: Release all allocated resources of all network runs.
		N	Network release. <p>Release resources with deallocation = N as well.</p> <p>For other parameters, see R.</p>

Parameter	Format	Use
		<p>F</p> <p>Forced release.</p> <p>Release resources with deallocation = N or K as well.</p> <p>For other parameters, see R.</p>
		<p>G</p> <p>Forced release, by resource usage.</p> <p>Release resources with deallocation = N or = K as well.</p> <p>For other parameters, see R.</p>
		<p>P</p> <p>Modify (active) prerequisite resource definition.</p>
		<p>T</p> <p>Test resource usage, sorted by active jobs.</p> <p>Subsequent calls read in the order of active jobs, then resource names.</p> <p>Sequential reading</p> <ul style="list-style-type: none"> ■ The sequential reading starts at the first found resource in the order of active jobs, then resource names. ■ Fields returned by the previous call must be kept and passed for the next call. ■ P-RC = 5 (resource usage entry not found) is returned, if there are no more entries for this resource.
		<p>U</p> <p>Test resource usage, sorted by resources, then by active jobs.</p> <p>Sequential reading</p> <ul style="list-style-type: none"> ■ The sequential reading starts at the first found resource, in alphabetical order, where the name of the first found resource is equal or subsequent to the name passed in P-RESOURCE. ■ Fields returned by the previous call must be kept and passed for the next call. ■ P-RC = 5 (resource usage entry not found) is returned at the end of all resource usage entries.

Parameter	Format	Use	
		<p>V</p> <p>Test resource usage, sorted by active jobs.</p> <p>Subsequent calls read in the order of resource names, then active jobs.</p> <p>Sequential reading</p> <ul style="list-style-type: none"> ■ The sequential reading starts at the first found resource followed by the superdescriptor of the active job (identified by P-OWNER, P-NETWORK, P-RUN, P-JOB). ■ Fields returned by the previous call must be kept and passed for the next call. ■ P-RC = 5 (resource usage entry not found) is returned at the end of all resource usage entries. 	
P-RC	N03	out	Return code:
		0	Function OK.
		1	Active job not found.
		2	Resource not defined for job.
		3	Resource (master) definition not found.
		4	Quantity is not available.
		5	OK, but no resource usage entries were found.
			This code may be returned:
			<ul style="list-style-type: none"> ■ if the resource has already been freed. ■ if the resource was never allocated.
		6	Prerequisite resource definition not found.
		101	Invalid function code.
		102	Parameter(s) missing.
		103	Wildcard not allowed.
104	P-DEALLOCATION has wrong value.		
121	Owner does not exist.		
P-DBENV	A10	in	Database environment (for future use). OPTIONAL(*) parameter.
P-OWNER	A10	in	Owner. Wildcard allowed. Function A: obligatory, no wildcard allowed.

Parameter	Format	Use
P-NETWORK	A10	in Network. Wildcard allowed. Function A: obligatory, no wildcard allowed.
P-NETWORK-VERSION	A10	in Network version.
P-RUN	I4	in Run. A value of zero means all runs of a network. Function A: obligatory, no wildcard allowed.
P-JOB	A10	in Job. If empty, the whole network is meant. Wildcard allowed. Function "A": obligatory, no wildcard allowed. Note: If this API is used to free a resource and if the allocation was made by another job, this field must remain empty.
P-RESOURCE	A20	in Name of the resource. If empty, all prerequisite resource definitions of a job are meant. Function A: obligatory, no wildcard allowed.
P-QUANTITY	N7.2	in Quantity to be allocated. Function Aonly.
P-DEALLOCATION	A1	in How to deallocate this allocation (for function A only).
		J after job termination
		N after network termination
		K keep until explicit release
For a detailed description of deallocation modes, see Resource Deallocation Modes in the section <i>Job Maintenance</i> .		
P-JOB-EXECUTED	L	in Not reusable resources are decreased only if the job was really executed.

NOPURS1N: Access Entire Operations Master Resources

This section covers the following topics:

- [Handling Entire Operations Master Resources](#)

▪ [Description of the Function Codes](#)

Handling Entire Operations Master Resources

You can handle master resources with the following statement:

```
CALLNAT 'NOPURS1N'
      P-FUNCTION P-RC P-NODE P-RESOURCE P-TYPE P-INIT-QTY P-USED-QTY
```

Meaning of the parameters:

Parameter	Format	Use		
P-FUNCTION	A1	in	Function code:	
			A	Add a resource.
			D	Delete a resource.
			M	Modify a resource.
			T	Test a resource.
P-RC	N3	out	Return code:	
			0	Function OK; resource found.
			1	Resource not found.
			2	Resource already exists.
			3	Inconsistent values.
			4	Initial quantity may not be modified.
			101	Invalid function code.
102	Resource name missing.			
P-NODE	N3	in	Not used; use constant with value=0.	
P-RESOURCE	A20	in	Name of the resource.	
P-TYPE	A1	in	Type of resource (only for FUNCTION = 'A')	
			N	Not quantitative.
			R	Quantitative, re-usable.
			U	Quantitative, not re-usable.
P-INIT-QTY	P7.2	in/out	Total quantity.	
P-USED-QTY	P7.2	in/out	Currently used quantity.	

Description of the Function Codes

Meaning of the function codes:

Code	Description
A	Adds a new resource definition. You must give values for P-TYPE, P-INIT-QTY and P-QTY.
D	Deletes a resource definition.
M	Used to modify the values P-INIT-QTY.
T	Returns the current values of P-INIT-QTY and P-QTY.



Notes:

1. Before using M for modifications, you should use T to read the present values (unless you want to enter an absolute value).
2. The currently used amount of a resource can no longer be modified directly. You must perform a resource allocation or deallocation with the API [NOPURE1N](#) to modify the currently used resource amount.

NOPUSN2N: Inquire Calling Job or Called Network for Subnetworks

You can use this API to:

- find the current subnetwork for a job of type NET;
- find the current job calling a subnetwork.

Use the following statement:

```
CALLNAT 'NOPUSN2N'
      P-FUNCTION P-RC P-DBENV *P-OWNER P-NETWORK P-NETWORK-VERSION
      P-RUN P-JOB
```

The parameters P-OWNER, P-NETWORK, P-RUN and P-JOB are input and output parameters. They are overwritten with the values found by the API. Therefore the caller must supply these fields with new entries before each new call.

Parameter Description

Parameter	Format	Use		
P-FUNCTION	A1	in	Function code	
			C	Find calling job (available for active subnetworks only).
			S	Find subnetwork.
			T	Find the topmost calling job (available for active subnetworks only). This function recursively goes back in the subnetwork hierarchy, until the caller job is no longer part of a subnetwork.
P-RC	N3	out	Return code	
			0	Function OK.
			1	Input object not found.
			2	Wrong job type.
			3	Not a calling job.
			101	Invalid function code.
			102	Parameter missing.
			121	Does not exist.
P-DBENV	A10	mod	Database environment (for future use). OPTIONAL (*) parameter.	
P-OWNER	A10	mod	Owner of the network.	
P-NETWORK	A10	mod	Network.	
P-NETWORK-VERSION	A10	mod	Network version.	
P-RUN	I4	mod	Run number. If the function S is used for a master job, this parameter must be zero.	
P-JOB	A10	mod	Job.	

NOPUSP3N: Display Long Texts for Symbol Prompting

You can display the symbol prompting long text for a specified symbol by using the following statement in your Natural application:

```
CALLNAT 'NOPUSP3N'
  P-FUNCTION P-DBENV P-OWNER P-NETWORK P-RUN P-SYMTAB P-SYMTAB-VERSION
  P-SYMBOL P-PROMPT-TYPE P-RC P-PROMPT-TEXT (*)
```

Parameter Description

Parameter	Format	Use		
P-FUNCTION	A1	in	Function code:	
			G	Get prompting attributes (for master and active).
			S	Set prompting attributes (for master and active).
P-DBENV	A10	in	Database environment (for future use). OPTIONAL(*) parameter.	
P-OWNER	A10	in	Owner of the symbol table.	
P-NETWORK	A10	in	Job network (for active symbols).	
P-RUN	I4	in	Job run number (for active symbols).	
P-SYMTAB	A10	in	Symbol table.	
P-SYMTAB-VERSION	A10	in	Symbol table version.	
P-SYMBOL	A40	in	Name of the prompted symbol.	
P-PROMPT-TYPE	A1	in/out	Input parameter for the function code S, and output parameter for the function code G. Possible values:	
			A	Prompt for a symbol at each activation.
			E	Prompt only if no value is specified in the symbol table.
			N	Never prompt for a symbol.
P-RC	N3	out	Return code:	
			0	Function ok.
			1	Symbol not found.
			99	Invalid parameter value.
			111	Symbol table name begins with reserved prefix =EOR=.
P-PROMPT-TEXT	A70/1:V	out	Symbol prompting long text. Caller should provide an array of at least 5 text lines.	

NOPUST3N: Inquire Network and Job Status, Symbol Table

You can inquire the current status of an active job network or of a single active job by using the following statement in your Natural application:

```
CALLNAT 'NOPUST3N'
  P-FUNCTION P-RC P-DBENV P-OWNER P-NETWORK P-NETWORK-VERSION P-JOB P-RUN
  P-SYMTAB P-SYMTAB-VERSION P-JOB-ID P-STATUS-TIME
```

This section covers the following topics:

- [Parameter Description](#)
- [Inquire Status of Whole Active Network](#)
- [Inquire Status of all Jobs of an Active Network Individually](#)
- [Inquire Symbol Table Used](#)
- [Example of Using NOPUST3N](#)

Parameter Description

Parameter	Format	Use	
P-FUNCTION	A1	in	Function code:
		A	Get next active run number (in numerical order) beginning with a starting run number. If the starting run number is equal to the run number limit, the range from 1 to the starting run number minus 1 is checked.
		R	Get last run number (in chronological order).
		S	Inquire status.
		N	Inquire status of next job (alphabetically).
		W	Status of next job waiting for anything, or of next job in status permanent error.
		Y	Get name of symbol table either from master network or from job definition.
P-RC	N03	out	Return code:
		0	Function OK
		1	Network/job not found, or run number does not exist
		20	Waiting for symbol prompting
		21	Waiting for job/network activation

Parameter	Format	Use
		25 Waiting for prerequisite
		26 Job(s) are in hold
		27 Waiting for next action or start time
		28 Waiting for input condition
		29 Waiting for resource
		30 Waiting for operating system node
		31 Waiting for JCL to be loaded
		32 Network activation error
		33 Job activation error
		34 Symbol replacement error
		35 Schedule extraction or network activation error
		36 Job is being submitted
		37 Job submission error
		38 Job is executing
		39 Job execution error
		41 Network/job is executing
		42 Performing End-of-Job checking
		43 End-of-Job checking error
		44 Performing End-of-Job actions
		45 End-of-Job actions error
		65 All jobs ended successfully
		66 Job(s) ended not successfully
		69 Permanent error(s)
		93 Waiting for network to be deactivated
		101 Invalid function code
		102 Parameters missing
		121 Owner does not exist
		999 Status not defined
P-DBENV	A10	in Database environment (for future use). OPTIONAL(*) parameter.
P-OWNER	A10	in Owner of network.
P-NETWORK	A10	in Job network.
P-NETWORK -VERSION	A10	in Version of job network.
P-JOB	A10	in Job. If left blank, inquiry is for whole network.
		out The job for which the return code is returned.

Parameter	Format	Use	
P-RUN	I4	in	Function code:
		A	Starting run number (can be 0).
		S	Run number to be checked.
		out	Function code:
		A	Next active run number (in numerical order).
		R	Last run number of the network in chronological order.
P-SYMTAB	A10	out	Name of the defined symbol table. Function R: Name of the defined symbol table on the network level. Function Y: Name of the defined symbol table on the network level (empty P-JOB), or the job level (given P-JOB). Empty, if a symbol table is not defined on the requested level.
P-SYMTAB-VERSION	A10	out	Version of the symbol table. The parameter only applies if P-SYMTAB contains a name.
P-JOB-ID	A10	out	ID of the job (for single jobs only, and only if the job was already submitted).
P-STATUS-TIME	A14	out	Timestamp when the current status of the job was set. Format: YYYYMMDDHHIISS.

*** OPTIONAL Parameters**

The parameter sequence is fixed.

A value must be passed from the invoking object to each parameter, except for a parameter defined with `OPTIONAL`. You can skip such a parameter by substituting it with `1X` in the `CALLNAT` statement as indicated in the [Example of Using NOPUST3N](#).

For further information on `OPTIONAL` parameters, refer to *Parameter Data Definition* in the `DEFINE DATA` section of the *Natural Statements* documentation.

Inquire Status of Whole Active Network

Use function S. Leave the P-JOB parameter blank.

Inquire Status of all Jobs of an Active Network Individually

Use function N. Delete the P-JOB parameter. Then invoke this API in a REPEAT loop until you get RC = 1 (network end).

Each call returns the status of a job. The name is contained in P-JOB. Do not change the content of P-JOB, because it is used as starting value for the next call.

Inquire Symbol Table Used

The symbol table used is always returned for active networks and jobs.

For master networks and jobs, you can use the function Y.

Notes:

- The status inquiry functions independently of Monitor activity, because it uses database entries.
- For a whole network inquiry (with function S), the statuses of the individual jobs are linked with logical AND. In the worst case, if at least one job has failed, the status `failed` is returned for the whole network.
- An active network is considered to have `terminated OK` only if all its active jobs have `terminated OK`.
- The status inquiry is only possible as long as the network or job involved has not been deactivated.
- To obtain all active run numbers of a network, proceed as follows:
 1. Use function A.
 2. Begin with starting run number 0.
 3. Call the API.
 4. Terminate when RC is not equal to 0, or when the number returned for the active run is less than the number of the previous run.
 5. Keep result as next starting run number and continue with the step Step 3 (Call the API).

Example of Using NOPUST3N

```
...
1 #RC (N3)
1 #OWNER (A10)
1 #NETWORK (A10)
1 #NETWORK-VERSION (A10)
1 #JOB (A10)
1 #RUN (I4)
1 #SYMTAB (A10)
1 #SYMTAB-VERSION (A10)
1 #JOB-ID (A10)
1 #STATUS-TIME (A14)
...
```

```
CALLNAT 'NOPUST3N'
'S' /* get status
  #RC 1X #OWNER #NETWORK #NETWORK-VERSION #JOB #RUN
  #SYMTAB #SYMTAB-VERSION #JOB-ID #STATUS-TIME
```

In the above example, 1X is the placeholder for the OPTIONAL parameter P-DBENV.

NOPUSY6N: Access Entire Operations Symbols

You can handle symbols in symbol tables using the following statement:

```
CALLNAT 'NOPUSY6N'
  FUNCTION RC DBENV OWNER NETWORK NETWORK-VERSION RUN JOB SYMTAB SYMTAB-VERSION ↵
SYMBOL FORMAT
  VALUE USER TIME
```

This section covers the following topics:

- [Description of the Parameters](#)
- [Description of the Function Codes](#)
- [Sequential Reading in a Symbol Table](#)
- [Example of Sequential Symbol Table Reading](#)

Related Topic:

- [Symbol Table and Symbol Maintenance](#)

Description of the Parameters

Parameter	Format	Use	
FUNCTION	A1	in	Function code:
		D	Delete one multiple value.
		M	Add one multiple value.
		N	Test next symbol.
		P	Test next symbol to be prompted.
		R	Reset a symbol.
		S	Set (add or modify) a symbol.
		T	Test existence of a symbol and inquire its value.
		1	Test existence of a symbol and inquire its value, with recursive resolution.
RC	N3	out	Return code:
		0	Function OK; symbol found.
		1	Symbol not found.
		2	Invalid numeric value.
		3	Invalid format.
		4	Format modification attempted.
		5	Multiple table full.
		6	Value missing.
		7	Attempted to write a predefined (read-only) symbol. Refer to <i>Predefined Symbols Ranges</i> .
		10	Invalid value by user exit.
		11	User exit not found.
		12	Access to user exit not allowed (Natural Security).
		20	OK; is a multiple value.
		30	OK; the master symbol was also modified.
		101	Invalid function code.
		102	Parameter missing.
		111	Symbol table name starts with reserved prefix =EOR=.
		121	Owner does not exist.
		131	Invalid symbol table name.
		132	Invalid symbol table version name.

List of Available Entire Operations API Routines

Parameter	Format	Use
		777 Internal parameter error. Refer to the Entire Operations log for more information.
DBENV	A10	in Database environment (for future use). OPTIONAL(*) parameter.
OWNER	A10	in Owner of the symbol table.
NETWORK	A10	in Network (for active symbol table only).
NETWORK-VERSION	A10	in Network version.
RUN	P13	in Run (for active symbol table only).
JOB	A10	in Job
SYMTAB	A10	in The symbol table.
SYMTAB-VERSION	A10	in Symbol table version.
SYMBOL	A40	in The symbol.
		out (for function codes N and P)
FORMAT	A1	in Symbol format. For the function 1 for master symbol tables, this parameter has a special meaning: FORMAT may contain the escape character to be used for the recursive resolution. The format values (below) and a question mark (?) must not be used as escape character.
		out (for function codes N, P and T)
		blank or A Alphanumeric. No case conversion.
		D Date in the format: YYYYMMDD See also <i>Date and Time Formats</i> .
		H Alphanumeric. Hidden.
		L Alphanumeric. Conversion to lower case.
		N Numeric.
		U Alphanumeric. Conversion to upper case.
VALUE	A80	in Symbol value.
		out (for function codes N, P and T)
USER	A8	out User who made the last modification.
TIME	T	out Time of the last modification.

About parameters marked “optional”:

The parameter sequence is fixed. However, optional parameters may be coded as 1X in the CALLNAT statement.

Description of the Function Codes

Meaning of the function codes:

Code	Description
D	<p>Delete one multiple value.</p> <p>Resets one value in a multiple-value symbol. If this is the last symbol value, the whole symbol is removed from the table.</p>
M	<p>Add one multiple value.</p> <p>Sets one more value in a multiple-value symbol. Even if the quantity of the multiple values is 1, the single value remains a multiple value. It will not be converted into a standard value.</p>
N	<p>Test next symbol.</p> <p>Tries to find the next symbol (in alphabetic order) from the specified symbol. The given name is overwritten by the name found. The other fields are returned as in function T.</p> <p>To find the first symbol of a symbol table, the symbol name can be omitted. If the end of the symbol table is reached, code 1 is returned.</p>
P	<p>Test next symbol to be prompted.</p> <p>Finds the next symbol in alphabetical order to be prompted for the current network/job run. Otherwise works like function code N.</p>
R	<p>Reset a symbol.</p> <p>The symbol is removed from the symbol table.</p>
S	<p>Set a symbol.</p> <p>If the symbol does not exist in the symbol table, it is inserted; if it already exists, it is overwritten.</p>
T	<p>Test a symbol.</p> <p>If the symbol is not found, code 1 is returned. If the symbol exists, its format and value are returned. The fields USER and TIME contain the user and time stamp of the last modification.</p>
1	<p>Test a symbol.</p> <p>If the symbol is not found, code 1 is returned. If the symbol exists, its format and value are returned. If the symbol value contains symbols, they will be resolved recursively. The fields USER and TIME contain the user and time stamp of the last modification.</p>



Notes:

1. To access a master symbol table, the fields `NETWORK` and `RUN` must be empty. To access an active symbol table, the fields `NETWORK` and `RUN` must be supplied by the caller.
2. If a **validation check user exit** (see the section *User Exits*) is defined for the symbol, it is invoked from this API routine, too. Make sure that the user exit is accessible in the calling environment. The library containing the user exit must be defined as a `steplib` for the executing environment.

of these APIs. A symbol value is rejected if the exit returns not OK (RC=10) or if the exit is missing (RC=11).

3. All actions will be logged.
4. The API is can also read the values of **predefined symbols** (described in *Symbol Table and Symbol Maintenance*).
5. Symbol created by this API will get the symbol prompting setting N (never). See also ***Symbol Prompting during Network or Job Activation*** in the section *Symbol Table and Symbol Maintenance*.

Sequential Reading in a Symbol Table

➤ To read symbols sequentially from a master or active symbol table, proceed as follows:

- 1 Set function to N.
- 2 If you want to start reading at the start of the table, reset SYMBOL first; otherwise put a start name into SYMBOL.
- 3 Call the user exit in a REPEAT loop; leave it if RC is not 0 and not 20.
- 4 Leave SYMBOL unchanged as start value for the next call.
- 5 Reset VALUE before the next call if RC=0.
- 6 Do not reset VALUE if RC=20.

Example of Sequential Symbol Table Reading

```
MOVE 'N' TO FUNCTION
RESET SYMBOL-NAME
R1. REPEAT
    CALLNAT 'NOPUSY6N' RC ...

    DECIDE ON FIRST VALUE OF RC
    VALUE 0, 20 IGNORE
    VALUE 1 ESCAPE BOTTOM (R1.)
    NONE VALUE
        /* Error handling
    END-DECIDE
        /* process symbol here
    IF RC NE 20
        RESET VALUE
    END-IF
END-REPEAT
```

NOPUVI2N: Obtain Entire Operations Version Information

You can use this API to find out the version of Entire Operations. Use the following statement:

```
CALLNAT 'NOPUVI2N'
      P-FUNCTION P-VERSION P-VERSION-DATE P-UPDATE-DATE
```

Parameter Description

Parameter	Format	Use	
P-FUNCTION	A1	in	Function code:
			V Version information.
P-VERSION	A20 BY VALUE RESULT	out	Version, in format <i>vv.rr.ss.pppp</i> : see Format for Product Versions .
P-VERSION-DATE	A8	out	Date of version. Format: YYYYMMDD. See also Date and Time Formats .
P-UPDATE-DATE	A8	out	Date of last updates. Format: YYYYMMDD. See also Date and Time Formats .

Format for Product Versions

A product version in the format *vv.rr.ss.pppp* denotes the following:

<i>vv</i>	Major version
<i>rr</i>	Minor version
<i>ss</i>	Service pack
<i>pppp</i>	Fix

Leading zeros are omitted, for example: 5.3.1.14.



Note: For further information on product versions, see *Version* in the *Glossary* of the Natural documentation.

NOPUXD1N: Maintain End-of-Job User Exits for a Network

You can maintain end-of-job (EJA) definitions of a specific activated Network using the following call from your Natural application.

```
CALLNAT 'NOPUXD1N' UXD-PARAMETER-AREA
```

The parameters have the following meaning:

Parameter	Format	Use
UXD-FUNCTION	A1	A Add.
		M Modify.
		D Delete.
UXD-RC	N4	Return code
		0 Function OK.
		1 Input object not found.
		2 Wrong job type.
		3 Not a calling job.
		4 Call place not supported.
		5 Invalid exit mode.
		6 Run number not in range.
		7 Exit is already defined.
		Note: For UXD-FUNCTION A only.
		101 Invalid function code.
		102 Parameter(s) missing.
121 Owner does not exist.		
UXD-DBENV	A10	Currently unused. (for future use)
UXD-OWNER	A10	Currently unused. (for future use)
UXD-NETWORK	A10	Currently unused. (for future use)
UXD-RUN	I4	Currently unused. (for future use)
UXD-JOB	A10	Currently unused. (for future use)
UXD-EVENT-TYPE	A3	JOK Job OK.
		JNO Job not OK.
		UEX No influence.
UXD-EVENT-NAME	A30	in Selected event. If empty, a new additional job-ok or job-not-ok event will be added.
		out Event name used.

XVII

User Exits

74 User Exits

▪ What User Exits Can Do	840
▪ Global User Exits	841
▪ Front-end User Exits	841
▪ Common User Exit Parameter Data Area NOPXPL-A	841
▪ Generation of Dynamic JCL and SYSOUT File Names (BS2000 only)	847
▪ User Exits for Resource Master Determination	848
▪ User Exits for Setting Input Conditions	848
▪ User Exits for End-of-Job Checking and Actions	848
▪ User Exits for Symbol Functions	850
▪ User Exits for Validation Checks of Symbol Values	853

User exits (user routines) must be coded as Natural subprograms and are automatically invoked by the Entire Operations Monitor according to where they are defined.

Entire Operation user exits require information specified in the Natural parameter data area NOPXPL-A.

When implementing Entire Operation user exits, you use the Entire Operations editor (see also [Editing Master JCL and Natural Sources](#) in the section *Job Maintenance*) and must obey the [User Exit Coding Restrictions](#).

What User Exits Can Do

User exits can be used to check job results and influence job flow by their return codes. Apart from running checks, however, the use of Entire System Server and Adabas technology allows user exits to perform various actions such as

- Database updates;
- File access and handling;
- Access to operating system functions;
- Validation checks.

User exits for the Monitor can perform any Adabas or Entire System Server function, but must not perform any type of screen I/O.

Entire Operations checks the existence of a user exit while it is being defined, as well as at invocation time. Entire Operations treats the non-existence of a user exit as an error. When such an error occurs at invocation time or during job processing, the job network execution is interrupted.

User exits can affect the Entire Operations job flow by sending a return code or other information back to the caller after execution, depending on the type and result of the routine.



Notes:

1. User exits are invoked by the Entire Operations Monitor, which performs them as subroutines. Do not forget that intensive use of user exits can influence other work to be performed by the Monitor.
2. For user exit coding, the [user exit coding restrictions](#) must be adhered to.

The following subsections describe the different types of user exit in more detail.

Global User Exits

Global user exits can be used in your whole Entire Operations environment. For more information, see the following sections in *Global User Exits* in the *Administration* documentation:

- *Global Exit for Version Names*
- *Global JCL Activation Exit*
- *Global Symbol Modification Exit*
- *Global Symbol Not Found Exit*
- *Global Message Sending Exit*

Front-end User Exits

Front-end user exits are coded as Natural subprograms and are invoked by the Entire Operations front end for specific processing.

They can use any user-defined map with the following restrictions:

- The map (Natural object, type M) must be accessible from the Entire Operations library SYSEOR. The easiest way is to copy the required maps into the Entire Operations library.
- Note that Entire Operations can be executed with the following Natural language codes: 1 = English; 2 = German. So it could be useful to supply two maps and to refer to them with the ampersand & as a placeholder representing the current language code. For more information on this technique, see the section *Multilingual User Interfaces* in the *Natural Programming Guide*.

You can define the following front-end user exit:

- Symbol prompting for a network or job activation. For more information, see [Symbol Prompting during Network or Job Activation](#) in the section [Symbol Table and Symbol Maintenance](#).

Common User Exit Parameter Data Area NOPXPL-A

All user exits under Entire Operations must use the common parameter data area NOPXPL-A, which is available in both object and source format in the SYSEOR library.

For this reason, the initial coding of a user exit must always resemble:

This section covers the following topics:

- [User Exit Types, P-CALL-PLACE](#)
- [Parameters Used for Different Call Places](#)
- [Additional Independent \(AIV\) Variables](#)
- [Predefined Symbols](#)
- [Common Considerations for User Exits](#)
- [Decimal Character \(DC\) Setting](#)
- [User Exit Coding Restrictions](#)

User Exit Types, P-CALL-PLACE

The field [P-CALL-PLACE](#) in the parameter list shows for which purpose the user exit was called:

P-CALL-PLACE	Description
EJA	End-of-Job Action See also Defining End-of-Job Action User Exit in the section <i>End-of-Job Checking and Actions</i> .
EJC	End-of-Job Checking See also Editing End-of-Job Checking or Action User Exits in the section <i>End-of-Job Checking and Actions</i> .
FSB	SYSOUT File Name Generation (BS2000) See also NOPFB2-N: Generate SYSOUT File Names for BS2000 in the section <i>API Routines</i> .
ICO	Input Condition Value Determination See also Input Condition: User Exit in the section <i>Job Maintenance</i> .
MAC	Dynamic JCL Generation See also Dynamic JCL Generation (JCL Location MAC) in the section <i>System Overview</i> .
NAT	Natural Program Standalone under Entire Operations (NAT-type Job) See also Job Types in the section <i>System Overview</i> .
NVN	Network version name, syntax check See also Global Exit for Version Names in the <i>Administration</i> documentation.
RMD	Resource Master Determination Exit See also Resource Master Determination Exit in the <i>Administration</i> documentation.
SFX	Symbol Function Exit See User Exits for Symbol Functions .

P-CALL-PLACE	Description
SNF	Global Symbol not Found Exit See <i>Global Symbol not Found Exit</i> in the <i>Administration</i> documentation.
SVN	Symbol table version name, syntax check See <i>Global Exit for Version Names</i> in the <i>Administration</i> documentation.
SYC	Symbol Validation Check See <i>User Exit for Validating Symbol Values</i> .
SYF	Symbol Function See <i>User Exits for Symbol Functions</i> .

Parameters Used for Different Call Places

Field Name	Format	Call Place													
P-CALL-PLACE	A3	EJA	EJC	FSB	ICO	MAC	NAT	NVN	RMD	SFX	SNF	SVN	SYC	SYF	
P-RC (Return code)	N4	out	out	out	out	n/a	out	out	out	out	out	out	out	out	
P-RT (Return text)	A66	out	out	out	out	n/a	out	out	out	out	out	out	out	out	
P-OWNER	A10	in	in	in	in	in	in	in	in	in	in	in	in	in	
P-NETWORK	A10	in	in	in	in	in	in	in	in	in	in	in	⁽³⁾	in	
P-NETWORK-VERSION-1	A10	n/a	n/a	n/a	n/a	n/a	n/a	in	n/a	n/a	n/a	n/a	n/a	n/a	
P-JOB	A10	in	in	in	in	in	in	n/a	in	in	in	n/a	n/a	in	
P-RUN	P13	in	in	in	in	in	in	n/a	in	in	in	n/a	⁽³⁾	in	
P-ACTIVATION-TIME	T	in	in	in	in	in	in	n/a	in ⁽⁷⁾	n/a	n/a	n/a	⁽³⁾	n/a	
P-EXECUTION-NODE ⁽¹⁰⁾	N3	in	in	in	in	in	in	n/a	in	in	in	n/a	n/a	in	
P-EXECUTION-OPSYS	A8	in	in	in	in	in	in	n/a	in	n/a	n/a	n/a	n/a	n/a	
P-SYMTAB	A10	in	in	in	in	in	in	n/a	in	in	in	n/a	n/a	in	
P-SYMTAB-VERSION-1	A10	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	in	n/a	n/a	
P-CONDITION	A20	⁽⁶⁾	⁽⁶⁾	n/a	in	⁽²⁾	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
P-RESOURCE	A20	in	n/a	n/a	n/a	n/a	n/a	n/a	in	n/a	n/a	n/a	n/a	n/a	
P-SYMBOL ⁽¹³⁾	A20	⁽¹⁾	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	mod	n/a	n/a	in	
P-JOB-ID	A10	out	out	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
P-JOB-NUMBER ⁽⁹⁾ (obsolete)	N5	out	out	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
P-SYSOUT-FILE ⁽¹¹⁾	A54	⁽¹⁾	⁽¹⁾	out	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
P-FSB-OBJECT-TYPE	A5	n/a	n/a	in	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	

Field Name	Format	Call Place													
P-FSB-USERID	A8	n/a	n/a	in	n/a										
P-FSB-CATID	A4	n/a	n/a	in	n/a										
P-FSB-SUFFIX	A2	n/a	n/a	in	n/a										
P-SYMBOL-VALUE ⁽¹⁴⁾	A54	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	out	n/a	n/a	in	
R-RMD-OLD-QTY-INITIAL	N7.2	n/a	n/a	n/a	n/a	n/a	n/a	n/a	in	n/a	n/a	n/a	n/a	n/a	
P-RMD-NEW-QTY-INITIAL	N7.2	n/a	n/a	n/a	n/a	n/a	n/a	n/a	out	n/a	n/a	n/a	n/a	n/a	
P-RMD-PARAMETER	A36	n/a	n/a	n/a	n/a	n/a	n/a	n/a	in	n/a	n/a	n/a	n/a	n/a	

where:

n/a = Parameter can contain invalid values, do not modify.

in = Parameter contains input values.

out = Parameter contains output values.

mod = Parameter modifications are saved.

Notes:

- ¹ in; BS2000 only
- ² in; contains P-SUFFIX in first 10 bytes
- ³ in; only for active symbols
- ⁶ in; contains P-JOB-ID (A5) in first 5 bytes
- ⁷ in; here: timestamp of previous exit call
- ⁹ This field is obsolete. Use P-JOB-ID-10 instead. The size of P-JOB-NUMBER is too small for some operating systems. For compatibility reasons, the field will be filled on return, if the job number is in the range of 1 to 99999.
- ¹⁰ Node numbers from 1 thru 999 are provided here. For node numbers greater than 999, P-EXECUTION-NODE is zero. Node numbers of any size are provided in the AIV variable +P-EXEC-NODE-I4 (I4).
- ¹¹ For longer file names, you must use the AIV variable +P-SYSOUT-FILE-250 (A250).
- ¹³ This field is only maintained for compatibility reasons. Parameter modifications using the internal A40 format/length can truncate values thus causing data loss.
- ¹⁴ This field is only maintained for compatibility reasons. Parameter modifications using the internal A80 format/length can truncate values thus causing data loss.

Additional Independent (AIV) Variables

Additionally to the parameter list NOPXPL-A, some parameters are provided as independent (AIV) variables.

Field Name	Format	Call Place												
		EJA	EJC	FSB	ICO	MAC	NAT	NVN	RMD	SFX	SNF	SVN	SYC	SYF
+P-NETWORK-VERSION	A10	in	in	in	in	in	in	in	in	in	in	in	n/a	in
+P-SYMTAB-VERSION	A10	in	in	in	in	in	in	n/a	in	in	in	n/a	n/a	in
+P-EXEC-NODE-I4	I4	in	in	in	in	in	in	n/a	in	in	in	n/a	in	in
+P-SYSOUT-FILE-250	A250	in	in	n/a										
+P-JI-JOB-TYPE	A3	in	in	n/a										
+P-JI-DUMMY-FLAG	A1	in	in	n/a										

Field Descriptions

+P-NETWORK-VERSION	Version of the network.	
+P-SYMTAB-VERSION	Version of the symbol table.	
+P-EXEC-NODE-I4	This field provides node numbers of any size (1 thru 99900). It is provided for each node number. You must use it instead of P-EXECUTION-NODE, if the node number is greater than 999.	
+P-SYSOUT-FILE-250	This field provides the SYSOUT file name in a field with a length of 250 bytes. You must use this field for the name of the SYSOUT file for a UNIX or Windows job (instead of P-SYSOUT-FILE) to avoid truncation.	
+P-JI-JOB-TYPE	The job type (A3). For valid values, see the three-letter codes in <i>Available Job Types</i> in the section <i>Job Maintenance</i> .	
+P-JI-DUMMY-FLAG	Reason why a job became a temporary dummy job (dummy due to ...).	
	J	JCL check.
	D	Definition.
	S	Schedule.
	R	Recovery.
	T	Repetition.
	E	Empty JCL.
	C	Condition.
	M	Multiple suffix.
K	To be deactivated.	

Predefined Symbols

For a list of predefined symbols, see [Predefined Symbols](#) in the section [Symbol Table and Symbol Maintenance](#).

Common Considerations for User Exits

All user exits must return the field P-RC (return code). 0 (zero) means OK.

The field P-RT (return text) can be used optionally to pass back some specific information. Entire Operations writes this text into the log and displays it on the **Active Jobs** screen, if supplied.

All other parameters pass environment information to the user exit. Some [parameters](#) and [AIV variables](#) are not always required; see the relevant field descriptions for details.

Decimal Character (DC) Setting

The parameter data area NOPXPL-A contains numbers with decimal characters. The default setting for decimal character in the delivered source of NOPXPL-A is a period (.). If you want to catalog your user exits and Natural programs with another decimal character (e.g. comma, DC=,), you must also modify NOPXPL-A:

- Make sure that the decimal character of your Natural session is set correctly. You can verify this with the system command GLOBALS (described in the *Natural System Commands* documentation).
- Use the Natural Editor to edit NOPXPL-A.
- Change the decimal character to your current DC parameter setting in the numeric fields, which contain a decimal character. Stow NOPXPL-A.

You will now be able to use NOPXPL-A with your own decimal character.

User Exit Coding Restrictions

1. User exits must reside in a Natural library on the FUSER system file, but may not reside in the Entire Operations SYSEOR system library.

Sub-objects invoked by user exits (subprograms, etc.) and global user exits (see the *Administration* documentation) and must reside in the SYSEORU system library.

2. ON ERROR routines are allowed in Entire Operations user exits, with the following restriction: The ON ERROR block must not be left by any ESCAPE ROUTINE, ESCAPE MODULE, TERMINATE or similar statement. The purpose of this restriction is that the common Entire Operations Monitor (or Entire Operations online) error transaction gets control after the execution of the user-defined ON ERROR block. The common Entire Operations error transaction performs some common error handling, logging, notifications etc.
3. No programs with their own GDA (global data area) may be invoked.

4. The names of user exits must not begin with the letter V, particularly if they are to be stored in the library SYSEOR, because the prefix V is reserved for the internal Entire Operations exit directory.
5. The following Natural statements must *not* be coded in any user exit invoked by Entire Operations:
 - END TRANSACTION

The transaction logic is handled by the Entire Operations Monitor itself.
 - BACKOUT TRANSACTION
 - STOP
 - TERMINATE
 - FETCH **without** RETURN (FETCH RETURN is allowed)
6. Entire System Server node logons and logoffs should be avoided. This applies for mainframe nodes (view NATPROC-LOGON), UNIX nodes, and Windows nodes. If any logons and/or logoffs to Entire System Server nodes were made in a user exit, the exit must invoke the Entire System Server node invalidation [API NOPUNI1N](#) prior to termination. This forces subsequent new logons to these nodes.
7. Do not code any screen and map input/output in user exits. An exception is the symbol prompting user exit in online usage mode. See also [Specifying User Exits for Symbol Modification](#) in the section [Symbol Table and Symbol Maintenance](#).
8. In general, the execution of all kinds of code, which can disturb the Entire Operations Monitor execution, is forbidden in user exits (exits). If in doubt, contact Software AG technical support.



Notes:

1. If you want to use the values of the special parameters in Macro JCL, you must first read them with the special statement [#GET-SYMBOL](#) (described in the section [System Overview](#)).
2. See also [User Exits for Validation Checks of Symbol Values](#). For foreground usage, this exit may use maps (with some restrictions).

Generation of Dynamic JCL and SYSOUT File Names (BS2000 only)

For MAC-type jobs, you can generate dynamic JCL: see [Dynamic JCL Generation \(JCL Location MAC\)](#) in the section [System Overview](#).

In a BS2000 environment, you can generate file names for SYSOUT: see [NOPFB2-N: Generate SYSOUT File Names for BS2000](#) in the section [API Routines](#).

User Exits for Resource Master Determination

For each master resource, you can define a resource amount determination exit: see *Resource Master Determination Exit* in the *Administration* documentation.

User Exits for Setting Input Conditions

Before Entire Operations can submit a job, all its input conditions must be fulfilled.

If a user exit is defined for a condition, Entire Operations invokes the defined exit (of type ICO) each time it checks the condition status. If the exit returns zero in the P-RC parameter, the condition will be satisfied.

Otherwise, the condition checking will be continued.

You can set a return text in the P-RT parameter additionally.

User Exits for End-of-Job Checking and Actions

You can define user exits as part of End-of-Job checking. Entire Operations bases its End-of-Job checking on the occurrence of defined events during job execution. You can define a user exit as an event check. Entire Operations automatically executes the user exit when the job terminates.

Typically, such a user exit routine can check the result of a job, for example. The user exit returns a code which determines the event status. A return code zero means *Check OK*, any other return code means *Check not OK*. Condition codes returned by user exits can determine End-of-Job status.

End-of-Job action user exits can be used to perform any actions after job termination.

End-of-Job checking and action user exits are supplied with both the job number and the name of the SYSOUT file in which Entire Operations collected the job SYSOUT (BS2000 only).

For End-of-Job user exits, the common exit parameter list **NOPXPL-A** must be used. The parameter P-CALL-PLACE contains EJC for End-of-Job checking user exits or EJA for End-of-Job action user exits.

Related Topic:

- [End-of-Job Checking and Actions](#)

This section covers the following topics:

- Example

Example

An End-of-Job checking user exit is shown in the example below.

This user exit routine adds up the contents of a numeric field in the SYSOUT and compares it with a given value. If the sum of the contents equals the sum of the given value, the event is treated as ok, otherwise as not ok.

```

DEFINE DATA PARAMETER USING NOPXPL-A
LOCAL
1 READ-SPOOL VIEW OF READ-SPOOL      /* AN ENTIRE SYSTEM SERVER VIEW
  2 RECORD
  2 REDEFINE RECORD
    3 RECORD-HEADER    (A05)
*
1 #NF          (I02) CONST <10>    /* MAX. NUMBER FOR SEPARATION
1 #SUM         (N08) INIT  <0>
1 #SUM-EXPECTED (N08) CONST <2000>
1 #FIELD       (A10/1:#NF)
1 #USERID      (A08)
1 #DBENV       (A10)
1 #NETWORK-VERSION (A10)

1 #LOG-STATUS  (N04)
1 #LOG-MESSAGE (A60)
1 #LOG-TARGET  (A03) CONST <'NOP'>
1 #USE-NAT-MSG-CACHE (L) CONST <FALSE>
END-DEFINE

```

```

* -----
* This EOJ User Exit will read the Job SYSOUT (data set 2)
* and will summarize a defined field.
* If the sum is equal to a given value, this routine returns
* an 'OK' (P-RC = 0), otherwise 'NOT OK'.
*
* A log record will be written to document the result.
*
RESET #SUM
RS. FIND READ-SPOOL
  WITH NODE = P-EXECUTION-NODE AND JOB-NUMBER = P-JOB-ID
  AND TYPE = 'SO' AND DATA-SET = 2 AND RECORD-NUMBER >= 1
  ACCEPT IF RS.RECORD-HEADER = 'REC01'      /* interesting only
  SEPARATE RS.RECORD LEFT JUSTIFIED INTO #FIELD(*)
  ADD VAL(#FIELD(3)) TO #SUM
END-FIND
*

```

```

IF #SUM = #SUM-EXPECTED
  RESET P-RC
  MOVE 'Sum check is ok' TO #LOG-MESSAGE
ELSE
  MOVE 1 TO P-RC
  COMPRESS 'Sum:' #SUM 'Expected:' #SUM-EXPECTED INTO #LOG-MESSAGE
END-IF
#USERID := *USER
MOVE 9999 TO #LOG-STATUS
CALLNAT 'NOPULW93N'          /* WRITE LOG RECORD
  #LOG-STATUS #LOG-MESSAGE #DBENV
  P-OWNER P-NETWORK #NETWORK-VERSION P-RUN P-JOB
  P-JOB-ID #LOG-TARGET # USE-NAT-MSG-CACHE
END

```

User Exits for Symbol Functions

You can perform user-specific symbol functions via a user exit. For the symbol functions available, see [Functions for Symbol Replacement](#) in the section [Symbol Table and Symbol Maintenance](#).

- [Symbol Function Syntax](#)
- [Symbol Function Coding Rules](#)
- [Symbol Function Parameter List](#)
- [Example of a Symbol Function](#)

Symbol Function Syntax

Syntax for symbol functions:

```

$!|?function[parm,...]

```

Symbol Function Coding Rules

- As for other user exits (exits), the [user exit coding restrictions](#) must be adhered to.
- For a function to be recognized, an exclamation mark (!) or a question mark (?) must always follow the escape character. In BS2000 the exclamation mark (!) is represented by an ö character.
- The parameter list is optional and is enclosed in square brackets ([]). No spaces are allowed between function name and parameter list.
- Parameters have to be separated by commas.
- The total length of the function call may not be longer than 20.
- The symbol functions have to be located in the SYSEORU library.
- The Entire Operations Monitor performs logging of symbol replacements. Additional logging may also be performed.

- After successful replacement, the symbol value is written to the active symbol table of the job. The symbol function is not called for subsequent replacements. Instead, the value is read from the active table.

The names of the symbol function exits begin with `SX`. Since the length of the names of Natural objects is limited to 8 characters, the function name (the exclamation mark (!) not counted) can have a maximum length of 6 characters.

Symbol Function Parameter List

The function `!RANDOM` is handled by the Natural subprogram `SXRANDOM`.

The symbol function exits are called with the parameter list `NOPXPL-A`, so the first line of the exit should be:

```
DEFINE DATA PARAMETER USING NOPXPL-A
```

Meaning of the parameters in `NOPXPL-A`:

Parameter
P-CALL-PLACE
P-RC
P-RT
P-OWNER
P-NETWORK
P-JOB
P-RUN
P-EXECUTION-NODE
P-SYMBOL-TABLE
P-SYMBOL
P-SYMBOL-VALUE

We strongly recommend that you only use the fields listed in the previous table. Using any other field from `NOPXPL-A`, can cause unpredictable results.

Notes:

1. Caller supplied `SFX` (symbol function exit) for `P-CALL-PLACE`.
2. Set `P-RC` to either of the following return codes:
 - 0 (Function ok.), or
 - 4531 (Symbol not found.)

3. P-SYMBOL only transfers the function parameters (without brackets). The function names can be identified from the Natural variable *PROGRAM (described in the *Natural System Variables* documentation).

Example of a Symbol Function

User Exit Subprogram SXQS

```
* SYMBOL FUNCTION EXIT EXAMPLE
* SXQS<PARM1>
* CROSSFOOTING OF THE PARAMETER
* -----
DEFINE DATA PARAMETER USING NOPXPL-A
LOCAL
1 #I   (I02)
1 #N   (I02)
1 #SUM (I02)
1 #A   (A01)
END-DEFINE
* -----
RESET #SUM
F1. FOR #I = 1 TO 20
  #A := SUBSTR (P-SYMBOL,#I,1)
  IF #A IS (N1)
    #N := VAL(#A)
    ADD #N TO #SUM
  END-IF
END-FOR    /* F1.
P-SYMBOL-VALUE := #SUM
RESET P-RC P-RT
END
```

Master JCL

```
//SN000001 JOB ,SN,CLASS=K
//IEFBR14 EXEC PGM=IEFBR14
/*
/* +!D<AQ+1>
/* $!D<AQ+1>
/*
/* +!QS<4711>
/* $!QS<4711>
/*
/* +!QS<+*DATE>           NESTED SYMBOLS
/* +!QS<$*DATE>           NESTED SYMBOLS
/* $!QS<$*DATE>           NESTED SYMBOLS
/*
/* +!QS<+!D<AM-1>>       NESTED FUNCTIONS
```

```

/** +!QS<$!D<AM-1>>          NESTED FUNCTIONS
/** $!QS<$!D<AM-1>>          NESTED FUNCTIONS
/**

```

Active JCL

```

//SN000001 JOB ,SN,CLASS=K
//IEFBR14 EXEC PGM=IEFBR14
/**
/** +!D<AQ+1>
/** 20080101
/**
/** +!QS<4711>
/** 13
/**
/** +!QS<+*DATE>            NESTED SYMBOLS
/** +!QS<29/10/08>          NESTED SYMBOLS
/** 18                       NESTED SYMBOLS
/**
/** +!QS<+!D<AM-1>>          NESTED FUNCTIONS
/** +!QS<20080131>          NESTED FUNCTIONS
/** 30                       NESTED FUNCTIONS
/**

```

Active Symbol Table

Symbol	F A Value	modified by
!D<AM-1>	A E 20080131	EORMON 11.01.08 17:25
!D<AQ+1>	A E 20080101	EORMON 11.01.08 17:25
!QS<11/01/08>	A E 20	EORMON 11.01.08 17:25
!QS<20080131>	A E 15	EORMON 11.01.08 17:25
!QS<4711>	A E 13	EORMON 11.01.08 17:25
X	A E x	SN 10.11.08 13:13

User Exits for Validation Checks of Symbol Values

You can write a user exit routine ([front-end user exit](#)) that defines your own prompting screens and validation (plausibility) checks for symbol prompting during a job network activation. For more information on symbol prompting, see [Symbol Prompting during Network or Job Activation](#) in the section [Symbol Table and Symbol Maintenance](#).

The user exit is called by the Entire Operations Monitor after the creation of the active symbol tables, but before the JCL loading. If the user exit is missing or receives a run time error, the network activation will be stopped and a message will be sent to the mailbox, which is assigned to the

network as message recipient. If no mailbox is assigned, the message is sent to the SYSDBA mailbox. For details, see the section [Mailboxes](#).

When writing the user exit routine, use the mandatory parameter data area NOPXPL-A and the parameter list [NOPSYP3A](#).

NOPXPL-A contains the owner, symbol table, symbol name, etc. It can therefore be used for different symbols.

[NOPSYP3A](#) contains all needed environment parameters. The list of symbol tables used for this network activation will be passed to the Entire Operations API routine.

The symbols must be read and updated by the Entire Operations API routine NOPUSY_xN which allows sequential reading in the active symbol table. *x* denotes a sequential version number of 1 through 6, where 6 is the most recent version of the API.

The parameters P-RC (return code) and P-RT (return text) are examined by the caller after execution of the user exit. If P-RC is zero, the symbol is accepted as OK. Otherwise, the symbol is not accepted.

If the text from P-RT is not blank, it will be displayed to the user with message code EOR1855. If P-RT is not zero and blank, a standard error message is displayed.

As in other user exits, Adabas, Entire System Server and Natural system variables can provide a lot of flexibility to symbol checking.

This section covers the following topic:

- [Parameter List NOPSYP3A](#)
- [Symbol Modification without Prompting](#)

Parameter List NOPSYP3A

```

*   NOPSYP3A
*   Entire Operations
*   Exit parameter list
*   for symbol prompting
*
*   27.09.12 N1399    SN541          /* successor of NOPSYP2A
*   11.03.13 N1574    SN541          /* N1574
*   20.08.13 N1649    SN541          /* P-USED-SYMTAB-34
*   -----
*   1 P-CALL-PLACE          A          3
*   'SYP'  symbol prompting online
*   'SYM'  symbol modif.   batch
*   1 P-RC                  N          4 /* in/out  return code
*   0  out: ok
*   1  out: no symbols prompted
*   2  out: activation cancelled

```

```

*      4 out: end (no modification)
*
*      3 in:  rewrite modified
*            symbols to symbol
*            master too
1 P-RT                A          70 /* out  return text
*
1 P-OWNER             A          10 /* in
1 P-NETWORK           A          10 /* in
1 P-NETWORK-VERSION  A          10 /* in  312641 add
1 P-RUN               I           4 /* in  312641 fmt, order
1 P-JOB               A          10 /* in  312641 order
1 P-ACTIVATION-TIME  T           /* in
1 P-EARLIEST-START   T           /* in
1 P-NETWORK-SYMTAB   A          10 /* in
R 1 P-NETWORK-SYMTAB          /* REDEF. BEGIN : P-NETWORK-SYMT
2 P-NETWORK-SYMBOL-TABLE      A          10 /* 312641.3
1 P-NETWORK-SYMTAB-VERSION    A          10 /* 312641 add
1 P-USED-SYMTAB-34           A          34 (1:V) /* N1649
*      -- Use LDA NOPSYT1L for the          N1649
*      -- layout / redefinition of          N1649
*      -- the table elements.              N1649
1 P-DATE-FORMAT         A          1 /* in
*      'A' American
*      'E' European
*      'G' German
*      'I' International
*
*      -- end of parameter data area --

```

This section covers the following topics:

- [Supplied by the Caller on Input](#)
- [Return Codes to Be Set by the Symbol Modification User Exit](#)

Supplied by the Caller on Input

P-RC	Meaning
3	Rewrite modified symbols to master symbol table.

Return Codes to Be Set by the Symbol Modification User Exit

P-RC	Meaning
0	OK, modifications were done.
1	OK, no symbols prompted or modified.
2	Activation cancelled.
4	End; no modification.

Symbol Modification without Prompting

Code the user exit analogous to the symbol prompt user exit and use the provided [parameter list NOPSYP3A](#), in conjunction with the local data area [NOPSYT1L](#). Do not code any screen I/Os if you do not use foreground prompting.

This section covers the following topics:

- [Data Area NOPSYT1L](#)

Data Area NOPSYT1L

The layout of the field P-USED-SYMTAB-34 is specified in the data area NOPSYT1L.

```

*      NOPSYT1L
*      'Symbol tables used'
*      table entry
*
*      Maintenance
*      13.08.13  N1649      SN541                created
*      07.10.13  N1649.A   SN541                'origin object type' val.
*      -----
*      -- values for SYTU-SYMTAB-STATUS
*
*      1 SYTU-SYMTAB-STATUS-EV-DUP      A      2 CONST<'ED'>
*      -- effective symbol table
*      -- version is a duplicate --
*      -- entry should be skipped
*      -- during prompting
*      1 SYTU-SYMTAB-STATUS-UNDEF      A      2 CONST<'00'>
*      -- undefined
*      1 SYTU-SYMTAB-STATUS-NPN        A      2 CONST<'10'>
*      -- no prompting necessary
*      1 SYTU-SYMTAB-STATUS-PN         A      2 CONST<'20'>
*      -- prompting necessary
*      1 SYTU-SYMTAB-STATUS-PD         A      2 CONST<'30'>
*      -- prompting done
*
*      -----
*

```

```

*   -- values for
*   -- SYTU-ORIGIN-OBJECT-TYPE
*
1 SYTU-ORIGIN-OBJECT-NV          A          2 CONST<'NV'>
*   -- network version definition
1 SYTU-ORIGIN-OBJECT-JM          A          2 CONST<'JM'>
*   -- job (master) definition
1 SYTU-ORIGIN-OBJECT-IM          A          2 CONST<'IM'>
*   -- input cond. (mult. suffix)
1 SYTU-ORIGIN-OBJECT-IS          A          2 CONST<'IS'>
*   -- input cond. (dep. on symbol)
1 SYTU-ORIGIN-OBJECT-AS          A          2 CONST<'AS'>
*   -- EOJ action (symbol setting)
*
*   -----
*
1 SYTU-ENTRY-34                  A          34
R 1 SYTU-ENTRY-34                  /* REDEF. BEGIN : SYTU-ENTRY-34
2 SYTU-SYMTAB-STATUS             A          2
*   -- values see above
2 SYTU-ORIGIN-OBJECT-TYPE        A          2
*   -- values see above          /* N1649.A
*
*   ...
2 SYTU-SYMTAB                    A          10
2 SYTU-SYMTAB-VERSION-DEFINED    A          10
2 SYTU-SYMTAB-VERSION-EFFECTIVE  A          10
*   blank:
*   not determined yet
*   KVE-TEXT-UNNAMED = '(none)'
*   determined and blank
*   other:
*   determined and non-blank
*
*   -- end of area --

```


XVIII

Passing Files to Entire Output Management

75

Passing Files to Entire Output Management

- Basic Requirements for File Transfers to Entire Output Management 862
- Listing Files Defined for Entire Output Management 863
- Adding and Modifying File Definitions 865
- Deleting File Definitions 870
- Handing Over SYSOUT and Files to Entire Output Management 870
- Defining the Handover of SYSOUT Files from UNIX/Windows to BS2000 874

You can define and pass job SYSOUT and/or one or more other files (for example, output files of a job) to Entire Output Management (NOM) to further process the output in this environment.

Basic Requirements for File Transfers to Entire Output Management

This section describes the requirements for transferring files from Entire Operations to Entire Output Management.

1. An `NTLFILE` or `LFILE` definition for the Entire Output Management system file must be entered in the start parameters for System Automation Tools or Entire Operations. For further information, see the section *Installing Entire Operations on Mainframes* in the *Installation and Setup* documentation.
2. If Entire Output Management (`LFILE 206`) is defined, and if the Entire Operations monitor receives a NAT0082 message calling the Entire Output Management API, the message `NOM Trigger interface not active` will be written to log before Entire Operations task restart. Subsequent attempts to pass SYSOUT to Entire Output Management will be ignored with the message `Passing to NOM impossible`.
3. If SYSOUT or another file cannot be passed to the Entire Output Management API, the Entire Output Management API calls for a job to be **retried**. Check the Entire Operations log for more detailed error information.
4. In Entire Output Management, the report name is EOR-SOUT.
5. For UNIX and Windows, refer to [Passing SYSOUT from UNIX and Windows Nodes to Entire Output Management](#).

See also:

- [Handing over SYSOUT and Files to Entire Output Management](#).
- The Entire Operations default setting **NOM API retry limit** described in *Default Setting (1)* in the *Administration* documentation.
- The Entire Operations default setting **Copy SYSOUT File before passing it to NOM** described in *Default Setting (2)* in the *Administration* documentation.

Listing Files Defined for Entire Output Management

> To list files defined for processing with Entire Output Management

- 1 Select a job and open the **End-of-Job Checking + Actions** screen.

The letter **T** in the **Action** column indicates that an EOJ action for Entire Output Management exists:

```

17.04.16          ***** Entire Operations *****          18:02:27
Owner REQUEST    Network P106270    Version                Job J-1
End-of-Job Checking + Actions  MVS/ESA                Run        Date
-----
C Action          Step          will be checked for          means OA
-                T                Additional Actions for Job-ok          ok

***** Bottom of Data *****
-                T X    All Checks ok
-                T    Any Check not ok
-----
A Activat. C Cond. D Delete E Edit Exit J JV L Re1 Res M Modify O Other
P Descr. R Recov. S SYSOUT T Output Mgmt U User Msg X Action Exit Y Symbol Set

Command => _____

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Add   End       Save       Up       Down              Menu

```

See also *End-of-Job Checking and Actions Screen* in the section *End-of-Job Checking and Actions*.

- 2 On the **End-of-Job Checking + Actions** screen, type T (Output Mgmt) in the **All checks ok** or **Any check not ok** line command field, and press ENTER.

If you select **All checks ok**, a **Files for Entire Output Management** window similar to the example below opens:

```

17.04.16          ***** Entire Operations *****          17:51:35
Owner REQUEST    Network P106270    Version                Job J-1
End-of-Job Checking + Actions  MVS/ESA                Run        Date
-----
C Acti +-----+
-      |          Job ended ok
      |          Files for Entire Output Management
      |          Cmd File
      |          -   SYSOUT A 1
      |          -   HUGO.HUGO.HUGO
      |          -   SYSOUT S0 3
      |          -   Procname PROC Stepname STEP DDName DD
*****      |          -   @EUMEL..TEST
t         |          -   SYSOUT S0 4
-         |          -   Procname XXX Stepname YYY DDName ZZZ
-----      |          -   TEST.TEST.TEST
A Acti |
P Desc |
Comman |          D Delete  M Modify
      |          Enter---PF1---PF2---PF3---PF5-----
      |          Help  Add  End  Save
      |-----+
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Add  End          Save          Up    Down          Menu
    
```

This window contains a list of all files to be handled by Entire Output Management.

Line Commands

Command	Description
D	Delete a file definition.
M	Modify a file definition. Depending on the definition, a Spool File or an Output File definition window opens.

Special PF Key

PF Key	Description
PF2	Add a file definition for Entire Output Management. For mainframe execution nodes, a selection window opens. For UNIX and Windows nodes, an Output File definition window opens directly.

Adding and Modifying File Definitions

This section provides instructions for adding and modifying file definitions for processing in Entire Output Management.

 **Note:** The number of files you can define per job event is limited to 10. If you want to specify more than 10 files for a job, define one or more additional events of the type A (**Add. Job-ok,-not-ok**), and define the extra files for these events.

> To add a file definition

- 1 In the **Files for Entire Output Management** window, press PF2 (Add).

A window similar to the example below opens:

```

+-----+
| Additional Actions for Job-ok                               |
| File Definition for Output Management (NOM)                |
|                                                             |
| Please select:                                           |
|   Spool File Definition  ==> _                          |
|   Output File Definition ==> _                          |
|                                                             |
| -----PF3-----                                         |
| End                                                         |
+-----+

```

- 2 Select **Spool File Definition** or **Output File Definition** by entering any character in the appropriate field, and press ENTER.

Depending on the option selected, one of the following windows opens:

For a spool file definition:

```

Additional Actions for Job-ok
  File Definition for Output Management (NOM):
    Spool File

  Operating System ==> MVS/ESA
  Spool Type       ==> JES2

Please specify either
  Spool File Type  ==> __
  Spool File Number ==> _____

or
  Procname        ==> _____
  Stepname        ==> _____
  DD Name         ==> _____

---PF1---PF3-----
  Help  End
  
```

For an output file definition:

```

Job ended ok
  File Definition for Output Management (NOM):
    Output File (sequential file)

  Operating System ==> MVS/ESA   Spool Type ==> JES2

  File    ==> _____
  CC Type ==> _      PName ==> _____
  VolSer  ==> _____ Recfm ==> _  Lrecl ==> _____ Blksize ==> _____

  Execute if temp. Dummy ==> Y   Pass to OPO ==> N

---PF1---PF3-----
  Help  End
  
```

In this window you can define an output file of the job to be handled.

The input fields are described in *Fields: Output File Definition*.

- 3 Enter the required definitions and press ENTER.
- 4 Press PF3 to close the window.

You return to the **Files for Entire Output Management** window where the new file definition is listed.

➤ **To modify a file definition**

- In the **Files for Entire Output Management** window, enter the line command M next to the file you want to modify, and press ENTER.

Depending on the type of the file selected, either a **Spool File** or an **Output File definition window** opens.

The input fields available are described in *Fields: Spool File Definition* and *Fields: Output File Definition*, respectively.

- [Fields: Spool File Definition](#)
- [Fields: Output File Definition](#)
- [Carriage Control Type \(CC Type\)](#)

Fields: Spool File Definition

The input fields required to define a file of the job SYSOUT to be handled are provided in the **Spool File** window.

Field	Description
Operating System	(Output field only) Operating system of the node on which the selected job runs.
Spool Type	(Output field only) Spool type (POWR, JES2, JES3).
Spool file type	Spool file type, for example, SO for JES SYSOUT.
Spool file number	(Not required for z/VSE) Spool file number.
Procname	PROCNAME that identifies the file.
Stepname	STEPNAME that identifies the file.
DD name	DD name that identifies the file.

For more information on spool file types and report identification, see *Attributes of a Report* in the section *Reports* of the Entire Output Management *User's Guide*.

Fields: Output File Definition

The fields available to define an output file of the job to be handled are described in the following table.

The fields are provided in the **Output File (sequential file)** window.

Field	Description
Operating System	(Output field only) Operating system of the node on which the selected job runs.
Spool Type	(Output field only) Spool type (POWR, JES2, JES3).
File	File to be passed to Entire Output Management. If the file name contains the activation escape character, a symbol replacement is performed, using the job's active symbol table. If the file name contains the submission escape character, a symbol replacement is performed using the job's active symbol table. By using a wildcard in the file name, you may pass all files to Entire Output Management at once, which are matching the wildcard.
CC Type	Not applicable to z/OS. Carriage control type. If the file contains carriage control characters, you must specify their type: see Carriage Control Type (CC Type) .
PName	Applies to BS2000 only. If you want to identify the job by the PNAME option of the BS2000 print command (/Print . . . ,PNAME=ADAREP), enter the PNAME here.
VolSer	For z/VSE, these fields are obligatory.
Recfm	Information required to allocate the data set/file: Volume serial number (Blksize), record format (Recfm), record length (Lrecl) and the block size (Blksize) of the file/data set.
Lrecl	
Blksize	
Execute if temporary Dummy	
Pass to OPO	Relevant for execution nodes on UNIX and Windows only. If Y is entered here, the file is passed to the Open Print Option (OPO). In this case, you cannot use wildcards.

Field	Description
	Passing files to OPO is not performed by default.



Notes:

1. Entire System Server/UNIX performs wildcard file copying by invocation of a cp (UNIX) or copy (Windows) command, instead of performing the copy by own code.
2. All file copying is performed synchronously.
3. Make sure not to copy too many files at once, due to the limitation of the EntireX Broker timeout, etc.

Normally Entire Output Management identifies report definitions by spool attributes. However, if Entire Operations triggers processing of output files, no such spool attributes exist. Instead, Entire Operations attributes are passed to Entire Output Management as follows:

Entire Operations Attribute	Entire Output Management Identification		
	z/OS	z/VSE	BS2000
JOBNAME	Job name	Job name	PNAME
USER ID	Destination	Destination	User ID

Carriage Control Type (CC Type)

You must specify a carriage control (CC) type for the output file depending on your operating system and the catalog entries contained in the file catalog.

No specification is required in z/OS, where the carriage control type is taken from the file's catalog entry.

Operating System	CC Type Setting	Explanation
BS2000	<i>blank</i>	If the file was created with RECFORM=(. . . ,M) or =(. . . ,A), CC type can be taken from the file's catalog entry; otherwise, it must be specified in the CC Type field.
	A	ASA code
	M	Machine code
	E	BS2000 EBCDIC carriage control characters
z/VSE	A	ASA code
	M	Machine code
z/OS	<i>blank</i>	CC type not required

Deleting File Definitions

You can delete single or multiple file definition for Entire Output Management by either deleting the EOJ action that corresponds to the file definition(s), or by deleting a single file defined for the EOJ action.

➤ To delete an entire EOJ action

- 1 On the **End-of-Job Checking + Actions** screen, enter the line command **D** next to the action which contains the file definitions for Entire Output Management, and press **ENTER**.

A confirmation window opens.

- 2 Enter **Y** (Yes) to confirm the deletion.

The EOJ action with all file definitions for Entire Output Management is deleted.

➤ To delete single files defined for an EOJ action

- 1 On the **End-of-Job Checking + Actions** screen, enter the line command **T** next to the action which contains the file definitions for Entire Output Management you want to delete, and press **ENTER**.

Depending on where you enter the **T** command (above or below **Bottom of Data**),

An **Additional Actions for Job-ok** window or **Job ended ok/Job ended not ok** with a list of files defined for Entire Output Management opens.

- 2 Enter the line command **D** next to the file you want to delete, and press **ENTER**.

A confirmation window opens.

- 3 Enter **Y** (Yes) to confirm the deletion.

Handing Over SYSOUT and Files to Entire Output Management

This section covers the following topics:

- [Repetition if File is missing or open](#)
- [Copying Files before passing them to Entire Output Management](#)
- [File Copy to another Entire System Server Node](#)
- [Connection with Defined Recovery Action](#)
- [SYSOUT Line Limit](#)
- [Passing SYSOUT from UNIX and Windows Nodes to Entire Output Management](#)

- [Naming Rules for Files copied into the EOR_NOM Directory \(UNIX and Windows\)](#)

Repetition if File is missing or open

If the file to be handed over did not (yet) exist or was (not) yet closed, the handover will be retried.

The following rules apply:

Case	Retries	Interval between retries
Output file cannot be found (does not apply to spool files)	3	5 minutes
All other cases	1000	30 minutes

The retry queue will not be deleted in case of a Monitor shutdown and restart.

Copying Files before passing them to Entire Output Management

In the default settings, you can define that the SYSOUT files are copied physically before handover to Entire Output Management.

The advantage of this is that exclusive use (as well as renaming and deleting) of the copy by Entire Output Management can still be looked at in Entire Operations. This, however, means that SYSOUT files need more space.

File Copy to another Entire System Server Node

Under the following conditions, the SYSOUT copy is created on another node:

- The symbol [SYSOUT-NODE-GLOBAL](#) exists in the symbol table SYSDBA/A and contains a valid node number.
- In the target environment, the same PubIDs and BS2000 user IDs exist as in the source environment.

Connection with Defined Recovery Action

Defined recovery actions for an erroneous job are only executed, after all activities for the handover of the SYSOUT to Entire Output Management have been concluded, or after the maximum number of handover tries has been reached.

In case of a recovery action, the SYSOUT of all job runs are handed over to Entire Output Management. For further information, see [Defining Recovery Action](#).

SYSOUT Line Limit

If the SYSOUT file or spool data set exceeds a given line limit, the copied file will be truncated: see **SYSOUT Line Limit** in *Default Setting (2)* in the *Administration* documentation.

Passing SYSOUT from UNIX and Windows Nodes to Entire Output Management

On UNIX and Windows, the environment variable `EOR_NOM` must be defined. It must contain a valid directory name. Entire Operations copies the SYSOUT files for Entire Output Management into this directory. If the directory does not exist, Entire System Server/UNIX (Entire System Server/Windows) creates it at the first usage attempt.

➤ To define the copying of SYSOUT to Entire Output Management

- 1 On the **End-of-Job Checking + Actions** screen, enter the line command `S` next to the required action, and press `ENTER`.

The **SYSOUT Actions** window opens.

- 2 Enter the line command `0` to select the option **Pass SYSOUT to Entire Output Mgmt**. See also *Defining Job SYSOUT Actions* in the section *SYSOUT Actions*.

This is a job-level definition that only applies to the selected job.

Entire Operations creates the SYSOUT file copies in this directory with unique names.

It is up to Entire Output Management to fetch the files from there.



Notes:

1. If copied SYSOUT files are not fetched by Entire Output Management within the retention period for active jobs, they are removed by Entire Operations cleanup after that time (only if Entire System Server Version 2.1.2 PL 2 or higher is in use). It is recommended to use Entire System Server Version 2.1.2 PL 2 or higher in conjunction with this functionality.
2. If the SYSOUT file or spool data set exceeds a given line limit, the copied file will be truncated: see **SYSOUT Line Limit** in *Default Setting (2)* in the *Administration* documentation.

Naming Rules for Files copied into the EOR_NOM Directory (UNIX and Windows)

SYSOUT Files

The file names for SYSOUT files for the \$EOR_NOM directory have the following syntax:

```
dbid-fnr.owner.network.run.job.txt
```

File Name	Meaning
<i>dbid-fnr</i>	Concatenated DBID and FNR, 10 digits.
<i>owner</i>	Owner of job.
<i>network</i>	Network of job.
<i>run</i>	Run number of job.
<i>job</i>	Job.

Other Files

The file names for other files for the \$EOR_NOM directory have the following syntax:

Format A with the components:

```
yymmdd.hhisst.rrrrr.filename.ext
```

File Name	Meaning
<i>yymmdd</i>	Year, month, day.
<i>hhisst</i>	Time, incl. 1/10 sec.
<i>rrrrr</i>	Entire Operations run number, with leading zeros.
<i>filename</i>	Unqualified part of original file name.
<i>ext</i>	Extension of original file name.

Format B with the components:

```
yymmdd.hhisst.rrrrr.job.ix
```

File Name	Meaning
<i>yymmdd</i>	Year, month, day.
<i>hhisst</i>	Time, incl. 1/10 sec.
<i>rrrrr</i>	Entire Operations run number, with leading zeros.
<i>job</i>	Entire Operations job name.
<i>ix</i>	Index of file definition in Entire Operations (2 digits).



Note: Format B is used only if the length of *filename.ext* is longer than 20 characters, because of a size limitation in Entire Output Management.

Defining the Handover of SYSOUT Files from UNIX/Windows to BS2000

> To define the handover of SYSOUT files from UNIX or Windows to BS2000

- Press PF9.

The following window opens:

```

24.10.08          ***** Entire Operations *****          16:40:39
Owner   SN          End-of-Job Checking + Actions   WIN/NT   Job AP01
Network WNT409A          Run          Date
-----
C Action          +-----+ ans OA
-                !                !
-                ! SYSOUT Copy from UNIX / Windows to Mainframe !
-                !                !
-                ! Owner      SN          SYSOUT Node   31          !
-                ! Network   WNT409A      SYSOUT Cat ID !
-                ! Run              SYSOUT User ID SN          !
-                ! Job        AP01          !
-                ! Exec.Node  409          !
*****          !                ! *****
s          S      ! Note: Please use Job Definition function 'Specials' !
-                !           (PF6) to modify the values          !
-----          !                ! -----
A Activation      ! ---PF1---PF3----- ! Other
P Descr. R Re    ! Help End          ! ol Set
-                !                !
Command => ___ +-----+
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Add End Save Up Down Menu
    
```

You can copy a UNIX or Windows SYSOUT file to a BS2000 system so that it can be handed over to the Output Management for instance.

The SYSOUT node must differ from the execution node, and it must be a BS2000 node.

The naming conventions for the copy are identical to the ones for SYSOUT files, which have been directly created on the mainframe. (Even **exits for SYSOUT file names**, which may be present, are run through.) SYSOUT copies are affected by the automatic file cleanup just as the original SYSOUT.

This section covers the following topics:

- [Accessing SYSOUT File Names in User Exits](#)

- [Field Descriptions](#)

Accessing SYSOUT File Names in User Exits

You can retrieve the names of SYSOUT source files (on UNIX or Windows) and target files (on BS2000) for end-of-job checking (EJC) and end-of-job action (EJA) user exits (see also [Editing End-of-Job Checking or Action User Exits](#)).

If you want to retrieve the file names, add the following to the definitions in the user exit:

- For the SYSOUT source file:

```
INDEPENDENT
1 +P-SYSOUT-FILE-250      (A250)
```

- For the SYSOUT target file:

```
INDEPENDENT
1 +P-SYSOUT-FILE-COPY-250 (A250)
```

For BS2000 target nodes, only the leading 54 bytes of the field are used.



Notes:

1. Press PF6 (Specials) on the [Job Definition \(Master\) window](#) to modify the values. See [Defining Job Type Specific Execution Features](#) in the section *Job Maintenance*.
2. The application of this function means an additional working effort for the Entire Operations Monitor, which depends on the number of the handovers and on the size of the SYSOUT files.
3. A log message is written at the beginning and at the end of a SYSOUT copy from UNIX or Windows to the mainframe.
4. The resubmission of a job with an ongoing SYSOUT copy is inhibited.
5. No SYSOUT copying will be attempted if the job has the type DUM (dummy) or if it is a temporary dummy job (see also [Job Execution as a Dummy Job](#) in the section *System Overview*).
6. Both SYSOUT source and target file names can vary depending on the repeat count of the active job. Refer to the reserved symbol P-REPEAT-COUNT (alias P-REPEAT) described in [Table of Pre-defined Symbols](#) in the section *Symbol Table and Symbol Maintenance*.
7. File truncation: If the SYSOUT file or spool data set exceeds a given line limit, the copied file will be truncated: see **SYSOUT Line Limit** in *Default Setting (2)* in the *Administration* documentation.

Field Descriptions

Field	Meaning
SYSOUT Node	A valid Entire System Server node on BS2000.
SYSOUT Cat ID	BS2000 Cat ID, which is to be used for the copied file (optional).
SYSOUT User ID	BS2000 User ID, under which the copied SYSOUT file is to be generated.

XIX

Special Monitor Features and Batch Jobs

76 Special Monitor Features and Batch Jobs

- Monitor Start Network 880
- Day Start Network 880
- Common Start Network Considerations 881
- Cleanup in Batch Mode 881

Monitor Start Network

You can define a job network to be executed *after each Monitor start* and before the activation of any other job.

If a network with the name MON-START is defined under the owner SYSDBA, it is executed at Monitor start time.

Execution

The start network is intended to run exclusively before any other network. Therefore, the absolute condition MON-START-RUNNING (owner SYSDBA) is set at activation time.

The setting of this condition is automatically taken over by the first job of the start network. This job sets no conditions during End-of-Job checking and actions.



Note: The absolute condition MON-START-RUNNING is to be reset, only if the whole start network ends normally. Any other activity of the Monitor is blocked during execution of the start network. If any error occurs in the start network, the whole processing of other networks is blocked until there is a manual intervention. To force the normal processing to start, just reset the condition MON-START-RUNNING manually.

Day Start Network

You can define a job network to be executed *at the start of each day* and before the activation of any other job.

If a network with the name DAY-START is defined under the owner SYSDBA, it is executed at Monitor start time.

Day Start Execution

The start network is intended to run exclusively before any other network. Therefore, the absolute condition DAY-START-RUNNING (owner SYSDBA) is set at activation time.

The setting of this condition is automatically taken over by the first job of the start network. This job sets no conditions during End-of-Job checking and actions.



Note: The absolute condition DAY-START-RUNNING is to be reset only if the whole start network ends normally. Any other activity of the Monitor is blocked during execution of the start network. If any error occurs in the start network, the whole processing of other

networks is blocked until there is a manual intervention. To force the normal processing to start, just reset the condition DAY-START-RUNNING manually.

Common Start Network Considerations

- [Exclusive Execution](#)
- [Use](#)

The considerations in this section apply to the Monitor start network and the day start network.

Exclusive Execution

While a start network is running, a warning message is repeatedly written to the log.

During the execution of the start network, the following Monitor activities are blocked:

- Schedule extraction
- Activation (except start network)
- Cleanup



Note: The Monitor start network and the day start network can execute in parallel.

Use

Some possibilities for the use of the start network are:

- Preparation of symbol tables for other networks;
- Activation of other networks;
- Condition setting;
- Any Entire System Server functions.

Cleanup in Batch Mode

- [Parameter for CLEAN](#)
- [Example for CLEAN](#)
- [Cleanup of Mailbox Messages to SYSDBA](#)
- [Parameter for MX-DEL1P](#)

▪ [Example for MX-DEL1P](#)

In addition to an automatic cleanup, or alternatively, you can also do a cleanup of the active database in batch mode outside of the Entire Operations Monitor.

Use a Natural batch job with the following commands:

```
LOGON SYSEOR
CLEAN <function> <log-target>
FIN
```

Parameter for CLEAN

Use	Name	Description
function	ALL	Cleanup, deactivation, deletion of work files
	CLEAN	Cleanup only
	DEACT	Deactivation, deletion of work files only
log-target	NOP	The cleanup is logged in the Entire Operations Log.
	SO	The cleanup is logged in the SYSOUT of the batch job.
	ALL	The start and end messages of the cleanup are also logged in the Entire Operations Log.
		The cleanup is both logged in the Entire Operations Log and the SYSOUT of the batch job.

The batch cleanup can take place with the Monitor running or not. It is recommended to do the cleanup at times of low monitor utilization.

Like the automatic cleanup the batch cleanup should be performed at least once a day. Several runs a day are possible. This may be useful to reduce the data amount of each cleanup run.

The Entire Operations retention periods also apply to batch cleanup.

Example for CLEAN

```
CLEAN ALL SO complete cleanup, log after SYSOUT
```

Cleanup of Mailbox Messages to SYSDBA

The cleanup of mailbox messages to SYSDBA can be executed in batch mode. To do this, use a Natural batch job including the following commands:

```
LOGON SYSEOR
MX-DEL1P <yyyymmdd> <hhiiss>
FIN
```

Parameter for MX-DEL1P

Name	Description
yyyymmdd	Day up to which cleanup is to take place.
hhiiss	Time up to which cleanup is to take place.

Example for MX-DEL1P

```
MX-DEL1P 20020201 100000
```

