

User's Guide

Version 5.4.3

December 2017

This document applies to 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 © 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: OGC-ONOPUSERGUIDE-543-20180305

Table of Contents

Preface	xv
I System Overview	1
1 System Overview	3
Operating System Classes and Related Operating Systems	5
Entire Operations User IDs	5
Operating System User IDs	6
Owner	8
Job	9
Job Network	10
Subnetworks	11
Logical Conditions	11
Prerequisite Check	13
Events	19
End-of-Job Checking and Actions	19
Resources	20
Mailboxes	23
Operating System Server Nodes	25
Master Database and Active Database	26
Monitor (Server)	27
Monitor Start Network	29
Activation of Job Networks or Jobs	29
Run Number	34
Schedules	34
Calendars	35
Symbol Tables and Symbols	35
Job Control (JCL)	36
Dynamic JCL Generation (JCL Location MAC)	37
Accessing Entire Operations from other Applications	37
Job Execution as a Dummy Job	38
Logging Facility	38
Message Sending	39
System Messages	39
User Language	40
Reporting	41
Cross References	41
Editor	41
Cleanup of the Active Database	42
II Using Entire Operations GUI Client	43
2 Online Help	45
Error Messages	46
Online Technical Information	47
3 Starting and Ending an Entire Operations Session	49
4 Logging on and off an Operating System Server Node	53

Logon Function	54
Fields: Node Logon	55
Monitoring the Node Connection Status	56
Logoff Function	58
5 Elements of the Main Application Window	61
Object Workspace	63
Menu Bar	66
Options Menu	67
My Desktop Menu	69
Toolbar	70
Content Pane	70
Results Window	71
Command Line	71
Status Bar	72
Context Menu	72
6 Common and Global Functions	75
Canceling Reading	110
Applying Changes	76
Listing Objects	76
Refreshing Object Lists	77
Filtering Objects	78
Saving a List View as a Report File	80
Drag & Drop	80
Displaying Objects	82
Copying Objects	82
Pasting Objects	83
Deleting Objects	84
Importing and Exporting Objects	84
Functions of the Metanode General	84
Monitor Status	85
Show Messages	86
Add to Workplan	87
Show Workplan	87
Show all Owners	91
Show linked Owners	91
7 Date and Time Formats	93
Date Display and Input Options	94
Time Display and Input Options	95
III Entire Operations Utilities	97
8 Entire Operations Utilities	99
Introduction	100
Extract Log Data to Log Selection File	101
Delete Old Data in Log Selection File	102
Output of Log Data to a File	103
Print Account Information from Entire Operations Log	103

Mass Change of Network / Symbol Table Version	104
Monitor or Task Wait Time Modification	105
Monitor Shutdown	105
Monitor Start	106
BS2000 Jobs	106
List or Delete TO-ACTIVATE Command Records	107
Check the Existence of Symbol Table Definitions	108
Mass Change of the Owner / Delete Owner with all Objects	108
Mass Change of Access Rights Granted to Networks	111
User ID Mass Update in Network and Job Definitions	113
Mass Change of Node Numbers	114
Mass Logon Processing in Batch Mode	114
Bulk Execution of MACRO Commands	116
Migration of Log Data from NOP Versions lower than 411 to NOP 412 and above	118
Data Migration to the current Entire Operations Version	120
IV Using Entire Operations in Batch Mode	123
9 Using Entire Operations in Batch Mode	125
Files Needed to Start Entire Operations in Batch	126
Files Needed to Start the Batch Command Client	131
Command Syntax for the Batch Command Client	133
Commands for the Batch Command Client	138
V Using Owners	161
10 Using Owners	163
Available Functions: Owner	164
Owner at Logon	165
Linking Additional Owners	165
Owners Granted Access to Individual Networks	166
VI Network Maintenance	169
11 Maintaining Job Network Definitions	171
Listing all Network Definitions	172
Available Functions: Network Master	173
Displaying a Network Definition	175
Modifying a Network Definition	175
Adding a Network Definition	178
Fields: Network Definition	179
OS Specials - Operating System and Environment Defaults	181
Message and Message Recipients - Specifying Recipients for Network Messages	186
Granting Definition: Authorizing Other Users or Owners to Access a Network	187
Long Description - Documenting Your Networks	191
Deleting a Job Network	192
12 Copying Job Network Definitions; Cloning Network Versions	195
Fields: Paste Data Network Master	197

13 Viewing and Maintaining a Job Network Diagram	199
Explanations of Diagram Symbols	200
Maintenance Functions for Diagrams	204
Editing and Navigating in the Network Diagram	209
Examples of Diagrams	211
14 Checking for a Loop in a Job Network	215
15 Maintaining the Usage of Network Versions	217
Handling Network Version Usage Definitions	218
16 Applying Network Defaults to Jobs (Mass Update)	221
17 Activating a Job Network Manually	225
Fields: Network Activation	227
VII Job Maintenance	229
18 Defining and Managing Jobs	231
Listing Jobs	232
Available Functions: Job Master	234
Displaying and Modifying a Job Definition	235
Adding a Job Definition	242
Writing and Viewing Online Documentation for a Job	243
Deleting a Job Definition	245
Activating a Single Job Manually	245
Displaying Job Dependencies	247
Defining Job-Specific Log Information	248
19 Defining Job Types and Job Execution Features	251
Available Job Types	252
Defining Job Type Specific Execution Features	254
Defining Operating System Specific Execution Features	257
20 Using a Dummy Job	263
Dummy Job Execution by the Monitor	264
Permanent Dummy Jobs	264
Temporary Dummy Jobs	264
Excluding a Job from Actual Execution	265
End-of-Job Actions after Execution as a Temporary Dummy Job	266
Supervision of Running Jobs	267
21 Defining a Subnetwork	269
Link to the Main Network	271
Subnetwork Activation and Execution	272
Restrictions for Subnetworks	273
Adding or Modifying a Subnetwork Definition	274
Listing Jobs of a Subnetwork	276
22 Defining Parameters for an FTP Job	277
Adding an FTP Parameter Definition	278
JCL Generation for FTP Jobs	280
23 Defining and Managing JCL for a Job	283
Job Control for Jobs under BS2000	284
Job Control for Jobs under UNIX	284

Job Control for Jobs under Windows	285
Defining Master JCL for a Job	288
Displaying Master JCL	292
JCL Locations	293
Defining Operating System Dependent JCL Specifications	297
Handling JCL during Job Submission	298
Symbol Replacement in JCL	301
Pregenerating Active JCL	301
24 Editing Master JCL and Natural Sources	303
General Editing Considerations	304
UNIX and Windows: Handling of Tab Characters (H'09') within JCL Lines	305
Usage of Text Objects in JCL	305
Using the Editor	305
Editing Macro Sources for Dynamic JCL Generation	310
Locking of Natural Sources	319
25 Defining and Managing Job Input Conditions	321
Listing Input Conditions	322
Adding and Modifying a Master Input Condition	325
Accessing another Network Defined for an Input Condition	329
Input Condition References	330
Global Conditions	333
Input Condition: User Exit	333
Input Condition: Multiple Suffixes	336
Input Condition: File Existence	337
Input Condition: Mailbox	339
Input Condition: Symbol Value	340
Input Condition: User Switch - BS2000	343
Input Condition: Job Variable - BS2000	344
Listing Jobs Linked to an Input Condition	346
Deleting an Input Condition Definition	347
26 Handling Prerequisite Resources for a Job	349
Listing Prerequisite Resources Defined for a Job	350
Viewing the Usage of a Prerequisite Resource	351
Displaying, Modifying and Adding a Prerequisite Resource Definition	352
Deleting a Prerequisite Resource Definition	353
Columns and Fields: Prerequisite Resource Definitions	354
VIII	357
27 End-of-Job Checking and Actions	359
28 End-of-Job Checking and Actions - Overview	361
List of End-of-Job Checking and Actions	362
Kinds of End-of-Job Actions	363
End-of-Job Checks for Various Operating Systems	364
29 Adding and Modifying Events and Actions	365
30 End-of-Job Checking and Actions: Columns EOJ Checking Page	369

Columns: End-of-Job Checking and Actions	370
31 Adding Event Definition for z/OS or z/VSE Job	371
Field Descriptions: Add/Modify Event Definition	372
32 Modifying Event Definition for z/OS or z/VSE Job	375
Example of an A-type Event	376
Example of a C-type Event	377
Example of an R-type Event	378
Example of an S-type Event	379
33 Adding Event Definition for BS2000 Job	381
Field Descriptions	382
34 Modifying Event Definition for BS2000 Job	385
Example of an S-type Event	386
Example of a U-type Event	387
Example of a V-type Event	388
Adding or Modifying a Job Variable	390
Example of a J-type Event	390
35 Adding Event Definition for UNIX or Windows Job	393
Field Descriptions	394
Exit Code Check Error Handling	395
36 Modifying Event Definition for UNIX or Windows Job	397
Example of an S-type Event	398
37 EOJ Checking Defaults for Various Operating Systems	401
Common Defaults for Event Checking	402
z/OS Defaults for Event Checking	402
z/VSE Defaults for Event Checking	403
BS2000 Defaults for Event Checking	403
UNIX and Windows Defaults for Event Checking	404
38 Adding Output Condition Definitions	405
Fields and Columns: Output Conditions	406
39 Define Symbol Modification or Setting	409
Field Descriptions: Symbol Modification	410
40 Editing End-of-Job Checking or Action User Exits	411
Creating or Editing User Exits	412
Example	412
41 Defining Activation of Jobs or Job Networks	415
42 Defining Recovery Action	417
Recovery Action Definition	418
System Symbols for Recovery Actions	419
43 SYSOUT Actions	421
Defining Job SYSOUT Actions	422
44 Message Sending	423
Introduction	424
Fields and Columns: Message and Message Recipients	424
Message Destination Types	425
Message Recipients	426

E-Mail on Mainframes	426
E-Mail on UNIX and Windows Systems	427
45 Defining End-of-Job Action User Exit	429
46 End-of-Job Action: Release Resource	431
IX Active Job Networks	433
47 Maintaining Active Job Networks	435
Available Functions: Network Active	436
Listing Active Job Networks	436
Deactivating Active Job Networks	437
Viewing the Execution History of a Network Active	439
Maintaining Active Runs	440
48 Maintaining Active Jobs	445
Available Functions: Job Active	446
Listing Active Jobs	448
Listing Jobs of an Active Subnetwork	453
Adding a New Job to the Active Network	454
Modifying the Latest Start Time for an Active Job	456
Choosing the Job ID of a Job for Logging	457
Canceling, Holding and Releasing Active Jobs	458
Resubmitting an Active Job	460
Deactivating a Job in an Active Network	462
Reactivating an Active Job	465
Displaying and Modifying an Active Job Definition	465
Modifying EOJ Checking and Actions	467
Viewing Long Descriptions of Active Jobs	468
Displaying Prerequisites for Active Jobs: Waiting for	468
Viewing and Modifying Resources Used by Active Jobs	470
Browsing Active Job SYSOUT	471
49 Maintaining Active Job Conditions	477
Available Functions: Condition Active	478
Listing Active Conditions	479
Viewing an Active Condition and Changing Its Status	481
Adding an Active Condition	481
Using Global Active Conditions	482
Viewing the Usage of Conditions Concatenated with an Active Job	485
Viewing the Last Job Used by a Condition Active	486
50 Maintaining Active JCL (Job Control Language)	487
Browsing Active JCL	488
Editing Active JCL	489
Viewing the Extended Log with Active JCL Changes	490
Exchanging Active JCL	491
Regenerating Active JCL	491
X Schedule Maintenance	493
51 General Scheduling Considerations	495
Optional Schedule Definition	496

Schedule Extraction Times	496
Manual and Automatic Activations on the same Day	496
Multiple Network Activations	496
Influence of Deactivations on Schedules	496
Imported Schedules	497
Schedule Dependencies across the Turn of the Year	497
Using Calendars	497
52 Maintaining Schedules	499
Listing Schedules	500
Displaying, Adding or Modifying a Schedule	501
Deleting a Schedule Definition	512
53 Scheduling a Network	513
Defining a Network Schedule	514
Defining and Deleting Time Frames for a Network	518
Defining Dates for Explicit Network Execution	523
Defining Multiple Network Activations	526
Viewing a Network Schedule Definition as a Calendar	529
Producing a Network Start Summary	531
Displaying Next Network Starts - Next Activations	531
Displaying the Network Execution History	536
54 Scheduling a Job	539
Defining Scheduling Parameters for a Job	540
Maintaining Schedule Dependencies for a Job	542
Defining Schedule Dependencies for an Input Condition	546
Sending a Late Message to One or More Users	546
Viewing Job Accounting Data	547
XI Calendar Maintenance	551
55 Calendar Maintenance	553
General Rules and Restrictions	554
Available Functions: Calendar	555
Listing Calendars	556
Displaying, Modifying or Adding a Calendar Definition	557
Defining Workdays and Holidays	559
Where Used - Listing Schedules Using a Calendar	560
Deleting a Calendar Definition	561
XII Working with Mailboxes	563
56 Working with Mailboxes	565
Available Functions: Mailbox Definition	566
Viewing Mailbox Messages	567
Fields, Columns and Functions: Messages and Requests	568
Message Types	570
Origin of Mailbox Messages	571
Group Mailbox	572
SYSDBA Mailbox	572
User Mailbox	572

Clearing Mailboxes	572
XIII Log Information	573
57 Log Information	575
Displaying Logged Information - Browse Log Function	576
Monitoring Entire Operations Activities	580
XIV Symbol Table and Symbol Maintenance	583
58 Purpose and Use of Symbol Tables and Symbols	585
Symbol Tables	586
Symbols	587
Subnetworks and Recovery Jobs	589
User Exits for User-specific Symbol Maintenance Tasks	589
59 Adding and Maintaining Symbol Tables	591
Available Functions: Symbol Table Master	592
Listing Symbol Table Masters	593
Displaying a Symbol Table Master	594
Adding a Symbol Table Master	595
Modifying a Symbol Table Master	598
Maintaining the Usage of Symbol Table Versions	601
Saving Symbol Tables as Files	604
Using Active Symbol Tables	605
Deleting a Symbol Table Master	608
60 Finding Symbol Tables Associated with Jobs and Networks	609
Listing Usable Symbol Tables	610
Listing Jobs and Networks that use a Symbol Table Master	611
61 Defining Symbols and Symbol Values	615
Available Functions: Symbol Master	616
Listing Symbol Masters	617
Viewing a Symbol Master Definition	617
Modifying a Symbol Master	618
Adding a Symbol Master	619
Maintaining a Symbol Active	619
Defining Multiple Symbol Values	623
Specifying a Range Check for Numeric Symbol Values	624
Predefined Symbols	625
Validating Symbol Values with a User Exit	633
Global Symbol Modification Exit	633
Symbol Setting triggered by the SYSOUT of a Job	633
Deleting a Symbol from a Symbol Table	634
62 Symbol Replacement	637
Symbol Escape Characters	638
Symbol Prompting during Network or Job Activation	640
Specifying User Exits for Symbol Modification	642
Rules and Restrictions for Symbol Replacement	645
63 Functions for Symbol Replacement	651
Results Returned by a Symbol Function	652

Function !D or ?D - Date from Today Calculation	653
Function !E or ?E - Date from Date Calculation	659
Functions !MV or ?MV and !MM or ?MM - Access to Multiple-Value Symbols	660
Function !TIMN or ?TIMN - Constant Time Values	662
Function !W or ?W - Date Calculation with Reference Calendar	663
XV Reporting	667
64 Reporting	669
Report Types	670
Viewing the Report Status List	672
Generating Online Reports	674
Viewing Report Properties and Deleting a Report	675
Fields and Columns: Reporting	676
Determination Date for Report Data	680
Report Output Options	681
Using Reports with Bar Charts	681
User Restrictions for Reports	685
Examples of Reports	686
Generating Batch Reports	697
XVI Cross References	699
65 Cross References	701
Types of Cross Reference Reports	702
Generating Cross Reference Reports Online	703
Fields and Columns: Cross References	709
Examples of Cross Reference Reports	714
Generating Cross Reference Reports in Batch	719
XVII API Routines	721
66 Using an API Routine	723
API Usage Rules and Restrictions	724
Invoking the API Routines	725
Example of Use of API Routines	725
67 List of Available Entire Operations API Routines	727
EORUCB1N: Check Use of BS2000 User IDs	728
NOPFB2-N: Generate SYSOUT File Names for BS2000	728
NOPMLA1N: Start and Stop the Monitor Activity Log	731
NOPU--1N: Read Network Available to a Specific User	732
NOPU--2N: Return all Usable Symbol Tables for a Network	733
NOPU--3N: Get Correlation ID for an Activated Network	734
NOPU--4N: Store New Event in Entire Operations System File	734
NOPUAC5N: Activate Job Networks or Jobs	735
NOPUAS1N: Retrieve Numbers of Active Jobs in Defined Status Ranges	738
NOPUCN3N: Access Entire Operations Conditions	740
NOPUCS1N: Access Calendars and Schedules	741
NOPUJI4N: Import Operating System Jobs into the Active Queue	745

NOPUJS2N: Job Schedule Inquiry and Modification	746
NOPULW9N: Write Messages to System Automation Tools Log	748
NOPUMI1N: Set/Reset Text Milestones in Master and Active Jobs	749
NOPUMT3N: Expand Message Texts	750
NOPUNI1N: Invalidate Entire System Server Node Table Entries	752
NOPUNX1N: Entire System Server Calls to Access UNIX and Windows Files	752
NOPURE2N: Handle Resource Allocations	760
NOPURS1N: Access Entire Operations Master Resources	763
NOPUSN2N: Inquire Calling Job or Called Network for Subnetworks	765
NOPUSP3N: Display Long Texts for Symbol Prompting	766
NOPUST3N: Inquire Network and Job Status, Symbol Table	768
NOPUSY6N: Access Entire Operations Symbols	772
NOPUVI2N: Obtain Entire Operations Version Information	777
NOPUXD1N: Maintain End-of-Job User Exits for a Network	778
NOPUXI0N: Add Input Condition to an Activated Job	779
XVIII User Exits	781
68 User Exits	783
What User Exits Can Do	784
Global User Exits	785
Front-end User Exits	785
Common User Exit Parameter Data Area NOPXPL-A	785
Generation of Dynamic JCL and SYSOUT File Names (BS2000 only)	791
User Exits for Resource Master Determination	792
User Exits for Setting Input Conditions	792
User Exits for End-of-Job Checking and Actions	792
User Exits for Symbol Functions	794
User Exits for Validation Checks of Symbol Values	797
XIX Passing Files to Entire Output Management	803
69 Passing Files to Entire Output Management	805
Basic Requirements for File Transfers to Entire Output Management	806
Listing Files Defined for Entire Output Management	807
Adding and Modifying File Definitions	809
Deleting File Definitions	813
Handing Over SYSOUT and Files to Entire Output Management	814
XX Special Monitor Features and Batch Jobs	819
70 Special Monitor Features and Batch Jobs	821
Monitor Start Network	822
Day Start Network	822
Common Start Network Considerations	823
Cleanup in Batch Mode	823

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.



Note: Since the Entire Operation functions on the Windows client correspond to the functions on a mainframe or UNIX host, this documentation also contains descriptions and examples that reference the screens and commands of the Character User Interface (CUI) application.

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 GUI Client	Describes the main functions of Entire Operations GUI Client.
Entire Operations Utilities	Describes Entire Operations utilities.
Using Entire Operations in Batch Mode	Describes the files and commands required to execute Entire Operations in batch mode.
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.

Basic Functions	
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.
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.

I System Overview

1 System Overview

▪ Operating System Classes and Related Operating Systems	5
▪ Entire Operations User IDs	5
▪ Operating System User IDs	6
▪ Owner	8
▪ Job	9
▪ Job Network	10
▪ Subnetworks	11
▪ Logical Conditions	11
▪ Prerequisite Check	13
▪ Events	19
▪ End-of-Job Checking and Actions	19
▪ Resources	20
▪ Mailboxes	23
▪ Operating System Server Nodes	25
▪ Master Database and Active Database	26
▪ Monitor (Server)	27
▪ Monitor Start Network	29
▪ Activation of Job Networks or Jobs	29
▪ Run Number	34
▪ Schedules	34
▪ Calendars	35
▪ Symbol Tables and Symbols	35
▪ Job Control (JCL)	36
▪ Dynamic JCL Generation (JCL Location MAC)	37
▪ Accessing Entire Operations from other Applications	37
▪ Job Execution as a Dummy Job	38
▪ Logging Facility	38
▪ Message Sending	39
▪ System Messages	39
▪ User Language	40
▪ Reporting	41
▪ Cross References	41

- Editor 41
- Cleanup of the Active Database 42

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.

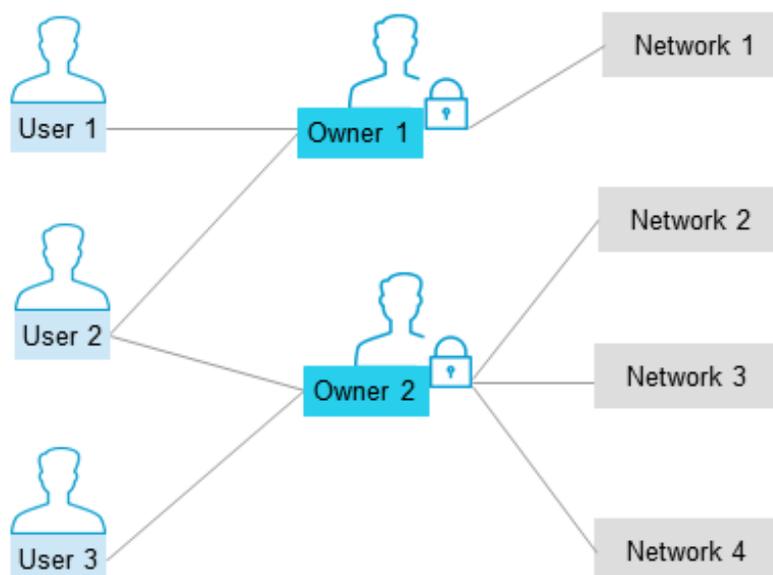


Note: In special cases, a user can be authorized to access networks belonging to other owners. The owner SYSDBA is authorized to access the networks of all owners.

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 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*.

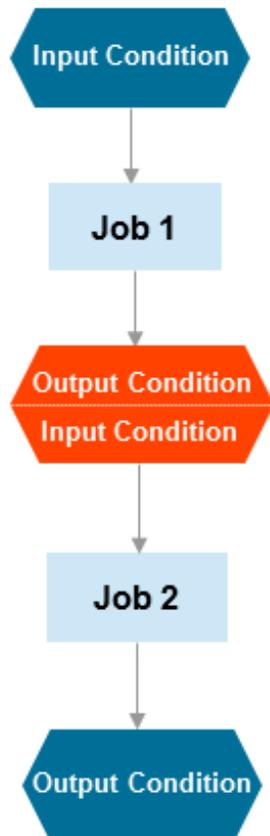
Input conditions can refer not only to the current run of a job network, but also to given time frames in the past or to previous runs.

You can also use an input condition to turn a job into a dummy job when it occurs.

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.



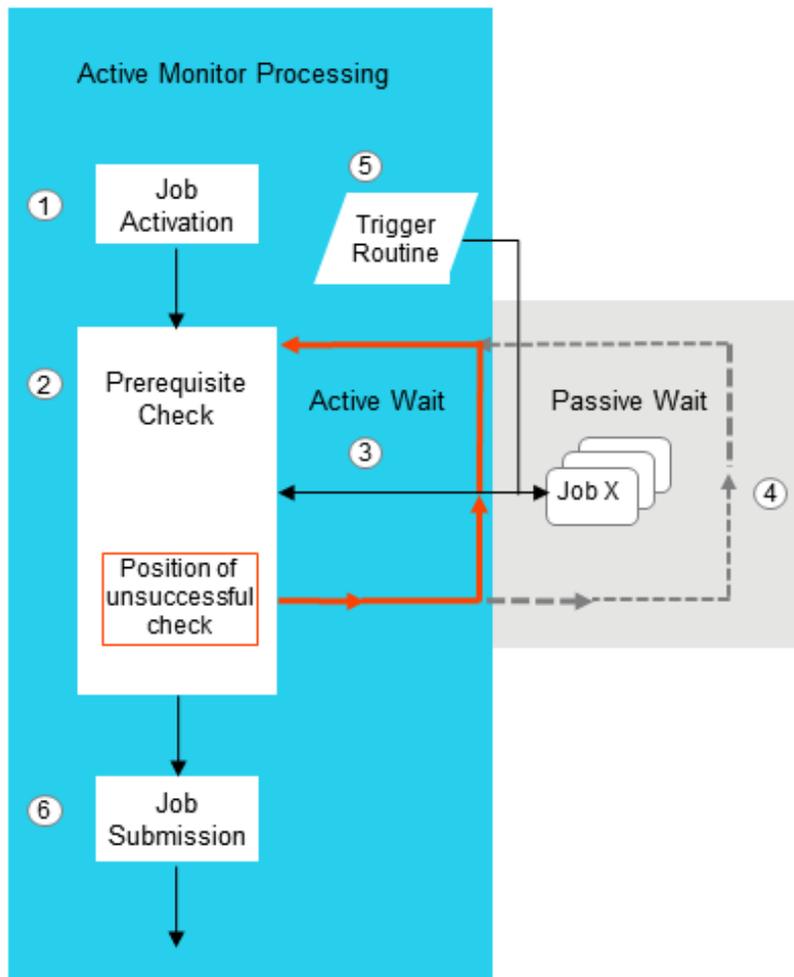
Note: The main passive wait release routine does not re-activate the waiting jobs at the same time. Instead, it performs the release in portions of 300 jobs. Between the portions,

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.

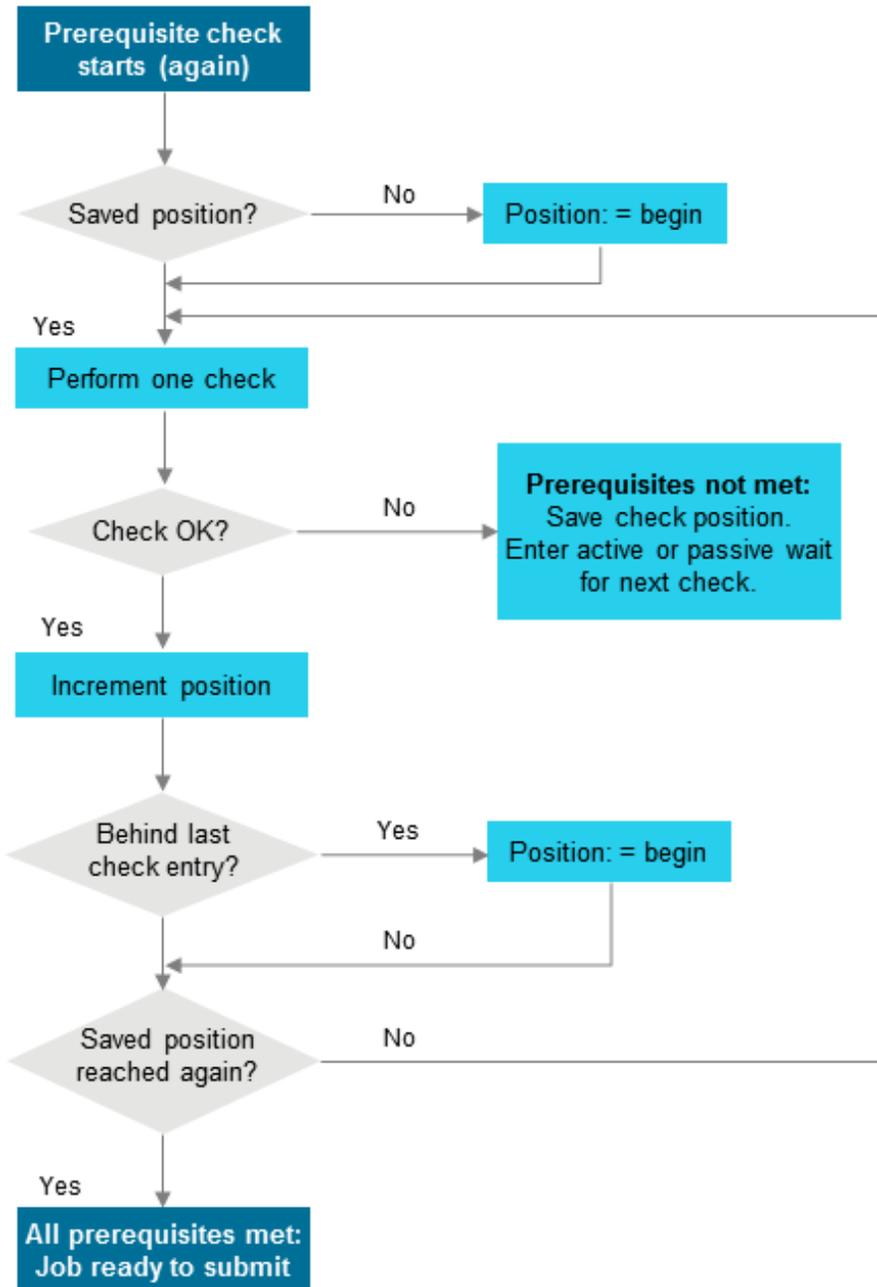
In other words: it is recommended to replace a wait for prerequisites with special dependencies by a wait for normal prerequisites.

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 option **Deallocate if job not ok** in the [Resource Prerequisite 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.

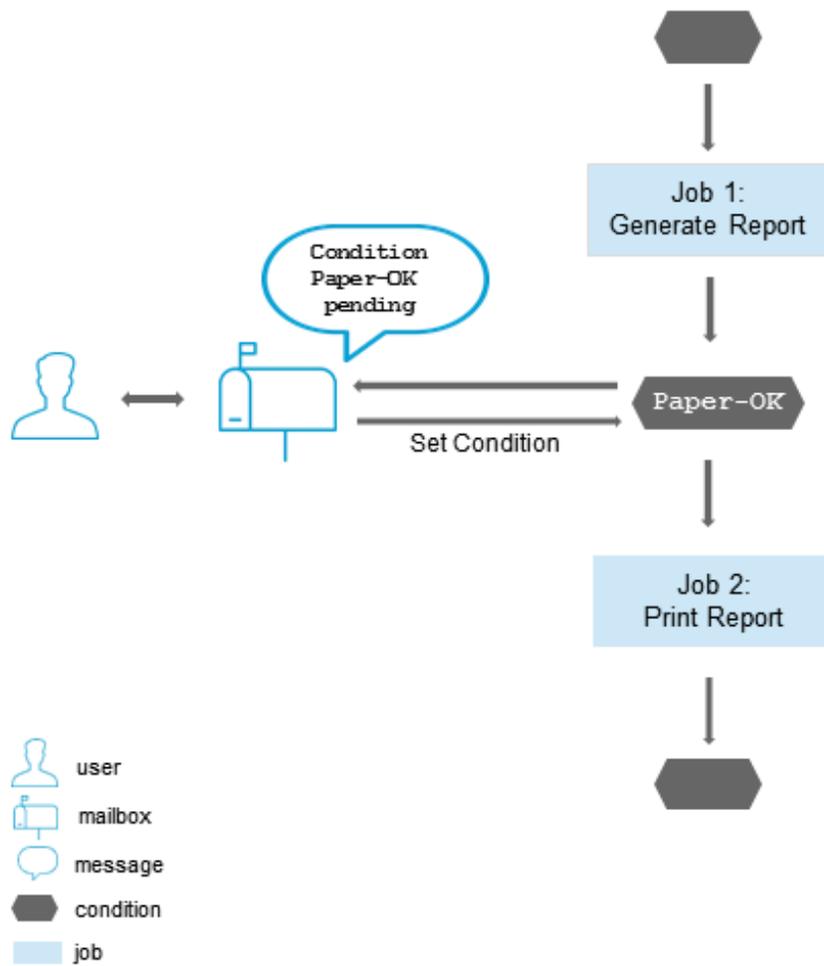
Mailboxes

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

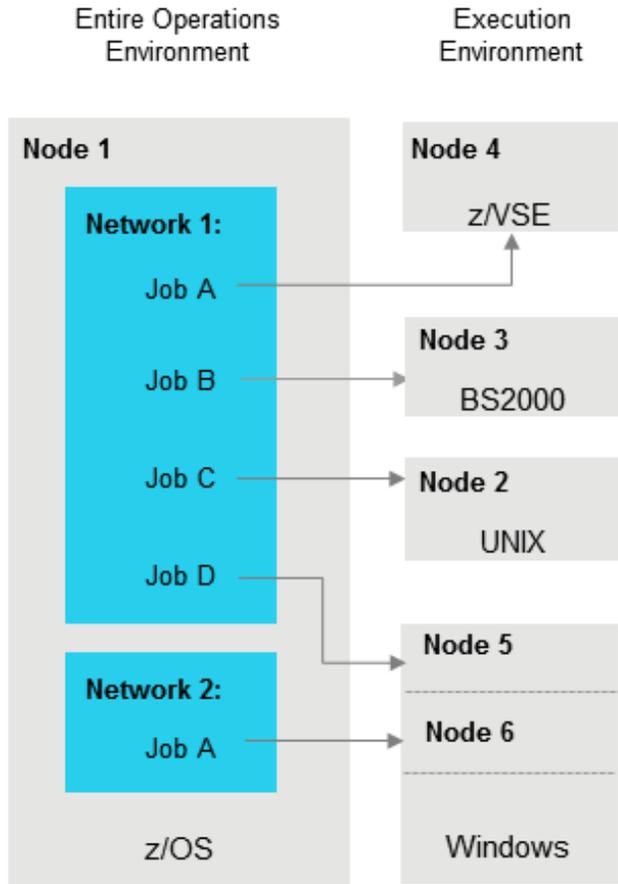
Nodes are Entire System Servers and refer to machines on which requests to the operating system are executed. They are distinguished by numerical identifiers in the same way as database IDs distinguish between different Adabas databases. Within Entire Operations, each machine is assigned a node name. 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
- *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 before activation** option described in *Defaults for Time Ranges* 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.)

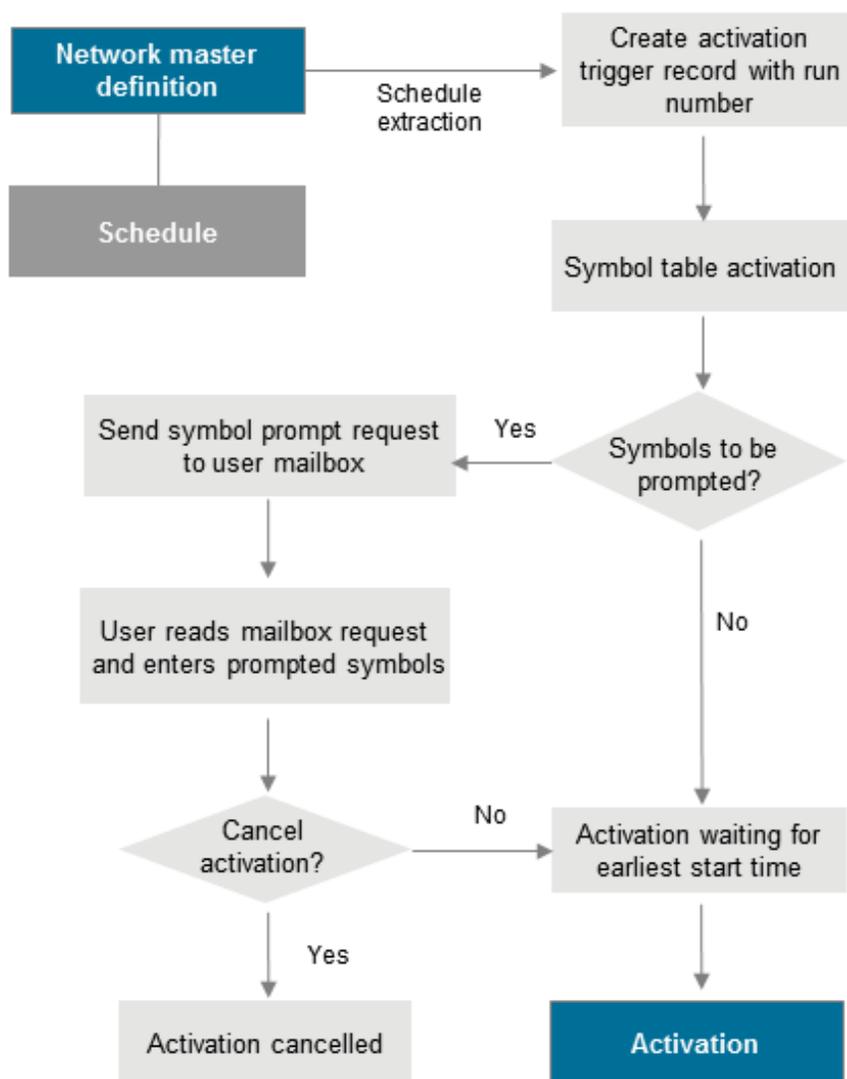
Manual Activation

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 *Defaults for Network Options* 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 internal 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 GUI Client displays status or error messages at the following locations:

Location	Description
In the active window or in an additional window	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 Displaying Logged Information .
Message column of the List Active Jobs window	Contains the last status message or error message for the active job. For more information, see the Message column described in List Active Jobs .
Browse Log window	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 Displaying Logged Information .
Monitor tasks SYSOUT	Contains mainly start and end messages of the Monitor tasks. In this case, some other important events are also logged in addition.

User Language

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

- The nodes of the tree view and the context menu functions in the **object workspace**;
- The field and columns in the open windows in the content pane and in the **result list**.



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:

- [Options Menu](#)
- [System Default and User Profile](#)
- [Natural ULANG Parameter](#)

Options Menu

From the **Options > Language** menu of the main application window, select **English** or **German**.

The language settings are kept for future Entire Operations sessions.

System Default and User Profile

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

- On the **Character interface settings** page of the Entire Operations Defaults window.
- In a user profile on the **Main** page of the **Maintenance/Create new User** window: see *Viewing, Adding and Modifying a User*.

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 output.

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

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

Reporting

The **Reporting** 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 an internal editor.

Before writing the file back, the editor creates a backup copy of the edited file.

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.

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

II

Using Entire Operations GUI Client

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

Online Help

Starting and Ending an Entire Operations Session

Logging on and off an Operating System Server Node

Elements of the Main Application Window

Common and Global Functions

Date and Time Formats

2 Online Help

- Error Messages 46
- Online Technical Information 47

You can access the *Entire Operations GUI Client* documentation online to obtain general usage information or help on a specific function.

The *Entire Operations GUI Client* documentation (including PDF books) is also available at Software AG's Empower web site at <https://empower.softwareag.com/> (Empower login required).

➤ **To invoke online help**

- Select an Entire Operations node and choose **Contents** from the **Help** menu.

The main overview of the *Entire Operations GUI Client* documentation appears.

➤ **To invoke context-sensitive help**

- Choose the **Help** button (if available) in a window or dialog or press F1.

Or:

Select a node in the **object workspace** and press F1.

Help on the current topic appears.

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.

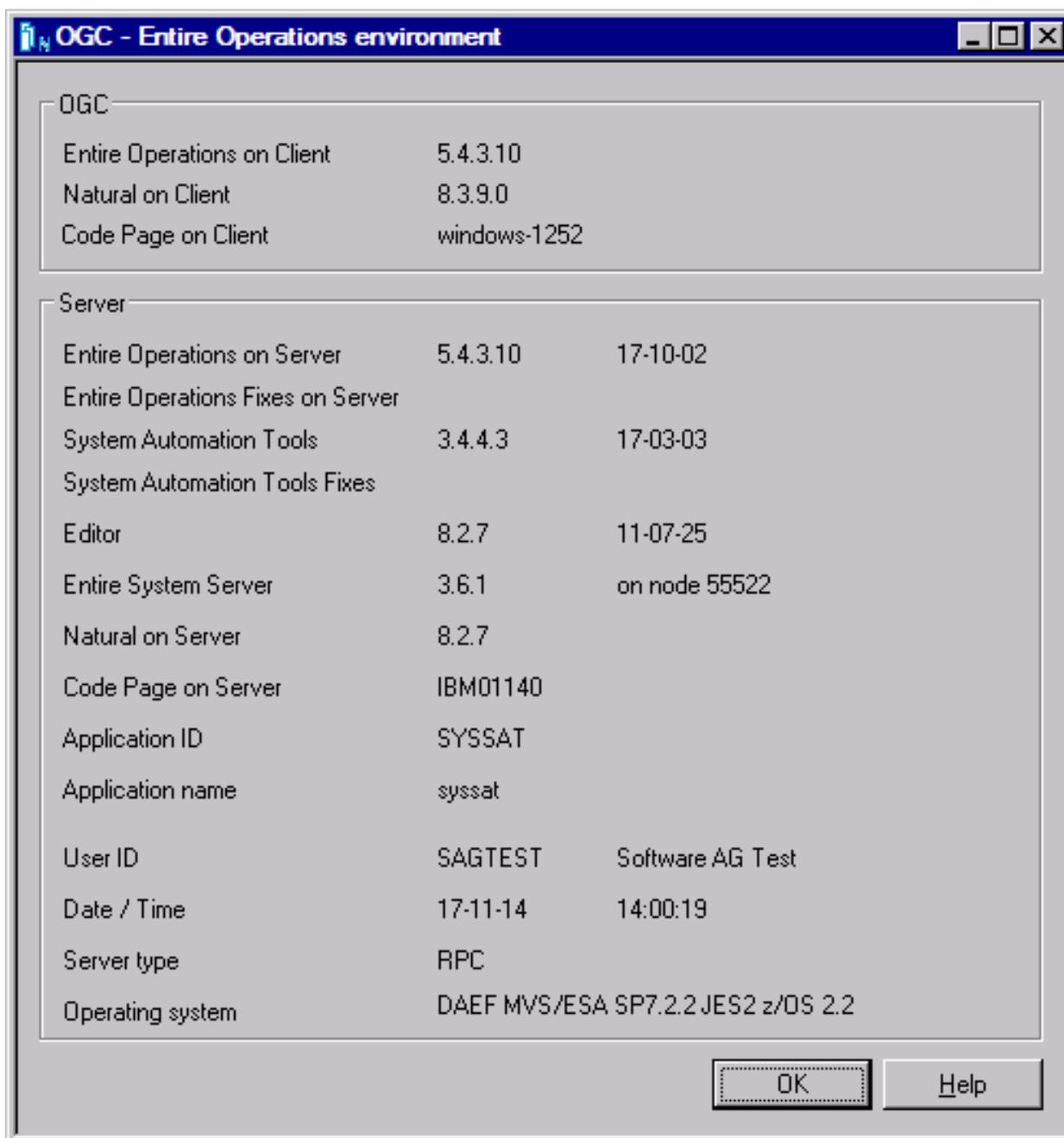
For further information on system messages and the locations where they appear, see the section *System Messages*.

Related Information:

- *Results Window* and *Status Bar*

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:



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

3 Starting and Ending an Entire Operations Session

» To establish an Entire Operations session

- 1 On your desktop, click on the **Entire Systems Management** shortcut icon.

(A desktop icon for Entire Systems Management is generated automatically after the installation procedure.)

A **Software AG ESM Logon** dialog similar to the example below opens:



2 Enter your credentials:

- Natural Security user ID and password. If required, choose **Change password** to change your password (see the next step).
- (optional) EntireX user ID and password.

Select the **Restore My Desktop** check box (selected by default) if you want to restore your current desktop settings. See also *My Desktop Menu* in the *User's Guide*.

3 Choose **OK** when you are finished.

Entire Operations Client is launched if your Natural Security and EntireX (if relevant) user IDs and passwords are accepted. Otherwise, appropriate error messages occur.

If a password has expired, a dialog similar to the example prompts you for a password change:

Change NSC password

Natural Security Logon to Server

User ID: SAGTEST

Old password:

New password:

Confirm new password:

EntireX Communicator Broker Security Authorization

Different User ID for EntireX Broker Security

User ID: BRKSECID

Password:

Default server

Node name: DAEF:4020

Server name: QA82ROP4

OK Cancel

Choose **OK**.

The **Entire Systems Management** main application window (see the *User's Guide*) opens.

Information in the Main Application Window

By default, the current server name and node name are displayed at the top and at the bottom of the main application window.

In addition, you can display text information of your choice at the top or bottom of the object workspace. For this purpose, System Automation Tools provides the user exit `ESUEX02N` in the system library `SYSSAT`. With the user exit, you can specify the text to be displayed, its color and location, and also choose to display this text instead of the server name and node name at the top of the main application window. For details, see the corresponding user exit source object `ZSUEX02N` in the library `SYSSAT`.

4 Logging on and off an Operating System Server Node

- Logon Function 54
- Fields: Node Logon 55
- Monitoring the Node Connection Status 56
- Logoff Function 58

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

Logon Function

➤ To log on a node

- 1 In the **object workspace**, select a node instance and choose **Logon** from the **context menu**, or press CTRL+ALT+L.

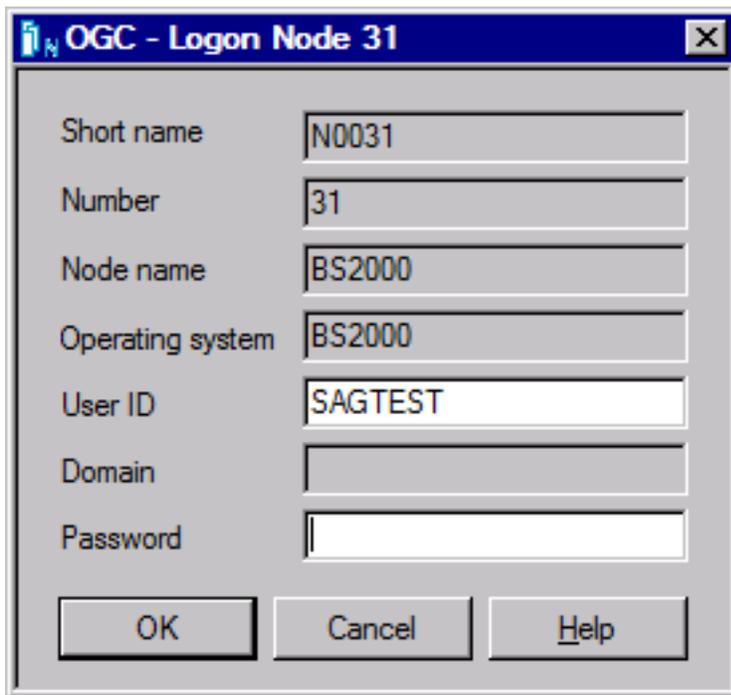
Or:

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

Or:

From the **General** node, choose **Monitor Status** (only applies if the node is not active yet).

A **Logon Node** dialog similar to the example below opens:



Short name	N0031
Number	31
Node name	BS2000
Operating system	BS2000
User ID	SAGTEST
Domain	
Password	

OK Cancel Help

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

- 2 Choose **OK**.

Fields: Node Logon

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

Field	Description
Short name	Short node name.
Number	Node number.
Node name	Logical (long) node name.
Operating system	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. 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). ■ UNIX: If this field is left 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 groups command.
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>

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. In addition, you can add a node to the list of nodes to be monitored.

➤ To monitor the node connection status

- 1 In the **object workspace**, select a **Node** metanode and choose **Show Connection Status** from the context menu.

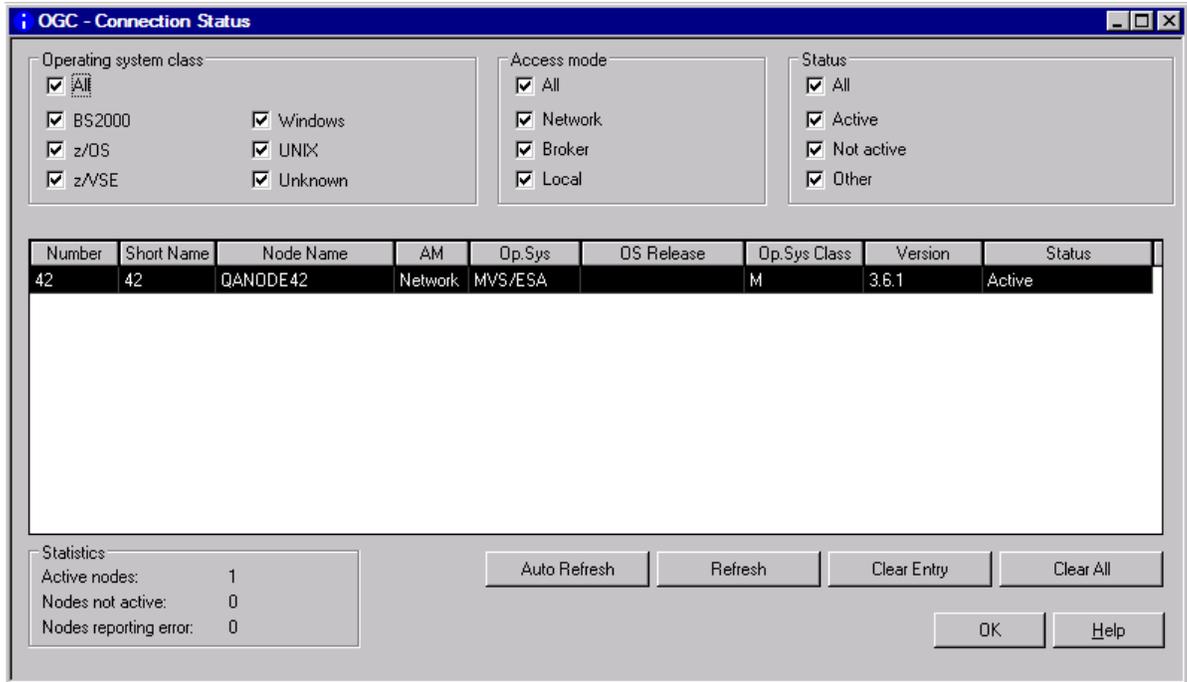
Or:

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

```
STATUS NODES
```

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

A **Connection Status** window similar to the example below opens:



The columns and selection options in the window are explained in *Columns and Selection Options: Node Connection Status*.

- 2 Choose **Clear Entry** to remove the selected node status entry from the list or choose **Clear All** to remove all node status entries.

➤ To add a node to be monitored

- In the **object workspace**, select a **Node** instance and open **Add to Connection Status** from the context menu.

The node is added to the **Connection Status window**.



Note: By default, a node connection status is transient and can change when you terminate an Entire Operations session. You can save the status of all node connections for a future session by selecting the **Node Connection Status** option in the session profile described in the section *User Maintenance* in the *Administration* documentation.

For further information, see also the API NOPUST3N described under *Inquire Network and Job Status, Symbol Table* in the section *API Routines*.

Columns and Selection Options: Node Connection Status

The columns, selection options and output fields contained in the **Connection Status window** are described in the following table.

Column/Section	Description
Operating system class	In this section, check or uncheck single or multiple operating systems (BS2000, z/OS, z/VSE, Windows and UNIX) defined for the nodes or select All for all operating systems. Unknown selects all nodes for which no operating system has been defined.
Access mode	In this section, check or uncheck single or multiple access modes used by the nodes (network for mainframe nodes, EntireX Broker for UNIX and Windows nodes and local node) or select All for all access modes.
Status	Node status of last Entire System Server call, for example, active or not active. In this section, check or uncheck single or multiple status values or select All for all status values. Other selects all nodes with a status other than active or not active.
Number	Node number.
Short Name	Short node name.
Node Name	Logical (long) node name of the server.
AM	Access mode.
Op.Sys.	Operating system on node.
OS Release	Operating system release.
Op.Sys Class	Operating system class.
Version	Entire System Server version.
Statistics	The output fields in this section show the numbers of active and inactive nodes and the nodes where errors occurred.

Logoff Function

» To log off a node

- In the **object workspace**, select the required node instance and choose **Logoff** from the context menu.

Or:

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

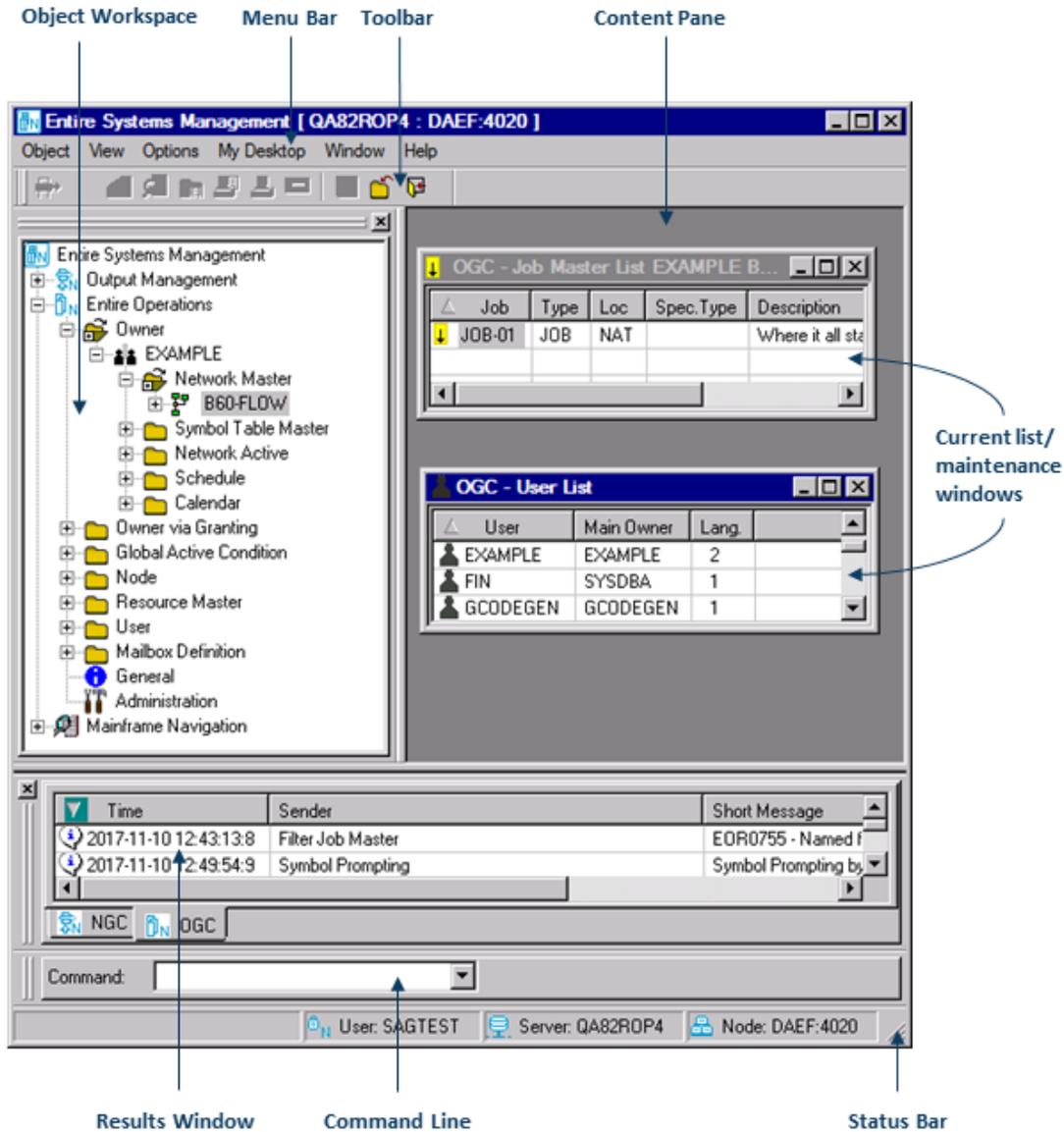
The previously selected node is now logged off.

5

Elements of the Main Application Window

▪ Object Workspace	63
▪ Menu Bar	66
▪ Options Menu	67
▪ My Desktop Menu	69
▪ Toolbar	70
▪ Content Pane	70
▪ Results Window	71
▪ Command Line	71
▪ Status Bar	72
▪ Context Menu	72

When you start the Entire Operations GUI Client, an **Entire Systems Management** main application window similar to the example below opens:



The main application window contains the **object workspace** with the **Entire Operations** tree view on the left and the **content pane** on the right. To modify the size of a pane, move the mouse pointer over the border separating the panes until the pointer changes, showing two arrows pointing in opposite directions $\leftarrow\rightarrow$. Then drag the border using the mouse until the panes have the required size.

In addition to the object workspace and the content pane, the main application window contains the **menu bar**, the **toolbar**, the **command line**, the **result list** and the **status bar**.



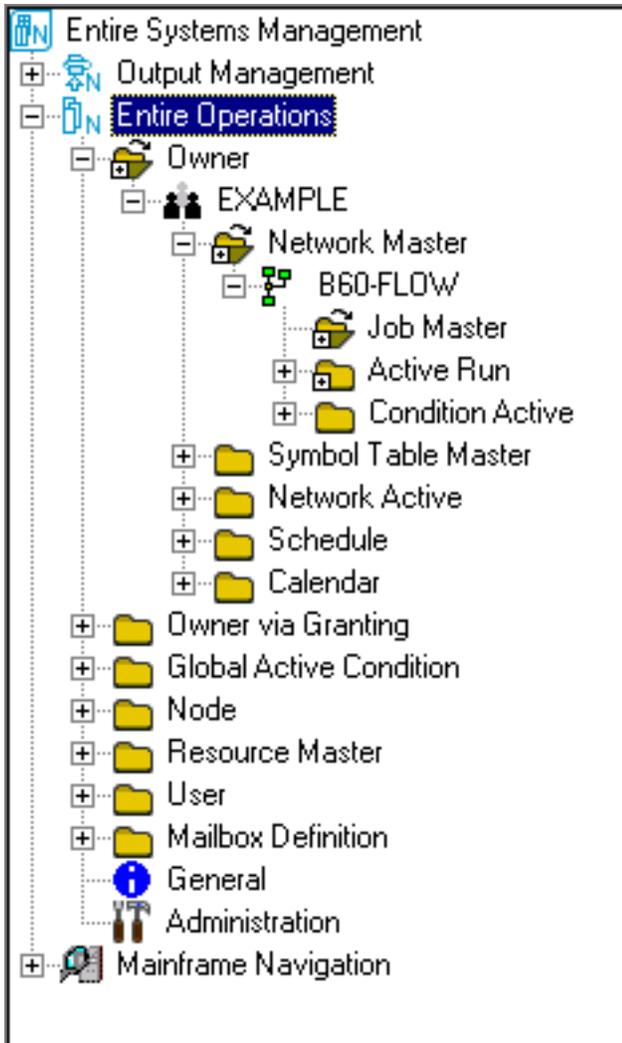
Note: You can hide application nodes that cannot be accessed for different reasons by using the ESUEX01N user exit routine for the System Automation Tools server. For details, see the source object ZSUEX01N (supplied for ESUEX01N) in the Natural SYSSAT system library.

The following topics are covered below:

Object Workspace

- [Expanding or Collapsing a Tree Node](#)
- [Working within the Tree View using Function Keys](#)

The Entire Operations object workspace with the tree view is on the left side of the main application window:



In the object workspace, all available Entire Operation objects are hierarchically listed as the nodes of a tree view. You can create and maintain objects by using the functions provided by the **context menu** of a selected node or subnode.

➤ **To show or hide the workspace**

- In the **View** menu, mark or unmark **Object Workspace** to show (default) or hide the object workspace respectively.

Expanding or Collapsing a Tree Node

You can expand or collapse a tree node to list and access all objects contained in a node or subnode.

» To expand or collapse a tree node

- With the left mouse button, click on the plus or minus sign in front of the node you want to expand or collapse respectively.

An object preceded by a minus sign is fully expanded; an object preceded by a plus sign is fully collapsed.

Or:

Use the **Expanded workspace tree** option of the menu item **My Desktop > Options** described in *My Desktop Menu*.

Working within the Tree View using Function Keys

- [Navigation](#)
- [Executing Object Functions](#)

Navigation

» To navigate within the tree view

- Press UP ARROW or DOWN ARROW to move the selection up or down in the tree view or list view.

Or:

Press LEFT ARROW or RIGHT ARROW to select or collapse an object in the tree view within the same hierarchy.

Or:

Press ENTER or double-click on an object to perform the default function for the currently selected object. It also collapses or selects an object, depending on its current state.

Or:

Press HOME or END to select the first or the last object within the tree view or list view.

Executing Object Functions

➤ To add an object

- 1 Select an object for which the **New** function is available in the **context menu**.
- 2 Press the `INSERT` key to trigger the **New** function.

A **Create new** window opens where you can enter all definitions required to add a new object.

➤ To delete an object

If the **Delete** function exists for the respective object:

- 1 Select an object for which the **Delete** function is available in the context menu.
- 2 Press `DELETE`.

The selected object is deleted.

Menu Bar

The menu bar is at the top of the main application window on the left:



This section covers the following topics:

- [Choosing a Menu](#)

Choosing a Menu

➤ To choose a menu from the menu bar

- 1 Select the appropriate menu.
A drop-down menu of options appears.
- 2 From the drop-down menu, choose the required function and object options.

The following menus are available:

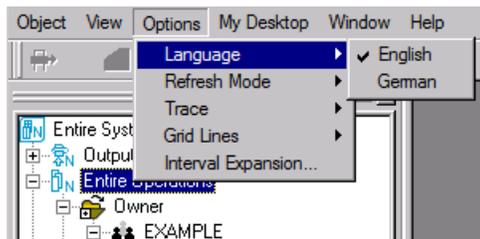
Menu	Function
Object	Filter, list, create, modify, display or perform other functions on the selected objects. Exit the Entire Operations GUI Client.
View	Show or hide the object workspace , command line , result list and/or status bar .
Options	Select either German or English, modify trace options. Select trace level and trace writing mode (continuous or overwrite), set refresh mode (current level or recursive). See <i>Options Menu</i> for detailed information.
My Desktop	Individual desktop settings described in <i>My Desktop Menu</i> .
Window	Select the next, previous or required content pane. Close all panes.
Help	For a selected Entire Operations node: Content: Display the main overview of the Entire Operations GUI Client documentation. See also <i>Online Help</i> . About Entire Operations: Display the current version of Entire Operations.

Options Menu

The **Options** menu provides the following:

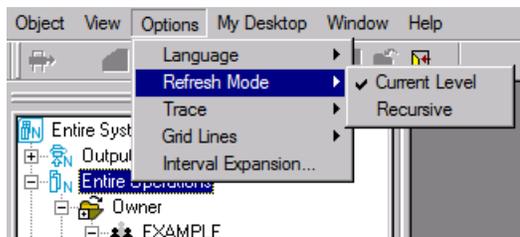
Language

Select the required language:



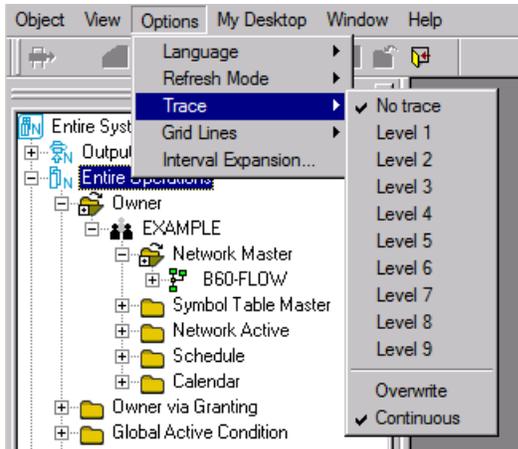
Refresh Mode

Set the refresh mode (current level or recursive):



Trace

Set trace options:



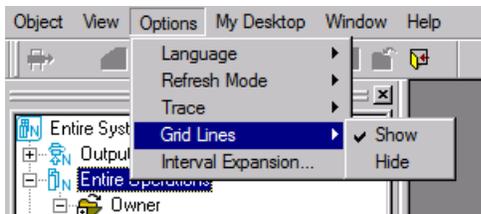
- **No trace** (default): No trace is written to the log file provided that the log file previously used is closed.
- **Level 1 to Level 9**: A trace is written to the log file with different complexity (**1** is low, **9** is high) provided that the log file is open from a previous use.
- **Overwrite**: Single log file with name `%temp%/estrace.log`. Each time the log is newly started, the file will be overwritten.
- **Continuous** (default): Multiple log files with name `%temp%/estrace_date_time.log`. Each time the log is newly started, a new file is generated, where *date* and *time* will be replaced with this specific date and time, for example, `estrace_161115_082545.log`. See also [Date and Time Formats](#).

Writing of the specific log file will be started if:

1. The trace level is set from **No trace** to a level of 1 to 9.
2. During startup if the last session is closed with active trace level 1 to 9.

Grid Lines

Display or hide grid lines for list views.



See also [Showing or Hiding Grid Lines](#).

Interval Expansion

Applies to Output Management only.

My Desktop Menu

The options provided by the **My Desktop** menu are used to customize or reset your Entire Systems Management environment. You can save the tree nodes and/or dialogs opened during the current session and the filters set to use them at the startup of a future session.

The following states can be saved:

- Opened dialogs and views;
- Expanded tree view of the **object workspace**;
- Defined filters.

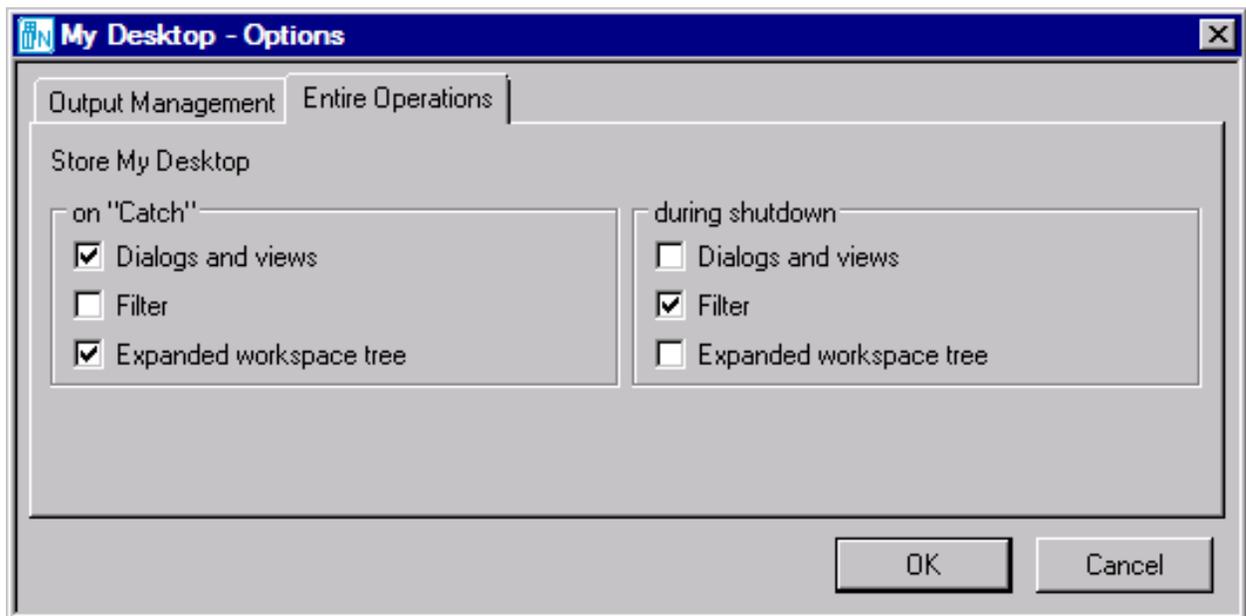
My Desktop Save/Restore

You can save the **My Desktop** configuration in two ways:

1. Manually with the menu item **My Desktop > Catch**, or
2. Automatically on Entire Systems Management shutdown.

You can choose the save options with the menu item **My Desktop > Options**.

You can select only one option. Selecting both options (**on Catch** and **during shutdown**) simultaneously is not possible. The settings are stored for each user separately in the user profile.



You can manually reset the complete **My Desktop** configuration or parts of with the menu item **My Desktop > Reset**. The following options are available:

- All
- Dialogs and views
- Filter
- Expanded workspace tree

The menu items **My Desktop > Catch** and **My Desktop > Reset** are only enabled when at least one option of **My Desktop** is saved in **Catch** mode.

It is possible to temporarily disable the **My Desktop** functionality on Entire Systems Management startup. This can be useful, when for a current session **My Desktop** functionality is not required or in case of a **My Desktop** startup problem. You can store your setup in the user profile for the next sessions by setting the **Restore My Desktop** option in the **Software AG ESM Logon** dialog (see *Starting Entire Operations GUI Client*).

Toolbar

The toolbar is just below the menu bar at the top of the screen.



➤ To perform a function from the toolbar

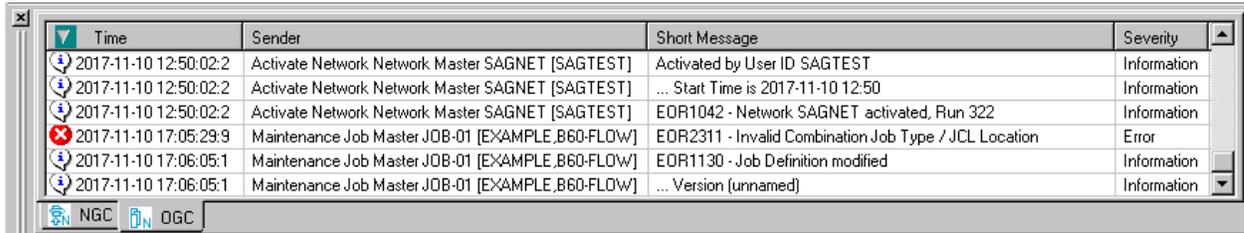
- Choose the appropriate function icon. If an icon is greyed-out, the function is not available for the selected object.

Content Pane

The content pane takes up the right side of the main application window. The appropriate list, window, diagram or dialog appears here, when you perform a function for an object.

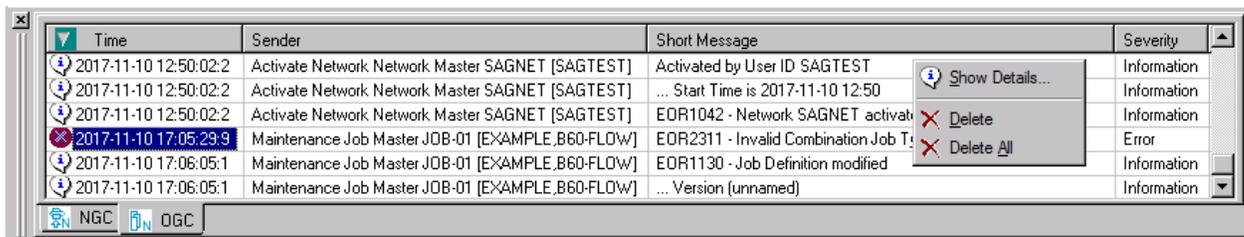
Results Window

The **Results** window (shown by default) contains a list of messages that report hints, errors and warnings that occur during the session as a result of function processing.



Time	Sender	Short Message	Severity
2017-11-10 12:50:02.2	Activate Network Network Master SAGNET [SAGTEST]	Activated by User ID SAGTEST	Information
2017-11-10 12:50:02.2	Activate Network Network Master SAGNET [SAGTEST]	... Start Time is 2017-11-10 12:50	Information
2017-11-10 12:50:02.2	Activate Network Network Master SAGNET [SAGTEST]	EOR1042 - Network SAGNET activated, Run 322	Information
2017-11-10 17:05:29.9	Maintenance Job Master JOB-01 [EXAMPLE,B60-FLOW]	EOR2311 - Invalid Combination Job Type / JCL Location	Error
2017-11-10 17:06:05.1	Maintenance Job Master JOB-01 [EXAMPLE,B60-FLOW]	EOR1130 - Job Definition modified	Information
2017-11-10 17:06:05.1	Maintenance Job Master JOB-01 [EXAMPLE,B60-FLOW]	... Version (unnamed)	Information

A **context menu** is available where you can get detailed information for a specific line and can delete one or all messages in the result list.



Time	Sender	Short Message	Severity
2017-11-10 12:50:02.2	Activate Network Network Master SAGNET [SAGTEST]	Activated by User ID SAGTEST	Information
2017-11-10 12:50:02.2	Activate Network Network Master SAGNET [SAGTEST]	... Start Time is 2017-11-10 12:50	Information
2017-11-10 12:50:02.2	Activate Network Network Master SAGNET [SAGTEST]	EOR1042 - Network SAGNET activat	Information
2017-11-10 17:05:29.9	Maintenance Job Master JOB-01 [EXAMPLE,B60-FLOW]	EOR2311 - Invalid Combination Job T	Error
2017-11-10 17:06:05.1	Maintenance Job Master JOB-01 [EXAMPLE,B60-FLOW]	EOR1130 - Job Definition modified	Information
2017-11-10 17:06:05.1	Maintenance Job Master JOB-01 [EXAMPLE,B60-FLOW]	... Version (unnamed)	Information

Click on the **Close** button or use the **View** menu option to close the **Results** window.

Command Line

The command line is the **Command** input field shown above the **status bar**:



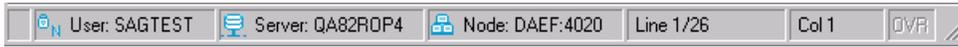
In the command line, you can enter an Entire Operations direct command if you want to directly execute an Entire Operations function.

From the drop-down list box, you can also select a direct command you executed previously during the current session.

For a description of all Entire Operations direct commands available and instructions on issuing a direct command, refer to the *Direct Commands* documentation.

Status Bar

The status bar is the horizontal information line at the bottom of the main application window:



It can contain the following information:

- The current user, server and node
- The total number of data records listed in an active window, for example: 7 data records received from server.

Or:

For an active editor window (example above): the current cursor position (line and column number), the total number of source lines and the edit mode (if OVR is activated, input overwrites existing text).

- For a selected item in a context menu: a brief description of the menu item, for example, the **Filter** menu item shows `Store filter criteria for a list` in the status bar.

Context Menu

A context menu allows you to perform a function on an object listed in the tree view of the **object workspace** or in the list or diagram view (see *Viewing and Maintaining a Job Network Diagram*) of an active window in the **content pane**.

In addition, a context menu can also be available for a name selected in the input field of an active maintenance window, for example, the name of a node, a file or a symbol table.

➤ To choose a function from the context menu

- 1 Select an object from the object workspace, open a list or diagram window, or select a name input field in a maintenance window.
- 2 Click the right mouse button.

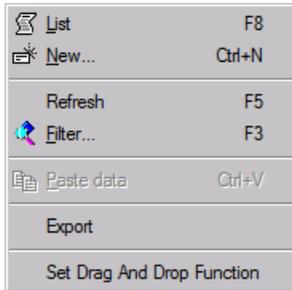
Or:

Press `SHIFT+F10`.

Or:

Press MENU (application key).

The context menu opens.



The example above shows the context menu for the **Network Master** metanode selected in the object workspace.

- 3 Choose the required function.

Functions that are greyed-out are not available for the selected object.

6 Common and Global Functions

▪ Canceling Reading	110
▪ Applying Changes	76
▪ Listing Objects	76
▪ Refreshing Object Lists	77
▪ Filtering Objects	78
▪ Saving a List View as a Report File	80
▪ Drag & Drop	80
▪ Displaying Objects	82
▪ Copying Objects	82
▪ Pasting Objects	83
▪ Deleting Objects	84
▪ Importing and Exporting Objects	84
▪ Functions of the Metanode General	84
▪ Monitor Status	85
▪ Show Messages	86
▪ Add to Workplan	87
▪ Show Workplan	87
▪ Show all Owners	91
▪ Show linked Owners	91

This section describes common functions of Entire Operations GUI Client that perform the same operations for different objects.

In addition, it describes the functions of the **General** metanode that perform global operations on all objects in your Entire Operations environment.

Canceling Reading

In Entire Operations GUI Client, a long reading process can be canceled by pressing ESC.

» To cancel reading

- Press ESC.

Applying Changes

All maintenance dialogs within the Entire Operations GUI Client provide an additional **Apply** button to save previously entered information without closing the maintenance dialog. The changes take effect just by selecting **Apply**.

If you create a new object and select **Apply**, this dialog changes from a **New** to a **Maintenance** behavior.

Listing Objects

» To view a list of objects

- 1 In the **object workspace**, select a metanode.
- 2 Open the **context menu** and choose **List**, or press F8.

Or:

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

A **List** window similar to the example below opens:

Job	Type	Loc	Spec.Type	Description	File or Library	Member
JOB-01	JOB	NAT		Where it all starts	SYSEXPB	AAREADME
JOB-012	JOB	MAC		Depending on Job-01	SYSEORU	B60-M01
JOB-013	JOB	MAC		Depending on JOB-012	SYSEORU	B60-M01
JOB-014	JOB	MAC		Depending on JOB-013	SYSEORU	B60-M01
JOB-015	DUM			Depending on JOB-014		
JOB-019	JOB	MAC	G	Depending on JOB-01	SYSEORU	B60-M01
JOB-02	JOB	MAC		Dep. JOB-15, JOB-19	SYSEORU	B60-M02
JOB-03	NAT	NAT		Depending on JOB-02	SYSEORU	B60-P01
JOB-04	JOB	MAC		Depending on JOB-03	SYSEORU	B60-M01
JOB-05	DUM			Depending on JOB-04		
JOB-06	JOB	MAC		Where it all ends	SYSEORU	B60-M02
JOB-1-TEST	JOB	NAT	D	Where it all starts	SYSEXPB	ACCEPX01

All objects of the selected metanode are listed.

In the example above, all jobs of the **Job Master** metanode are listed for the B60 - FLOW network of the owner EXAMPLE.

This section covers the following topics:

- [Showing or Hiding Grid Lines](#)

Showing or Hiding Grid Lines

The **context menu** in a **List** window provides the **Grid Lines** function with the options **Show** and **Hide** to show grid lines in the **List** window or hide them, respectively. Changes affect the view in the active **List** window only.

You can set initial values for all **List** views:

From the **Options** menu, choose **Grid Lines** and select **Show** or **Hide**. This setting is stored between sessions in user XML profile and is used as the default value for all **List** views.

Refreshing Object Lists

➤ To refresh objects listed in the object workspace

- In the **object workspace**, select a metanode and choose **Refresh** from the context menu, or press F5.

The list of objects of the selected metanode is updated immediately.

➤ **To refresh objects from an open list window**

- Open a window that contains a list of objects and press F5 or choose the **Refresh** button, if available.

The list is updated immediately.

Or:

Open a window that contains a list of objects and choose **Auto Refresh**, if available.

In the **Automatic Refresh** window that opens, select the **Automatic refresh** check box. If required, change the time interval (the default is 30 seconds) in which you want to update the list.

Choose **OK**.

The list is updated automatically in the time interval specified.

Filtering Objects

You can define a filter for the objects to be shown by default in the tree view nodes of the **object workspace** and listed in an object **List** window.

If a filter is defined for a set of objects, the following applies:

- The default values set in the **Filter** option of the user profile (see the *Administration* documentation) are ignored.
- The icon of the folder that contains the filtered objects changes from  to  in the object workspace.

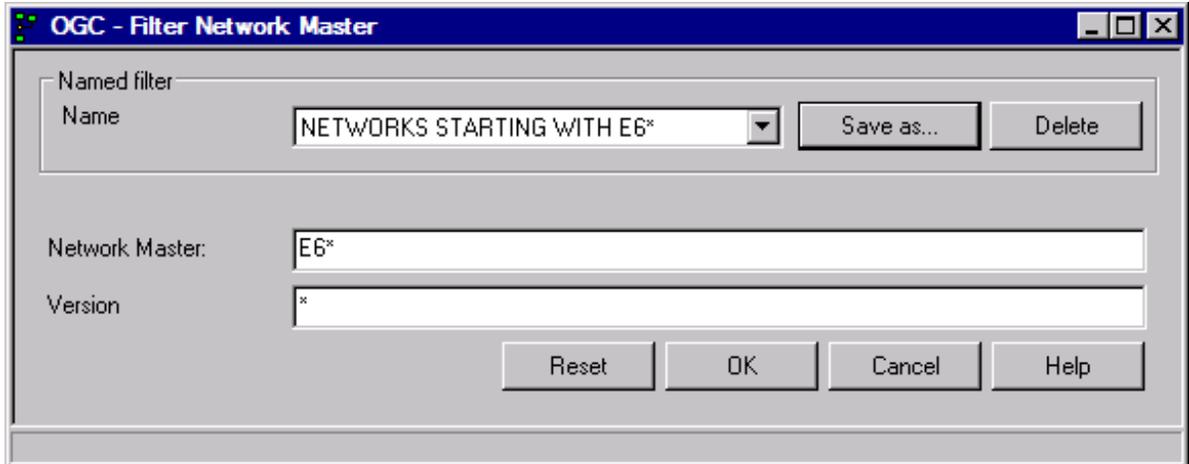
➤ **To filter objects**

- 1 In the **object workspace**, select a metanode.
- 2 Open the **context menu** and choose **Filter**.

Or:

Press F3.

A **Filter** window similar to the example below opens:



The input fields provided in the window depend on the type of objects contained in the selected node. For example, the field **Version** is only available for objects of the type network master and symbol table master.

- 3 You can define a filter or choose a predefined filter from a template (common named filter). Choose either option:

Leave the **Named filter Name** field blank and enter the required filter criteria in the remaining input fields.

You can use an asterisk (*) and a question mark (?) as wildcards at any position in the identifier string.

The question mark (?) indicates a single position not to be examined.

The asterisk (*) indicates any number of positions (zero or more characters) not to be examined.

In the example above, all networks with names starting with E6 are to be selected for all network versions.

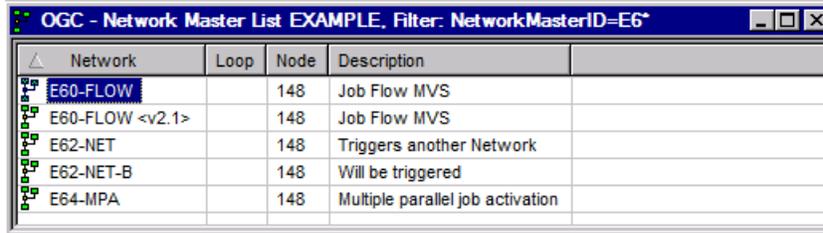
Or:

From the **Named filter Name**, select the name of a defined filter with predefined filter criteria related to this object type.

You must have appropriate access rights to read, modify and delete common named filters. See also the default **Filter** settings for user profiles described in the *Administration* documentation.

- 4 Choose **OK**.

Now, only objects which satisfy the selection criteria are listed in the tree view and **List** window as shown in the following example:



Network	Loop	Node	Description
E60-FLOW		148	Job Flow MVS
E60-FLOW <v2.1>		148	Job Flow MVS
E62-NET		148	Triggers another Network
E62-NET-B		148	Will be triggered
E64-MPA		148	Multiple parallel job activation

The filter criteria used for the list is indicated in the window title (here: **Filter: NetworkMasterID=E6***).

A filter remains active until you explicitly **Reset** it.

5 The **Filter** window provides the following functions:

- Choose **Reset** to reset all filter criteria and deactivate a filter.
- Choose **Save as...** to save the current filter criteria as a named filter.

A window opens where you can enter the filter name.

Choose **OK** to save the new filter.

- Choose **Delete** to remove a selected named filter.

A window prompts you to confirm the deletion.

Saving a List View as a Report File

The **context menu** in a **List** window provides the **Save View as File** function to save the content of a **List** window as a report file containing columns and column headings and filter information. For detailed information, see *Report Output Options* in the section *Reporting*.

Drag & Drop

The drag and drop function is used to specify and perform a preferred menu function on all objects in a node or a node instance contained in the **object workspace**.

This can help perform frequently used functions more easily and avoids having to search through context menus.

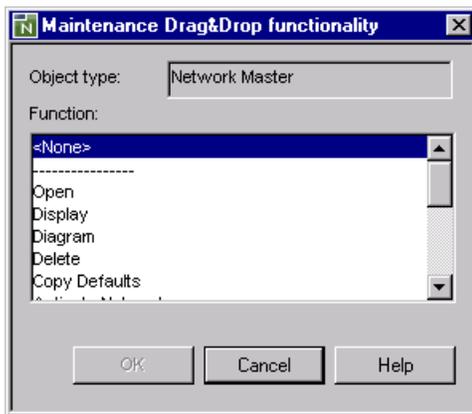
➤ **To execute a function with drag and drop**

- 1 From the tree view in the **object workspace**, drag a node or a node instance and drop it onto the empty content pane.

Or:

Select the required node and choose **Set Drag and Drop Function** from the context menu.

If the selected node is not yet preset with a drag and drop function, a **Maintenance Drag&Drop** dialog similar to the example below opens:



The functions available for selection depend on the object type (here: Network Master) of the node selected. There are different functions available for node instances and metanodes.

Select the required function and choose **OK**.

The dialog closes.

- 2 If the selected node is already preset with a drag and drop function, the selected function is executed immediately.

If a drag and drop function cannot be performed, the function terminates with an appropriate message.

➤ **To change or reset a drag and drop function**

- 1 Open the **Maintenance Drag&Drop dialog** as described in the previous instructions.
- 2 Select the required function or select **<None>** to remove a drag and drop function and reset the function.
- 3 Choose **OK**.

Displaying Objects

The **Display** function opens a window that displays information for the currently selected item. You can only view the information, not edit it. If you want to edit the item, you have to use the **Open** function.

➤ To display an object

- 1 In the **object workspace**, select the object.
- 2 Open the **context menu** and choose **Display**.

Or:

Press CTRL+D.

A dialog appears that displays all available information for the selected object.



Note: For a detailed description of the displayed fields, see the description of the **Open** function for the respective object.

Copying Objects

 **Important:** Copying an object always means that the complete underlying structure is also copied into the target destination.

➤ To copy single or multiple objects

- 1 In the **object workspace**, select a node instance or an entire node.

Open the **context menu** and choose **Copy data**, or press CTRL+C.

- 2 Or:

In the **object workspace**, select a node for which you can choose **List** from the context menu. A **Master List** window opens.

From the list table in the **Master List** window, select single or multiple objects and choose **Copy data** from the **context menu** within the table, or press CTRL+C.

The object data is copied to the clipboard and ready to paste at the required location within the object workspace.

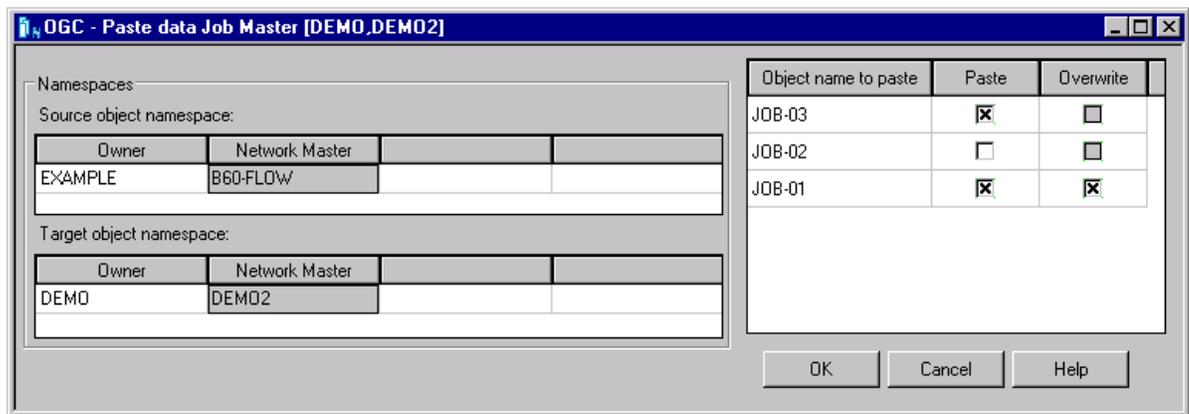
- 3 Proceed as described in *Pasting Objects*.

Pasting Objects

> To paste objects

- 1 In the **object workspace**, select the node where you want to place the object data **previously copied** to the clipboard.
- 2 Choose **Paste data** from the context menu, or press CTRL+V.

A **Paste data** window similar to the example below opens:



The **Source object namespace** table indicates the source location (here: owner EXAMPLE, network B60-FLOW) that contains the objects to be copied.

The **Target object namespace** table indicates the target location (here: owner DEMO, network DEMO2) where to copy the selected objects.

The **Object name to paste** column in the right table lists the source objects selected for the copy operation (here: jobs JOB-03, JOB-02, JOB-01).

- 3 In the **Object name to paste** field, replace the object name, if required.

Select the **Paste** check box next to the object you want to copy. The check boxes are selected by default. Remove the check mark for an object you want to skip. In the example above, JOB-02 is not copied.

Select the **Overwrite** check box to replace an existing target object with the same name.

- 4 When you are finished, choose **OK**.

The objects are added to the specified target location and listed in the previously selected node in the object workspace.

Deleting Objects

The **Delete** function opens a dialog to confirm the selection for this operation.

➤ To delete an object

- 1 In the **object workspace**, select the object.
- 2 Open the **context menu** and choose **Delete**.

Or:

Press DELETE.

A dialog appears where you have to confirm the deletion.

Importing and Exporting Objects

You can import and export objects using the **Import/Export** function of the **General** metanode or the **Export** function of a selected node or node instance.

For detailed information on using the functions, see *Importing Objects* and *Exporting Objects* in the *Import/Export Functions* documentation.

Functions of the Metanode General

The **General** metanode provides menu functions that are performed on all objects available in your current Entire Operations environment:

Function	Described in Section
Next Activations	Viewing next network activations. See Viewing and Modifying Next Network Activations in the section <i>Schedule Maintenance</i> .
List Active Jobs	See Listing Active Jobs in the section <i>Maintaining Active Jobs</i> .
Activity Log	See Monitoring Entire Operations Activities in the section <i>Log Information</i> .
Browse Log	See Displaying Logged Information - Browse Log Function in the section <i>Log Information</i> .
Monitor Status	See Monitor Status .
Determination Date	See Determination Date for Report Data in the section <i>Reporting</i> .

Function	Described in Section
Reporting	See <i>Report Types</i> in the section <i>Reporting</i> .
Cross References	See <i>Types of Cross-References</i> in the section <i>Cross References</i> .
Import/Export	<i>Importing Objects</i> and <i>Exporting Objects</i> in the <i>Import/Export Functions</i> documentation.
Show Messages	See <i>Show Messages</i> .
Show Workplan	See <i>Show Workplan</i> .
Show all Owners	See <i>Show all Owners</i>
Show linked Owners	See <i>Show linked Owners</i> .
Set Drag And Drop Function	See <i>Drag & Drop</i> .

Monitor Status

The Monitor is the heart of Entire Operations. The Monitor is a complex program that wakes up periodically and checks the definitions stored in the master database for any work to do. It activates and processes networks and jobs according to their prerequisites and controls running job networks, even if they reside on the nodes of different computers.

The **Monitor Status** function of the **General** metanode is used, for example, to start and stop the Monitor.

Before you can monitor Entire Operations activities, you may have to log on a node.

› To monitor Entire Operations

- 1 In the **object workspace**, select the **General** metanode.
- 2 Open the **context menu** and choose **Monitor Status**.

For an inactive node, a **Logon Node dialog** opens first.

The fields contained in this dialog are explained in *Fields: Node Logon*.

- 3 Enter your user ID and password and choose **OK** to log on to the specified node.

A **Monitor Status** window opens displaying all Monitor tasks.

Here you can:

- Start and stop the Monitor.
- Hold or release a certain Monitor task.
- If you want to display the Monitor tasks before the dialog is automatically refreshed, you can refresh it manually by selecting the **Refresh** button.

- Select **OK**.

For detailed information, see *Status of the Entire Operations Monitor* in the *Administration* documentation.

Show Messages

The **Show Messages** function of the **General** metanode is used to list mailbox messages and requests as shown in the following example:

Time	Mailbox	Owner	Network	Run	Job	Description
2016-02-25 09:01:07	SYSDBA	EXAMPLE	SAGNET	8	J-OGCDEMO	BATCH
2016-02-25 09:01:07	SYSDBA	EXAMPLE	SAGNET	8	J-OGCDEMO	BATCH
2016-02-25 09:01:07	SYSDBA	EXAMPLE	SAGNET	9	J-OGCDEMO	BATCH
2016-02-25 09:01:07	SYSDBA	EXAMPLE	SAGNET	9	J-OGCDEMO	BATCH
2016-02-25 09:01:18	SYSDBA	INCIDENT	I5095089A	944		Network not terminated - no deactivation
2016-02-25 09:01:18	SYSDBA	INCIDENT	I5095089A	945		Network not terminated - no deactivation
2016-02-25 09:01:18	SYSDBA	INCIDENT	I5095089A	946		Network not terminated - no deactivation
2016-02-25 09:02:55	SYSDBA	EXAMPLE	SAGNET	10	J-OGCDEMO	BATCH
2016-02-25 09:02:55	SYSDBA	EXAMPLE	SAGNET	10	J-OGCDEMO	BATCH
2016-02-25 09:57:57	SYSDBA					Symbol Prompting SAGNET-P Run 1 (in 11 : 30 hours)

All messages and requests of all mailboxes to which the current Entire Operations user (**Owner at Logon**) is linked are listed in the table.

The table view is filled dynamically with data if scrolled to the end of the table.

The columns, fields and functions provided in the window are described in *Fields and Columns: Messages and Requests* and *Available Functions: Show Messages and Requests* in the section *Working with Mailboxes*.

Add to Workplan

The **Add to Workplan** function stores a reference to the currently selected object in a list of activities to be done.

» To add an object to the workplan

- 1 In the **object workspace**, select an object.
- 2 Open the **context menu** and choose **Add to Workplan**.

The object is listed within the workplan for further processing.

You can view all entries of your workplan by using the **Show Workplan** menu function of the **General** metanode.

Show Workplan

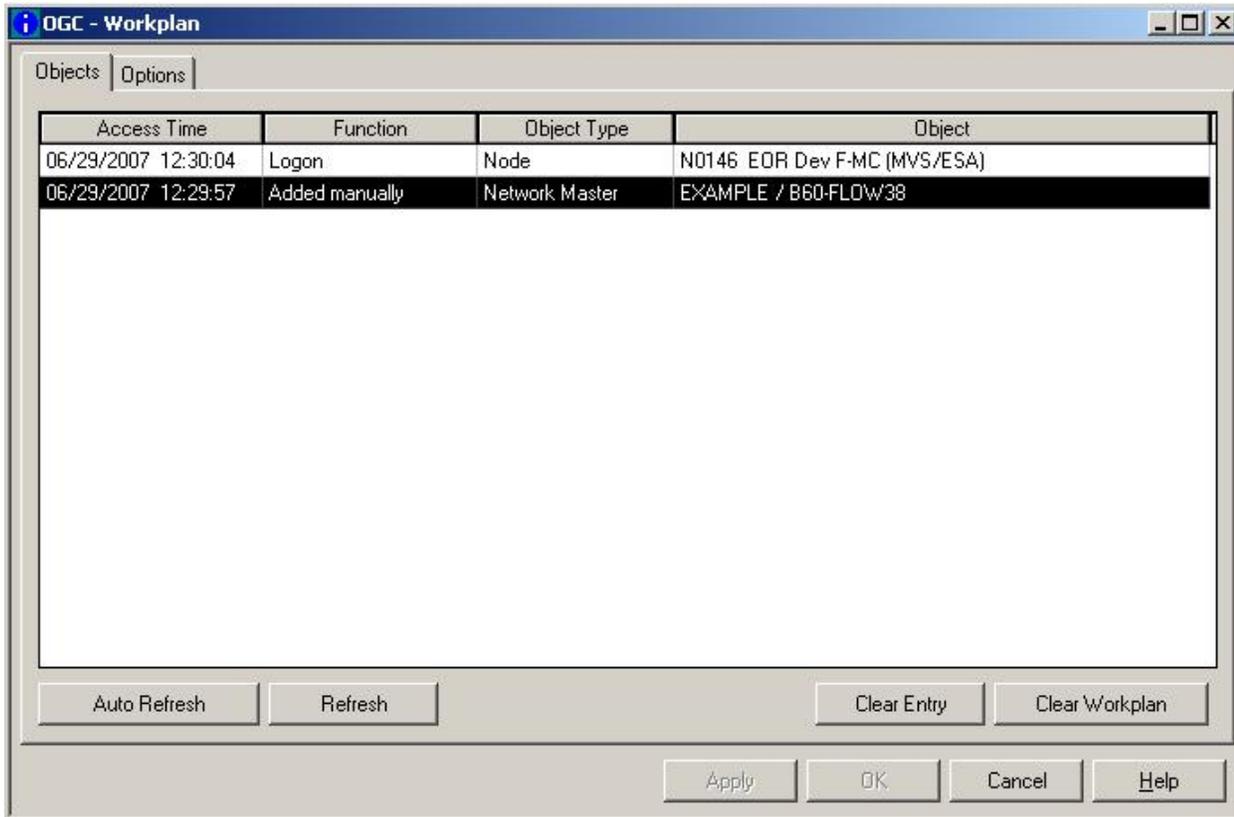
The **Show Workplan** function of the **General** metanode is used to open the workplan.

The workplan contains a list of Entire Operations objects that are accessed during the current Entire Operations GUI Client user session. This list facilitates repeated access to these objects.

- [Objects Tab](#)
- [Options Tab](#)

Objects Tab

Invoking the workplan, a **Workplan** window with an open **Objects** page is displayed first.



The list of objects contains the following columns:

Column	Description
Access Time	Date and time of the last user access to this object. See also <i>Date and Time Formats</i> .
Function	Name of the last function executed for this object.
Object Type	Type of the object.
Object	Name of the object including the superordinate objects that identify it unambiguously within a namespace. The single constituents of the name are separated by slash (/) characters.

> **To refresh the workplan**

The workplan is automatically refreshed for local operating actions of users. Otherwise:

- Choose the **Refresh** button.

The workplan is now manually refreshed.

➤ To add objects to the workplan

Certain objects are inserted into the workplan during an Entire Operations session automatically if you display or change them. Otherwise:

- Choose **Add to Workplan** from the **context menu** for the required object to add it to the workplan manually.

The workplan is automatically refreshed by inserting new objects. Adding a new object to the workplan has no impact on the object itself.

➤ To clear entries in the workplan

- Choose the **Clear Entry** button to delete the selected entry from the workplan.

Or:

Choose the **Clear Workplan** button to delete all workplan entries.

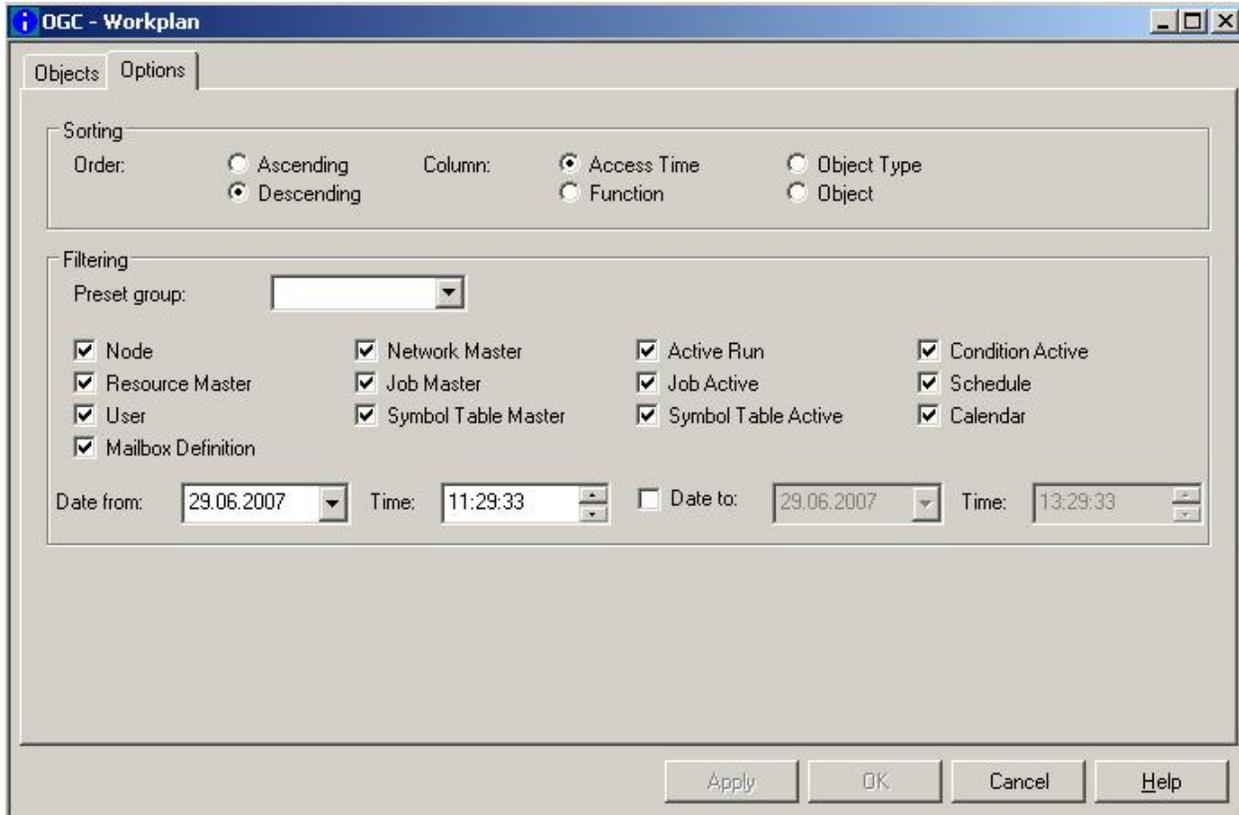
Deleting object entries from the workplan does not delete or deactivate any objects.

➤ To work with objects via the workplan

- On the **Objects** page of the **Workplan** window, select the object you want to process and right-click in the window to open the **context menu** and select the required function.

Options Tab

The **Options** tab displays all possible workplan options.



The workplan is transient by default. However, you can make the workplan persistent between sessions by setting the **Show workplan after activation** option in your user profile as described in **GUI General** in the *Administration* documentation.

The number of recent entries saved can be configured in the user profile (see the *Session Profile* in the *Administration* documentation). If the workplan lists more than the configured maximum number of saved entries, only the last recent records will be saved for the next session.

Filtering: Preset Group

This selection box allows a preselection for the following individual filters.

All

Display all entries.

Master

Display only master objects

Active

Display only active objects

Custom

The object types become determined via marking them in this dialog. It is automatically activated if you diverge from the standard preselection.

None

Deletes all subsequent individual filters.

Show all Owners

This function of the **General** metanode is allowed for users with administrator rights only. It allows you to switch to the SYSDBA mode:

- All existing owners are listed in the **Owner** metanode.
- The **Owner** metanode changes its name to **Owner (SYSDBA access)**.
- No owners are listed below **Owner via Granting**.

This function is allowed for users with administrator rights only. This function corresponds to the direct command `SET OWNERLIST ALL` described in the *Direct Commands* documentation.

See also the section [Using Owners](#).

Show linked Owners

This function of the **General** metanode is allowed for users with administrator rights only.

In opposite to the **Show all Owners** function, this function allows you to switch back from SYSDBA access mode to normal mode:

- Only linked owners are listed in the **Owner** metanode.
- Granted owners are listed in the **Owner via Granting** metanode.

This function corresponds to the direct command `SET OWNERLIST LINKED` described in the *Direct Commands* documentation.

See also the section [Owner Maintenance](#).

7 Date and Time Formats

- Date Display and Input Options 94
- Time Display and Input Options 95

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 date displayed in a window is represented in the Entire Operations default format selected in the **Date format** field of the *Entire Operations Defaults* (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 in the window), 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/Selection Option	Examples
DD.MM.YY	G	30.12.16
	German DD.MM.YY	301216
	German format.	30.12
		3012
DD/MM/YY	E	30/12/16
	European DD/MM/YY European format.	301216
MM/DD/YY	A	12/30/16
	American MM/DD/YY American format.	123016
YY-MM-DD	I	16-12-30
	International YY-MM-DD	161230
	International format (default).	1612

Date Pattern	Date Format Setting/Selection Option	Examples
YYYYMMDD	8	20161230
	8 digits YYYYMMDD	2016
	Eight-digit format.	

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

8 Entire Operations Utilities

▪ Introduction	100
▪ Extract Log Data to Log Selection File	101
▪ Delete Old Data in Log Selection File	102
▪ Output of Log Data to a File	103
▪ Print Account Information from Entire Operations Log	103
▪ Mass Change of Network / Symbol Table Version	104
▪ Monitor or Task Wait Time Modification	105
▪ Monitor Shutdown	105
▪ Monitor Start	106
▪ BS2000 Jobs	106
▪ List or Delete TO-ACTIVATE Command Records	107
▪ Check the Existence of Symbol Table Definitions	108
▪ Mass Change of the Owner / Delete Owner with all Objects	108
▪ Mass Change of Access Rights Granted to Networks	111
▪ User ID Mass Update in Network and Job Definitions	113
▪ Mass Change of Node Numbers	114
▪ Mass Logon Processing in Batch Mode	114
▪ Bulk Execution of MACRO Commands	116
▪ Migration of Log Data from NOP Versions lower than 411 to NOP 412 and above	118
▪ Data Migration to the current Entire Operations Version	120

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 Entire Operations in Batch Mode

9 Using Entire Operations in Batch Mode

- Files Needed to Start Entire Operations in Batch 126
- Files Needed to Start the Batch Command Client 131
- Command Syntax for the Batch Command Client 133
- Commands for the Batch Command Client 138

This section describes the files and commands required to perform Entire Operations functions in batch mode from a Windows environment.



Note: For explanations of the syntax symbols used in this chapter, see the section *Direct Command Syntax*.

Files Needed to Start Entire Operations in Batch

The following files are required to start Entire Operations in batch mode:

1. A windows batch file (.bat)
2. A command file (.cmd)
3. A file with input data (.inp)

Examples of such files are contained in the Natural directory SYSSATGF/RES: exportapi.bat, exportapi.cmd and exportapi.inp.

- [exportapi.bat](#)
- [exportapi.cmd](#)
- [exportapi.inp](#)
- [EXPORT DIAGRAM Command Syntax](#)
- [Files Generated from EXPORT DIAGRAM Command](#)
- [Starting Entire Systems Management in Batch Mode](#)

exportapi.bat

```
"C:\SoftwareAG\Natural\bin\nderun.exe" BATCHMODE
PARM=SYSESM2 CMSYNIN="C:\exportapi.cmd" CMOBJIN="C:\exportapi.inp"
CMPRINT="C:\exportapi.out" BMSIM=MF NATLOG=OFF
```

File	Explanation
C:\SoftwareAG\Natural\bin\nderun.exe	Full path to Natural executable file.
PARM=SYSESM2	Natural configuration file name.
C:\exportapi.cmd	Full path to the file with commands (file must exist).
C:\exportapi.inp	Full path to the file with input data (file must exist).
C:\exportapi.out	Full path to the file where the output information will be exported to. File will not exist. It is created on first launch of batch file.

We recommend that you only change the last three parameters.

exportapi.cmd

```
LOGON SYSSATGF
ESSTRPCD
FIN
```

SYSSATGF is the name of the library from which Entire Operations should be launched.

We recommend that you do not change this parameter and file.

exportapi.inp

```
/* Maximum line length: 249.
/* Blank lines are not allowed.
/* Comment lines must begin with "/*".
/*
/* Input delimiter character (= ID parameter) separates user IDs from
/* passwords or substitutes them. Default is ",".
/*
/* First coded line must contain:
/* <natural-security-userid><input-delimiter-character><natural-security-password>
/* (or <input-delimiter-character> only)
/*
/* Second coded line must contain:
/* <entirex-userid><input-delimiter-character><entirex-password>
/* (or <input-delimiter-character> only)
/*
/* Parameter definitions must conform to the direct command syntax.
/*
/* File must end with the END statement; lines that follow are ignored.
/*
NSC-USERID,NSC-PSW-38292076742878722694553347482051
,
/*
FOCUS TO SYSOGC
/*
/*---- Start of direct commands
/*
EXPORT DIAGRAM OWNER=EXAMPLE NETWORK=B60-FLOW FORMAT=JPG PREFIX=C:\
/*
/* --- End of direct commands
/*
END
```

This is the main file you can modify as required.

General Syntax Requirements:

- A maximum of 249 characters is allowed per line.
- Blank lines are not allowed.

- Comment lines must begin with the symbol sequence /*.
- Any parameters definitions must comply with the direct command syntax.

The syntax of the exportapi.inp file is as follows:

```
[natural-security-userid],[natural-security-password]
[entirex-userid],[entirex-password]
FOCUS TO SYSOGC
direct-command ...
END
```

Syntax Element	Description
natural-security-userid	Optional user ID and password for Natural Security logon to be entered in the first coded.
natural-security-password	
	We recommend that you encrypt <i>natural-security-password</i> with the SYSSATGF\RES\nprpwc.exe program.
entirex-userid	Optional user ID and password to logon for EntireX logon to be entered in the second coded line.
entirex-password	
	We recommend that you encrypt <i>entirex-password</i> with the SYSSATGF\RES\nprpwc.exe program.
,	The input delimiter character (a comma in the syntax diagram above) as defined with the Natural session/profile parameter ID (described in the <i>Parameter Reference</i> of Natural for Windows).
(input-delimiter-character)	
	This character either separates a user ID from a password or substitutes a user ID and password if not available. The first and second coded source line can contain the following, for example:
	<pre>, ENTIREX-USERID, FOCUS TO SYSOGC ...</pre>
	The default input delimiter is a comma (,).
FOCUS TO SYSOGC	Logs on to the Natural SYSOGC system library.
direct-command	Insert the direct commands as required, each in a separate line.
END	The file must end with the END statement in a separate line. Any lines that follow are ignored.

EXPORT DIAGRAM Command Syntax

General Syntax:

```
EXPORT DIAGRAM OWNER=owner
[NETWORK=network [RUN=run-number] [VERSION=version]]
[FORMAT=format] [PREFIX=prefix]
```

Syntax Description:

Syntax Element	Description	Valid Values	Examples
<i>owner</i>	Network owner name	Alphanumeric, wildcard (*) allowed at the end of expression.	EXAMPLE, EX*, *
<i>network</i>	Network name. If a wildcard is used, all networks of a network master that meet the specified name range are selected. If no wildcard is used, all networks of a network master and all active networks are selected depending on whether <i>run-number</i> is specified.	Alphanumeric, wildcard (*) allowed at the end of expression.	NET01, NET*, *
<i>run-number</i>	Run number. If specified, all networks from the Active Network node are selected.	Numeric, asterisk (*) allowed for all numbers, colon (:) allowed for run range.	23, * 12:23 , not allowed: 2*
<i>version</i>	Version identifier of network master.	Alphanumeric, wildcard (*) allowed as an expression. Reserved keywords like (current) are not allowed. See also <i>Reserved Version Names for Symbol Tables in the Concepts and Facilities</i> documentation.	vTest, vTe*, not allowed: (current)
<i>format</i>	File extension for export format. Default is WMF.	Alphanumeric	BMP, EPS, GIF, JPG, PCX, PNG, TIF, WMF, VMF
<i>prefix</i>	Path to directory, where exported diagrams should be placed. Default is %temp%/.	A valid Windows path name See also Rules for Path Names and Directories .	"C:\temp\Export graphic via batch\ C:\temp\ C:\temp\batch01_

A wildcard (*) is only allowed in a parameter that defines an owner, network, version or run. When you define several of these parameters, you can only use a wildcard in the first parameter (all other wildcards are then ignored).

Examples of valid commands:

EXPORT DIAGRAM OWNER=* FORMAT=BMP	Export all network masters for all owners.
EXPORT DIAGRAM OWNER=EX* FORMAT=BMP	Export all network masters for owners starting with EX.
EXPORT DIAGRAM OWNER=EXAMPLE NETWORK=* FORMAT=GIF	Export all network masters for owner EXAMPLE.
EXPORT DIAGRAM OWNER=EXAMPLE NETWORK=NET* FORMAT=GIF	Export all network masters with prefix NET for owner EXAMPLE.
EXPORT DIAGRAM OWNER=EXAMPLE NETWORK=NET01 VERSION=vP* FORMAT=JPG	Export all versions with prefix vP of network masters for owner EXAMPLE and name NET01.
EXPORT DIAGRAM OWNER=EXAMPLE NETWORK=NET01 RUN=23 FORMAT=JPG	Export active network EXAMPLE/NET01/23.
EXPORT DIAGRAM OWNER=EXAMPLE NETWORK=NET01 RUN=* FORMAT=JPG	Export all active run networks of EXAMPLE/NET01.
EXPORT DIAGRAM OWNER=EXAMPLE NETWORK=NET01 RUN=12:23 FORMAT=JPG	Export all runs from 12 to 23 of network EXAMPLE/NET01.

Examples of Invalid Commands:

EXPORT DIAGRAM OWNER=EX* NETWORK=NET01 FORMAT=JPG
EXPORT DIAGRAM OWNER=EXAMPLE NETWORK=NET* RUN=23 FORMAT=JPG
EXPORT DIAGRAM OWNER=EXAMPLE NETWORK=NET* RUN=* FORMAT=JPG

Files Generated from EXPORT DIAGRAM Command

Files names are generated from the EXPORT DIAGRAM command according to the command input sequence in the following way:

prefix_owner_network_version_run.format

Rules for Path Names and Directories

- The default *prefix* for file name is : EXDIAGRAM_ownership_network_version_run.format
- If *prefix* is not specified, the default value %temp%/ is used.
- For a *prefix* that does not end with a backslash or slash an underscore (_) is generated instead into the path name.
- A relative path specification is not allowed.
- A specified directory must exist.

Examples of Files generated from a Command

1. `EXPORT DIAGRAM OWNER=EXAMPLE network=NET01 run=23 format=JPG ↵
prefix="C:\temp\Export graphic via batch\"`

generates:

```
C:\temp\Export graphic via batch\EXDIAGRAM_EXAMPLE_NET01_23.JPG
```

2. `EXPORT DIAGRAM OWNER=EXAMPLE network=NET* format=JPG prefix="C:\temp\Export ↵
graphic via batch\pref"`

generates:

```
C:\temp\Export graphic via batch\pref_EXAMPLE_NET01.JPG  
C:\temp\Export graphic via batch\pref_EXAMPLE_NET02.JPG  
C:\temp\Export graphic via batch\pref_EXAMPLE_NET03.JPG
```

Starting Entire Systems Management in Batch Mode

» To launch Entire Systems Management in batch mode

1. Adapt the example files according to your requirements (Natural directory SYSSATGF/RES: exportapi.bat, exportapi.cmd and exportapi.inp).
2. Launch the .bat file.
3. View the .out file (exportapi.out file) to see the results and output messages.

If an error occurs, ensure that Entire Systems Management was launched in normal mode using the correct parameters.

Files Needed to Start the Batch Command Client

The following files are required to start the batch command client of Entire Systems Management:

1. A Windows batch file (.bat)
2. An XML batch file (.xml)

Examples of such files are located in the Natural SYSSATGF/RES directory: batchClientCommand.bat and batchClientReports.xml.

- [batchClientCommand.bat](#)

- [batchClientReports.xml](#)

batchClientCommand.bat

```
SET dirNatural=<natural-product-directory>
SET ESM_BCC_IN=<natural-fnat-directory>\SYSOGC\RES\batchClientReports.xml
SET ESM_BCC_OUT=<user-temp-directory>\batchClientReports_out.xml

"%dirNatural%\bin\nderun.exe" : PARM=SYSESM2 STACK=(LOGON SYSSATGF;ESSTBC-P;FIN)
```

Data	Explanation
<i>natural-product-directory</i>	Full path to the Natural executable file.
PARM=SYSESM2	Natural configuration file name.
<i>natural-fnat-directory</i>	Full path to the Natural FNAT system file with input data. Note: File must exist.
<i>user-temp-directory</i>	Full path to the file where the output information will be exported to. File will not exist. It is created on the first launch of the batch file.
STACK=(LOGON SYSSATGF;ESSTBC-P;FIN)	Name of the library from which to launch the batch command client of Entire Systems Management and the main program. We recommend that you do not change this name.

Environment Variable	Explanation
ESM_BCC_IN	Mandatory variable containing the full path to input data file containing the instructions.
ESM_BCC_OUT	Mandatory variable containing the full path to the file where the output information will be exported to. File will not exist. It is created on first launch of batch file.
ESM_BCC_TRACE	Optional variable to specify the trace level which will be written to the log file.
ESM_BCC_LOG	Optional variable containing the full path of the file where the log/trace information will be written to. File will not exist. It is created on first launch of batch file.

batchClientReports.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<ESM_Batch_Command_Client xmlns="http://www.softwareag.com/ESM/sat34">
  <RPC User="<userid>" Password="<password>" />
  <EXX ServerName="<server-name>" ServerNode="<server-node>" />
  <ESM ApplicationID="OP" Language="2">
    <Command Object="Reports" Name="List">
      <Parameter Name="Filter_Status">Completed</Parameter>
    </Command>
  </ESM>
</ESM_Batch_Command_Client>
```

Data	Explanation
<i>userid</i>	ID of a valid user.
<i>password</i>	Valid password of a user. We recommend that you encrypt <i>password</i> with the SYSSATGF\RES\nprpwc.exe program.
<i>server-name</i>	Valid server name. See also the Natural SRVNAME profile parameter described in the Natural for Windows <i>Parameter Reference</i> documentation.
<i>server-node</i>	Node name. See also the Natural SRVNODE profile parameter described in the Natural for Windows <i>Parameter Reference</i> documentation.

Command Syntax for the Batch Command Client

The batch command client of Entire Systems Management is programmed with an XML file.

XML documents use a self-describing and simple syntax. The first line is the XML declaration. It defines the XML version (1.0) and the encoding used. The second line describes the root element of the document. It is the parent element of all other elements. These child elements contain the description of the further on data or in our case processing. The last line defines the end of the root element. All tags and attributes used inside the xml file are case sensitive.

Encoding

If no encoding is given it is supposed that the XML document only contains standard ASCII characters. Special characters like the German *ü* or *ä* are not valid. If you want to use this kind of characters you have to use the encoding setting. For editing an xml file use an editor that supports encoding. Make sure that you use the same encoding during editing that is specified at the encoding attribute. Otherwise, your document may not load or your data is misinterpreted.

DTD Input File

```
<!ELEMENT ESM_Batch_Command_Client (RPC, EXX?, ESM)>
<!ATTLIST ESM_Batch_Command_Client
  xmlns          CDATA #FIXED "http://www.softwareag.com/ESM/SAT34"
>

<!ELEMENT RPC EMPTY>
<!ATTLIST RPC
  Password      CDATA #REQUIRED
  User          CDATA #REQUIRED
>

<!ELEMENT EXX EMPTY>
<!ATTLIST EXX
  ServerName    CDATA #REQUIRED
  ServerNode    CDATA #REQUIRED
  Password      CDATA #IMPLIED
  User          CDATA #IMPLIED
>
```

```

>
<!ELEMENT ESM ( Command+ )>
<!ATTLIST ESM
  ApplicationID      CDATA #REQUIRED
  Language           CDATA #REQUIRED
  DeterminationDate CDATA #IMPLIED
  ServerLibrary      CDATA #IMPLIED
>

<!ELEMENT Command (Parameter*)>
<!ATTLIST Command
  Object            CDATA #IMPLIED
  Name              CDATA #REQUIRED
>

<!ELEMENT Parameter (#PCDATA)>
<!ATTLIST Parameter
  Name             CDATA #REQUIRED>

```

Tag Reference

Tag	Explanation
<ESM_Batch_Command_Client>	The <ESM_Batch_Command_Client> tells the processor of the batch command client of Entire Systems Management the remote environment to be used and the commands to be processed.
<RPC>	Defines the Natural RPC connection with user and password.
<EXX>	Defines the EntireX remote server and service and if necessary user and password.
<ESM>	Defines the Entire Systems Management application and additional settings such as language or determination date.
<Command>	Defines a single command which should be processed.
<Parameter>	Defines a parameter of a command.

The tags are explained in the following section.



Note: Within the following pages the tables contain a format column. The specified formats are based on the definition of Natural user-defined variables. Additionally, the format “enum” is used if only a fixed string out of a list of values can be used. “const” specifies a fixed string of the Natural data format A.

- [<Command> Tag](#)
- [<ESM_Batch_Command_Client> Tag](#)
- [<ESM> Tag](#)
- [<EXX> Tag](#)
- [<Parameter> Tag](#)

- **<RPC> Tag**

<Command> Tag

The **<Command>** tag defines the commands that should be processed. The element contains the child element **<Parameter>**.

The following attributes are available:

Attribute	Format	Value	Description
Object	enum	Entire Systems Management object type	Name of the object type for which the command should be processed.
Name	enum	Name of the command	Name of a command for a specific object type.

Example:

```
<Command Object="Reports" Name="List"/>
```

Output:

```
<Command Object="Reports" Name="List">
```

```
</Command>
```

<ESM_Batch_Command_Client> Tag

The **<ESM_Batch_Command_Client>** tag defines the xml document that provides all settings and the sequence of the commands to be processed. At the moment only report generation related commands of Entire Operations can be processed. The default namespace (**xmlns** attribute) of the document defines that the used elements belongs to the batch command client application of Entire Systems Management.

The following attributes are available:

Attribute	Format	Value	Description
xmlns	const	http://www.softwareag.com/ESM/SAT34	Specifies the XML namespace attribute.

Example of an XML document with encoding setting and root element:

```
<?xml version="1.0" encoding="utf-8" ?>
<ESM_Batch_Command_Client xmlns="http://www.softwareag.com/ESM/SAT34">

</ESM_Batch_Command_Client>
```

<ESM> Tag

The <ESM> tag defines settings for the Entire Systems Management application.

The following attributes are available:

Attribute	Format	Value	Description
ApplicationID	enum	ID of the application	For Entire Operations GUI Client use the application ID OP.
Language	N2	Application language	Same as Natural system variable *LANGUAGE: 1 for English (default) 2 for German
ServerLibrary	A8	Natural library name	Optional: Specifies the server library in case not use the default name SYSSAT is used.

Example:

```
<ESM ApplicationID="OP" Language="2"/>
```

Output:

```
<?xml version="1.0" encoding="windows-1252" ?>
<Commands ApplicationID="OP" ApplicationName="Entire Operations" ↵
ApplicationLibrary="SYSOGC" ApplicationVersion="0504010000">

</Commands>
```

<EXX> Tag

The <EXX> tag defines EntireX settings. The server is defined by a ServerName and the ServerNode settings. The settings are equal to the Natural profile parameter for Entire Systems Management (see the Natural SYSESM2 configuration file).

The following attributes are available:

Attribute	Format	Value	Description
Name	A	Natural SRVNAME profile parameter	Name of RPC server
Node	A	Natural SRVNODE profile parameter	Name of node
User	A	Valid EntireX user ID	EntireX user ID
Password	A	Valid EntireX password We recommend that you encrypt the password with the SYSSATGF\RES\nprpwc.exe program.	EntireX password

Example:

```
<EXX ServerName="NOP54SRV" ServerNode="bkr034:3800:TCP"/>
```

<Parameter> Tag

The <Parameter> tag defines parameter for a command.

The following attributes are available:

Attribute	Format	Value	Description
Name	enum	Name of a parameter	See Command description.
Value	A	Value of a parameter	See Command description.

Example:

```
<Command Object="Reports" Name="List">
  <Parameter Name="Filter_Status">Completed</Parameter>
</Command>
```

<RPC> Tag

The <RPC> tag defines setting for the Natural RPC command.

The following attributes are available:

Attribute	Format	Explanation
User	A	Valid Entire Systems Management user ID
Password	A	Valid Entire Systems Management password

Example:

```
<RPC User="<user>" Password="<password>" />
```

Commands for the Batch Command Client

This section describes the commands provided by the batch command client of Entire Systems Management.

- Wildcard Usage
- Command - Object: Reports - Name: List
- Command - Object: Reports - Name: SaveToFile
- Command - Object: CrossReferences - Name: SaveToFile
- Command - Object: Reports - Name: Add
- Command - Object: Reports - Name: Delete
- Command - Object: - Name: DeterminationDate
- Command - Object: - Name: Language
- Command - Object: - Name: Stop
- Variable Handling
- Relative Date Feature

Wildcard Usage

The wildcard character has to be the last character in a variable value that defines a range of names.

Wildcard Character	Meaning
*	Only select objects for which the name begins with the given prefix.
>	Only select objects with a name greater than the given value.
<	Only select objects with a name less than the given value.



Note: The use of wildcards can be disabled in the User Profile. For further information, see the *Reporting* section in the *User's Guide*.

Wildcard Usage (Owner, Network, Symbol Table, Version)

Every object is referenced according his namespace parts. Only the last part of the namespace can be extended with a wildcard.

Valid Specification

```
<Owner>EXAMPLE</Owner>
<Network>*</Network>
```

Invalid Specifications

```
<Owner>EXAMPLE</Owner>
<Network>E*</Network>
<NetworkVersion>v2*</NetworkVersion>
```

```
<Owner>EXAMPLE</Owner>
<NetworkVersion>v2*</NetworkVersion>
```

```
<Owner>E*</Owner>
<NetworkVersion>v2*</NetworkVersion>
```

Command - Object: Reports - Name: List

The `List` command evaluates data from the Reports list. This command provides major usages:

- Find reports of a specific kind by specifying filters e.g. to delete this reports or
- Wait until a report generation has been ended with the status `Aborted` or `Completed` e.g. to save `Completed reports` as file.

The `<Parameter>` tag defines the parameters for the command.

Parameter	Format	Value	Explanation
Filter_Status	enum	Completed Aborted Running	Status of report generation.
Filter_ShortType	enum	A valid short type (see: Add command)	Report types.
Filter_CreationDate	Date	A valid Natural date with format YYYYMMDD	Date the report has been added.
WaitID	A	A FixedID or a VariableName	ID of a report given as variable name or as fixed value.
AutoRefresh_Retries	I2	Number greater than 0	Number of retries if the correct value is delivered.
AutoRefresh_Interval	I2	Time in seconds	Time to wait between two retries.
returnID	A	A VariableArrayName	Name of an array containing the IDs of the found objects.

Examples:

```
<Command Object="Reports" Name="List">
  <Parameter Name="Filter_Status">Completed</Parameter>
</Command>
```

```
<Command Object="Reports" Name="List">
  <Parameter Name="WaitID ">%new_ID%</Parameter>
  <Parameter Name="AutoRefresh_Retries">20</Parameter>
  <Parameter Name="AutoRefresh_Interval">5</Parameter>
</Command>
```

Output:

```
<Command Object="Reports" Name="List">
  <Filter Name="Status">Completed</Filter>
  <Report>
    <ID>3092265</ID>
    <Status>Completed</Status>
    <Type ShortType="LGT">Log - Terminated Jobs</Type>
    <Creation><Date Format="G">20131115</Date><Time>155134</Time></Creation>
  </Report>
  <Report>
    <ID>3092267</ID>
    <Status>Completed</Status>
    <Type ShortType="LGB">Log - Abended Jobs</Type>
    <Creation><Date Format="G">20131115</Date><Time>155136</Time></Creation>
  </Report>
  <Report>
    <ID>3092268</ID>
    <Status>Completed</Status>
    <Type ShortType="LGX">Log - Jobs not started</Type>
    <Creation><Date Format="G">20131115</Date><Time>155138</Time></Creation>
  </Report>
</Command>
```

```
<Command Object="Reports" Name="List">
  <WaitFor ID="3190047"/>
  <AutoRefresh Interval="10">
    <Retry Loop="0">07:09:51.0</Retry>
    <Retry Loop="1">07:10:01.2</Retry>
    <Retry Loop="2">07:10:13.8</Retry>
    <WaitResult Retries="2">Completed</WaitResult>
  </AutoRefresh>
</Command>
```

Command - Object: Reports - Name: SaveToFile

The `SaveToFile` command saves completed report generations with different output formats as file.

If the output type is XML an additional `XmlStylesheet` parameter can be provided. This can be used as replacement of the online type "XML with style sheet for HTML". The style sheet has to be copied manually. Examples of style sheets can be found in the Natural SYSOGC/RES directory. The files are named `REPreport-name.xsl`.

The `<Parameter>` tag defines the parameters for the command.

Parameter	Format	Value	Explanation
ID	A	A <code>FixedID</code> or a <code>VariableName</code>	ID of a report given as variable name or as fixed value.
OutputType	enum	HTML XML CSV	Document type of the generated report. In general the types Extensible Markup Language (XML) and HyperText Markup Language (HTML) available. Some reports can be also saved as comma-separated values (CSV).
FileName	A250	Valid local file name to the operating system	A valid path to your file system where the report will be saved.
XmlStylesheet	A	Uniform Resource Identifier (URI) as absolute or relative address	If specified, and the report type is XML, a processing instruction with a reference to an external style sheet is added to generated XML Document. This processing instruction (<code><?xml-stylesheet... ?></code>) will be interpreted by browsers to automatically render the XML document e.g. as HTML. You can find style sheets for every report type in the SYSOGC/RES directory of the Natural FNAT system file.

Example:

```
<Command Object="Reports" Name="SaveToFile">
  <Parameter Name="ID">%new_ID_AC0%</Parameter>
  <Parameter Name="OutputType">HTML</Parameter>
  <Parameter Name="FileName">C:\AC0.html</Parameter>
</Command>
```

Output:

```
<Command Object="Reports" Name="SaveToFile">
  <Report ID="3190047" ShortType="LGY" OutputType="HTML" />
  <ExportFile>D:\ESM\Reports\LGY.html</ExportFile>
</Command>
```

Command - Object: CrossReferences - Name: SaveToFile

The `SaveToFile` command for `Object="CrossReferences"` generates a list of cross referenced objects. The types of cross references that can be generated are listed in the following table. They are further explained in the section *Types of Cross Reference Reports* in the section *Cross References*.

Types of Cross References

Cross Reference Type	Short Type
JCL usage	JCL
Node usage	NDU
Undefined objects	NDF
Resource usage	RES
Symbol table usage	STU
User exit usage	URU
Symbol search by value	XSM

Command Parameters

The `<Parameter>` tag defines the parameters for the command:

Parameter	Format	Description		
ShortType	enum	Type of cross reference generation Specify a valid short type .		
Owner	A10	Uppercase name of an owner or a range of names ¹ Default value: * (all owners)		
Network	A10	Uppercase name of a network or a range of names ¹		
NetworkVersion	A10	Name of a network version or a range of names ¹		
SymbolTable	A10	Uppercase name of a symbol table or a range of names ¹		
SymbolTableVersion	A10	Version of a symbol table or a range of names ¹		
UserExitUsage	enum	Type of user exit <hr/> Valid values: <hr/> <table border="1"> <tr> <td>IC</td> <td>Input condition user exits</td> </tr> </table>	IC	Input condition user exits
IC	Input condition user exits			

Parameter	Format	Description																								
		<table border="1"> <tr> <td>RM</td> <td>Resource master determination exits</td> </tr> <tr> <td>MJ</td> <td>Master JCL and Natural programs</td> </tr> <tr> <td>EC</td> <td>End-of-Job check user exits</td> </tr> <tr> <td>EA</td> <td>End-of-Job action user exits</td> </tr> <tr> <td>SC</td> <td>Symbol check user exits</td> </tr> <tr> <td>SP</td> <td>Symbol prompt user exits</td> </tr> </table>	RM	Resource master determination exits	MJ	Master JCL and Natural programs	EC	End-of-Job check user exits	EA	End-of-Job action user exits	SC	Symbol check user exits	SP	Symbol prompt user exits												
RM	Resource master determination exits																									
MJ	Master JCL and Natural programs																									
EC	End-of-Job check user exits																									
EA	End-of-Job action user exits																									
SC	Symbol check user exits																									
SP	Symbol prompt user exits																									
UsageInActiveJobs	L	<p>Determines the jobs in which to search</p> <hr/> <p>Valid values:</p> <table border="1"> <tr> <td>false</td> <td>Searches in all jobs (default)</td> </tr> <tr> <td>true</td> <td>Searches in active jobs only</td> </tr> </table>	false	Searches in all jobs (default)	true	Searches in active jobs only																				
false	Searches in all jobs (default)																									
true	Searches in active jobs only																									
JCLLocation	enum	<p>Supported JCL location</p> <hr/> <p>Valid values:</p> <table border="1"> <tr> <td><i>blank</i></td> <td>A blank character selects all JCL locations.</td> </tr> <tr> <td>BS2</td> <td>SAM or ISAM file</td> </tr> <tr> <td>EXE</td> <td>Executable program</td> </tr> <tr> <td>LIB</td> <td>Librarian</td> </tr> <tr> <td>LMS</td> <td>LMS library</td> </tr> <tr> <td>MAC</td> <td>Natural source object containing a macro program</td> </tr> <tr> <td>NAT</td> <td>Natural source object</td> </tr> <tr> <td>PDS</td> <td>Partitioned data set</td> </tr> <tr> <td>PRC</td> <td>Callable procedure</td> </tr> <tr> <td>RDR</td> <td>z/VSE reader queue</td> </tr> <tr> <td>TXT</td> <td>Text file</td> </tr> <tr> <td>VSE</td> <td>z/VSE sublibrary</td> </tr> </table>	<i>blank</i>	A blank character selects all JCL locations.	BS2	SAM or ISAM file	EXE	Executable program	LIB	Librarian	LMS	LMS library	MAC	Natural source object containing a macro program	NAT	Natural source object	PDS	Partitioned data set	PRC	Callable procedure	RDR	z/VSE reader queue	TXT	Text file	VSE	z/VSE sublibrary
<i>blank</i>	A blank character selects all JCL locations.																									
BS2	SAM or ISAM file																									
EXE	Executable program																									
LIB	Librarian																									
LMS	LMS library																									
MAC	Natural source object containing a macro program																									
NAT	Natural source object																									
PDS	Partitioned data set																									
PRC	Callable procedure																									
RDR	z/VSE reader queue																									
TXT	Text file																									
VSE	z/VSE sublibrary																									
FileNatLib	A54	Name of a file or Natural library or a range of names ¹																								
Member	A64	Name of a member or a range of names ¹																								
JCLNode_OpSysClass	enum	<p>Operating system for specified JCL node</p> <hr/> <p>Valid values:</p> <table border="1"> <tr> <td>B</td> <td>BS2000</td> </tr> <tr> <td>M</td> <td>z/OS (MVS)</td> </tr> <tr> <td>V</td> <td>z/VSE</td> </tr> </table>	B	BS2000	M	z/OS (MVS)	V	z/VSE																		
B	BS2000																									
M	z/OS (MVS)																									
V	z/VSE																									

Parameter	Format	Description										
		<table border="1"> <tr> <td>W</td> <td>Windows</td> </tr> <tr> <td>X</td> <td>UNIX</td> </tr> </table>	W	Windows	X	UNIX						
W	Windows											
X	UNIX											
JCLNode_NodeNumber	N5	Numer of a node										
ExecutionNode_OpSysClass	enum	<p>Operating system for specified execution node</p> <hr/> <p>Valid values:</p> <table border="1"> <tr> <td>B</td> <td>BS2000</td> </tr> <tr> <td>M</td> <td>z/OS (MVS)</td> </tr> <tr> <td>V</td> <td>z/VSE</td> </tr> <tr> <td>W</td> <td>Windows</td> </tr> <tr> <td>X</td> <td>UNIX</td> </tr> </table>	B	BS2000	M	z/OS (MVS)	V	z/VSE	W	Windows	X	UNIX
B	BS2000											
M	z/OS (MVS)											
V	z/VSE											
W	Windows											
X	UNIX											
ExecutionNode_NodeNumber	N5	Number of a node										
UsageIn	enum	<p>Jobs for which to search</p> <hr/> <p>Valid values:</p> <table border="1"> <tr> <td>A</td> <td>Active jobs</td> </tr> <tr> <td>M</td> <td>Master jobs</td> </tr> </table> <hr/> <p>Multiple values are allowed for ShortType XSM (symbol search by value).</p>	A	Active jobs	M	Master jobs						
A	Active jobs											
M	Master jobs											
Symbol	A40	Name of a symbol or a range of names ¹										
MultValueMinIndex	N3	<p>Minimum index value of a multiple-value symbol</p> <p>Valid values: 1 to 100</p> <p>Default value: 1</p>										
MultValueMaxIndex	N3 or A1	<p>Minimum index value of a multiple-value symbol</p> <p>Valid values: 1 to 100 or 1 to * (all multiple values)</p> <p>Default value: *</p>										
AtPosition	enum	<p>Position at which to search within the symbol value</p> <hr/> <p>Valid values:</p> <table border="1"> <tr> <td>*</td> <td>Anywhere within the symbol value (default)</td> </tr> <tr> <td>1</td> <td>At the beginning of the symbol value</td> </tr> <tr> <td>E</td> <td>At the end of the symbol value</td> </tr> </table>	*	Anywhere within the symbol value (default)	1	At the beginning of the symbol value	E	At the end of the symbol value				
*	Anywhere within the symbol value (default)											
1	At the beginning of the symbol value											
E	At the end of the symbol value											

Parameter	Format	Description																
ScanFor	A80	String of up to 80 characters for which to search within the symbol value																
CaseSensitive	L	Specify <code>true</code> to perform a case-sensitive search or <code>false</code> (default) for a case-insensitive search.																
ObjectTypeToBeValidated	enum	Type(s) of objects for which to search Valid values: <table border="1"> <tr> <td>*</td> <td>All types of objects (default)</td> </tr> <tr> <td>SM</td> <td>Symbols</td> </tr> <tr> <td>S1</td> <td>Symbol tables</td> </tr> <tr> <td>NM</td> <td>Networks</td> </tr> <tr> <td>JM</td> <td>Jobs</td> </tr> <tr> <td>RM</td> <td>Resources</td> </tr> <tr> <td>SC</td> <td>Schedules</td> </tr> <tr> <td>CA</td> <td>Calendars</td> </tr> </table>	*	All types of objects (default)	SM	Symbols	S1	Symbol tables	NM	Networks	JM	Jobs	RM	Resources	SC	Schedules	CA	Calendars
*	All types of objects (default)																	
SM	Symbols																	
S1	Symbol tables																	
NM	Networks																	
JM	Jobs																	
RM	Resources																	
SC	Schedules																	
CA	Calendars																	
Resource	A40	Name of a resource or a range of names ¹																
FileName	A250	Name of a valid directory path where the report is saved																
OutputType	enum	Type of the report file to generate Valid values: <table border="1"> <tr> <td>HTML</td> <td>HTML-formatted file</td> </tr> <tr> <td>XML</td> <td>XML-formatted file</td> </tr> <tr> <td>CSV</td> <td>Comma-separated value file</td> </tr> </table>	HTML	HTML-formatted file	XML	XML-formatted file	CSV	Comma-separated value file										
HTML	HTML-formatted file																	
XML	XML-formatted file																	
CSV	Comma-separated value file																	
XmlStylesheet	A	Uniform Resource Identifier (URI) to specify an absolute or a relative address If an URI is specified for <code>OutputType XML</code> , a processing instruction with a reference to an external style sheet is added to the generated XML file. The browser then interprets this instruction (<code><?xml-stylesheet...?></code>) and renders the requested file form, for example, HTML. You can find style sheets for all report types in the <code>SYSOGC/RES</code> directory of the Natural FNAT system file.																

¹ You can specify a range of names by using an asterisk (*) to determine a start value or to select all names available (see also *Valid Name Specifications in Direct Commands*).

The command only processes a single range definition and ignores any additional definitions. If more than one range is defined in a command, the parameters are processed in the following

order of precedence (from highest to lowest): Owner, Network, NetworkVersion, SymbolTable, SymbolTableVersion. For example: If ranges are specified for Owner, Network and SymbolTable, only the range specified for Owner (highest priority) is processed.

Use of Parameters Dependent on Cross Reference Type

Cross Reference Type	JCL	NDF	NDU	RES	STU	URU	XSM
ShortType	X	X	X	X	X	X	X
Owner	X	X	X	X	X		X
Network	X	X	X	X			
NetworkVersion	X	X	X	X			
SymbolTable					X		X
SymbolTableVersion					X		X
Symbol							X
UserExitUsage						X	
UsageInActiveJobs					X		
JCLLocation	X						
FileNatLib	X						
Member	X						
JCLNode_OpSysClass			X				
JCLNode_NodeNumber			X				
ExecutionNode_OpSysClass			X				
ExecutionNode_NodeNumber			X				X
UsageIn			X				X
MultValueMinIndex							X
MultValueMaxIndex							X
AtPosition							X
ScanFor							X
CaseSensitive							X
ObjectTypeToBeValidated		X					
Resource				X			
FileName	X	X	X	X	X	X	X
OutputType	X	X	X	X	X	X	X
Xmlstylesheet	X	X	X	X	X	X	X

DTD of the Generated Output

```

<!ELEMENT Command (((Error+, CrossReference, Owner, Network, NetworkVersion,
JCLLocation, FileNatlib, Member) | (CrossReference, (((UserExitUsage | ↵
(XmlStylesheet,
Owner, SymbolTable, UsageInActiveJobs)), ExportFile) | (Owner, ((SymbolTable,
(UsageInActiveJobs | (Symbol, UsageIn, MultValueMinIndex, MultValueMaxIndex, ↵
AtPosition,
ScanFor, CaseSensitive)), ExportFile) | (Network, ((ObjectTypeToBeValidated+, ↵
ExportFile) |
(NetworkVersion, (((Resource | (JCLLocation, FileNatlib, Member)), ExportFile) |
(JCLNode_OpSysClass, JCLNode_NodeNumber, ExecutionNode_OpSysClass,
ExecutionNode_NodeNumber, UsageIn)))))))))>
<!ATTLIST Command
  Name CDATA #REQUIRED
  Object CDATA #REQUIRED
>
<!ELEMENT CaseSensitive (#PCDATA)>
<!ELEMENT AtPosition (#PCDATA)>
<!ATTLIST AtPosition
  ShortType CDATA REQUIRED
>
<!ELEMENT XmlStylesheet (#PCDATA)>
<!ELEMENT UserExitUsage (#PCDATA)>
<!ATTLIST UserExitUsage
  ShortType CDATA REQUIRED
>
<!ELEMENT UsageInActiveJobs (#PCDATA)>
<!ELEMENT UsageIn (#PCDATA)>
<!ATTLIST UsageIn
  ShortType CDATA #REQUIRED
>
<!ELEMENT SymbolTable (#PCDATA)>
<!ELEMENT Symbol EMPTY>
<!ELEMENT ScanFor EMPTY>
<!ELEMENT Resource EMPTY>
<!ELEMENT Owner (#PCDATA)>
<!ELEMENT ObjectTypeToBeValidated (#PCDATA)>
<!ATTLIST ObjectTypeToBeValidated
  ShortType CDATA #REQUIRED
>
<!ELEMENT NetworkVersion (#PCDATA)>
<!ELEMENT Network (#PCDATA)>
<!ELEMENT MultValueMinIndex (#PCDATA)>
<!ELEMENT MultValueMaxIndex (#PCDATA)>
<!ELEMENT Member (#PCDATA)>
<!ELEMENT JCLNode_OpSysClass EMPTY>
<!ELEMENT JCLNode_NodeNumber EMPTY>
<!ELEMENT JCLLocation (#PCDATA)>
<!ATTLIST JCLLocation
  ShortType CDATA REQUIRED

```

```

>
<!ELEMENT FileNatLib (#PCDATA)>
<!ELEMENT ExportFile (#PCDATA)>
<!ELEMENT ExecutionNode_OpSysClass (#PCDATA)>
<!ELEMENT ExecutionNode_NodeNumber (#PCDATA)>
<!ELEMENT Error (#PCDATA)>
<!ATTLIST Error
  Type CDATA #REQUIRED
>
<!ELEMENT CrossReference (#PCDATA)>
<!ATTLIST CrossReference
  OutputType (XML | HTML | CSV) #REQUIRED
  ShortType CDATA #REQUIRED
>

```

Examples of Cross Reference SavetoFile Commands

```

<Command Name="SaveToFile" Object="CrossReferences">
  <Parameter Name="ShortType">JCL</Parameter>
  <Parameter Name="Owner">INCIDENT</Parameter>
  <Parameter Name="Network">*</Parameter>
  <Parameter Name="OutputType">HTML</Parameter>
  <Parameter Name="FileName">C:\temp\G_xRef_JclUsage.html</Parameter>
</Command>

```

```

<Command Name="SaveToFile" Object="CrossReferences">
  <Parameter Name="ShortType">NDF</Parameter>
  <Parameter Name="Owner">INCIDENT</Parameter>
  <Parameter Name="Network">*</Parameter>
  <Parameter Name="ObjectTypeToBeValidated">*</Parameter>
  <Parameter Name="OutputType">HTML</Parameter>
  <Parameter Name="FileName">C:\temp\xRef_UndefinedObjects.html</Parameter>
</Command>

```

```

<Command Name="SaveToFile" Object="CrossReferences">
  <Parameter Name="ShortType">NDU</Parameter>
  <Parameter Name="Owner">INCIDENT</Parameter>
  <Parameter Name="Network">*</Parameter>
  <Parameter Name="OutputType">HTML</Parameter>
  <Parameter Name="FileName">C:\temp\xRef_NodeUsage.html</Parameter>
</Command>

```

```

<Command Name="SaveToFile" Object="CrossReferences">
  <Parameter Name="ShortType">RES</Parameter>
  <Parameter Name="Owner">*</Parameter>
  <Parameter Name="OutputType">XML</Parameter>
  <Parameter Name="FileName">C:\temp\xRef_Resource_Usage.xml</Parameter>
</Command>

```

```
<Command Name="SaveToFile" Object="CrossReferences">
  <Parameter Name="ShortType">STU</Parameter>
  <Parameter Name="Owner">INCIDENT</Parameter>
  <Parameter Name="SymbolTable">*</Parameter>
  <Parameter Name="OutputType">CSV</Parameter>
  <Parameter Name="FileName">C:\temp\xRef_SymbolTableUsage.csv</Parameter>
</Command>
```

```
<Command Name="SaveToFile" Object="CrossReferences">
  <Parameter Name="ShortType">URU</Parameter>
  <Parameter Name="UserExitUsage">EC</Parameter>
  <Parameter Name="OutputType">XML</Parameter>
  <Parameter Name="FileName">C:\temp\xRef_UserExitUsage_EC.xml</Parameter>
</Command>
```

```
<Command Name="SaveToFile" Object="CrossReferences">
  <Parameter Name="ShortType">XSM</Parameter>
  <Parameter Name="Owner">INCIDENT</Parameter>
  <Parameter Name="SymbolTable">*</Parameter>
  <Parameter Name="OutputType">CSV</Parameter>
  <Parameter Name="FileName">C:\temp\xref_SearchSymbolByValue.csv</Parameter>
</Command>
```

Command - Object: Reports - Name: Add

The Add command invokes the user-dependent report generation. The generation is invoked within the Entire Operations GUI Client whereas the reports themselves are generated asynchronously in the background via the Entire Operations monitor. Reports added with the batch command client of Entire Systems Management can be seen by performing the report functions of the Entire Operations GUI Client.

Report Types

Short Type	Report *	Description
ACC	Accounting Data	Information on job elapsed times and CPU times of previous network executions.
ACO	Activation Overview	An overview of network activations.
CNV	Compare Networks	Compare a network with another network.
CST	Compare Symbol Tables	Compare a symbol table with other symbol tables.
LGB	Log - Abended Jobs	Information on jobs, which is extracted from the Entire Operations log.
LGN	Log - Networks not activated	
LGT	Log - Terminated Jobs	
LGX	Log - Jobs not started	
LGY	Log - Jobs with permanent errors	

Short Type	Report *	Description
NDD	Network Description (detailed)	Displays the same information as the short version, but includes all prose descriptions defined at the network, job or event level using the Editor facility.
NDS	Network Description (short)	Displays information on networks and jobs as defined on the master database, including scheduling information, prerequisites and end-of-job checking and actions.
NOD	Node Overview	Displays an overview of node reports.
NSO	Network Schedule Overview	An overview of scheduled and/or not yet executed, system-wide network activations.
NSS	Network Start Summary	A status report of all network starts for a given day.
SCJ	Schedule of Jobs	Displays a job schedule for a specific date range.
XRF	Subnetwork Usage	Lists networks and related subnetworks.
BC1	Network Start Overview (Bar Chart)	Displays the network start and end times of a previous network execution as a bar chart.
BC3	Network Schedule Overview (Bar Chart)	Displays the schedule selection of a network as a bar chart.
BC4	Network and Job Start Overview (Bar Chart)	Displays the network and job starts as a bar chart.

* For further information, see *Reporting* in the *User's Guide*.

Command Parameters

The <Parameter> tag defines the parameters for the command:

Parameter	Format	Default	Value	Explanation
ShortType	enum		A valid short type	Report generation type.
Owner	A10		Uppercase name (may be terminated with wildcard characters)	
Network	A10		Uppercase name (may be terminated with wildcard characters)	
NetworkVersion	A10		Name (may be terminated with wildcard characters)	
Job	A10	*	Uppercase name (may be terminated with wildcard characters)	
Date_From	D	current date	A valid date with format YYYYMMDD	Specify the start date and time for displaying the messages.
Date_Thru	D	current date	A valid date with format YYYYMMDD	Specify the start date and time for

Parameter	Format	Default	Value	Explanation
				displaying the messages.
Time_From	T	000000	A valid date with format HHISS	
Time_Thru	T	current time	A valid date with format HHISS	
Run_From	N5	1	Positive numeric	
Run_Thru	N5	99999	Positive numeric	
ScheduleType	enum	SNT	SNT Schedule, sorted by network, time ONT Only sorted by network, time OT Only sorted by time STN Schedule, sorted by time, network	
ActivationType	enum	All types	All types Manual Recovery API EOJ Scheduled	Specifies the activation type. Can be specified multiple times. Note: All types cannot be used with other activation types.
Email	A98			Enter an e-mail address if you want to send a notification when the report generation has been completed.
Owner2	A10		Uppercase name (may be terminated with wildcard characters)	
Network2	A10		Uppercase name (may be terminated with wildcard characters)	
NetworkVersion2	A10		Version (may be terminated with wildcard characters)	
Job2	A10	*	Uppercase name (may be terminated with wildcard characters)	
Node_From	N5	1	A positive numeric value	
Node_Thru	N5	1	A positive numeric value	
SymbolTable	A10		Uppercase name (may be terminated with wildcard characters)	

Parameter	Format	Default	Value	Explanation
SymbolTableVersion	A10		Version (may be terminated with wildcard characters)	
SymbolTable2	A10		Uppercase name (may be terminated with wildcard characters)	
SymbolTableVersion2	A10		Version (may be terminated with wildcard characters)	
Show	enum	All	All Differences	
returnID	A		VariableName	

Use of Parameters Dependent on Report Type

Parameter /Report Type	ACC	ACO	CNV	CST	LGB	LGN	LGT	LGX	LGY	NDD	NDS	NOD	NSO	NSS	SCJ	XRF	BC1	BC3	BC4
ShortType	X	X	X	X	X	X	X	X	X	X	X		X	X	X	X	X	X	X
Owner	X	X	X	X	X	X	X	X	X	X	X		X	X	X	X	X	X	X
Network	X	X	X		X	X	X	X	X	X	X		X	X	X	X	X	X	X
NetworkVersion	X	X	X		X	X	X	X	X	X	X		X	X	X	X	X	X	X
SymbolTable				X															
SymbolTableVersion				X															
Job	X				X	X	X	X	X							X			
Date_From	X	X			X	X	X	X	X				X	X	X		X	X	X
Date_Thru	X	X			X	X	X	X	X				X	X	X		X	X	X
Time_From	X	X			X	X	X	X	X										
Time_Thru	X	X			X	X	X	X	X										
Run_From	X	X			X	X	X	X	X										
Run_Thru	X	X			X	X	X	X	X										
ScheduleType													X						
ActivationType		X																	
Email	X	X	X	X	X	X	X	X	X	X	X		X	X	X				
Owner2			X	X												X			
Network2			X													X			
NetworkVersion2			X													X			
Node_From												X							
Node_Thru												X							
SymbolTable2				X															
SymbolTableVersion2				X															
Job2																X			

Show				X															
returnID	X	X	X	X	X	X	X	X	X	X	X		X	X	X				

DTD of the Generated Output

```

<!ELEMENT Command (Type, DeterminationDate?,
Owner, ((Network, NetworkVersion?) | (SymbolTable, SymbolTableVersion?))?,
(Owner2, ((Network2, NetworkVersion2?) | (SymbolTable, SymbolTableVersion2?)))?,
Job?, (Time, Run, Date)?, ScheduleType?, ActivationType?, Show?, Email?, Report)>
<!ATTLIST Command
  Object          CDATA #REQUIRED
  Name            CDATA #REQUIRED
>

<!ELEMENT Type (#PCDATA)>
<!ELEMENT DeterminationDate (#PCDATA)>

<!ELEMENT Owner (#PCDATA)>
<!ELEMENT Network (#PCDATA)>
<!ELEMENT NetworkVersion (#PCDATA)>
<!ELEMENT SymbolTable (#PCDATA)>
<!ELEMENT SymbolTableVersion (#PCDATA)>

<!ELEMENT Owner2 (#PCDATA)>
<!ELEMENT Network2 (#PCDATA)>
<!ELEMENT NetworkVersion2 (#PCDATA)>
<!ELEMENT SymbolTable2 (#PCDATA)>
<!ELEMENT SymbolTableVersion2 (#PCDATA)>

<!ELEMENT Time (From, Thru)>
<!ELEMENT Run (From, Thru)>
<!ELEMENT Date (From, Thru)>
<!ELEMENT From (#PCDATA)>
<!ELEMENT Thru (#PCDATA)>
<!ELEMENT ScheduleType (#PCDATA)>
<!ELEMENT ActivationType (#PCDATA)>

<!ELEMENT Show (#PCDATA)>
<!ELEMENT Email (#PCDATA)>
<!ELEMENT Report (#PCDATA)>

```

Examples of Add Commands

```

<Command Object="Reports" Name="Add">
  <Type ShortType="LGT">Log - Terminated Jobs</Type>
  <Owner>EXAMPLE</Owner>
  <Network>*</Network>
  <Job>*</Job>
  <Time><From>000000</From><Thru>070858</Thru></Time>
  <Run><From>1</From><Thru>99999</Thru></Run>
  <Date><From>20131118</From><Thru>20131118</Thru></Date>
  <Email>noreply@softwareag.com</Email>
  <Report ID="3190043"/>
</Command>

<Command Object="Reports" Name="Add">
  <Type ShortType="NDS">Network Description (short)</Type>
  <DeterminationDate>20131118</DeterminationDate>
  <Owner>HEB</Owner>
  <Network>AA</Network>
  <NetworkVersion>*</NetworkVersion>
  <Email>noreply@softwareag.com</Email>
  <Report ID="3190050"/>
</Command>

<Command Object="Reports" Name="Add">
  <Type ShortType="NSO">Network Schedule Overview</Type>
  <Owner>EXAMPLE</Owner>
  <Network>*</Network>
  <Date><From>20131118</From><Thru>20131118</Thru></Date>
  <ScheduleType>A</ScheduleType>
  <Email>noreply@softwareag.com</Email>
  <Report ID="3190058"/>
</Command>

<Command Object="Reports" Name="Add">
  <DeterminationDate>20131118</DeterminationDate>
  <Type ShortType="ACO">Activation Overview</Type>
  <Owner>EXAMPLE</Owner>
  <Network>*</Network>
  <Date><From>20131118</From><Thru>20131118</Thru></Date>
  <ActivationType>*</ActivationType>
  <Email>noreply@softwareag.com</Email>
  <Report ID="3190060"/>
</Command>

<Command Object="Reports" Name="Add">
  <Type ShortType="CST">Compare Symbol Tables</Type>
  <Owner>EXAMPLE</Owner>
  <SymbolTable>E20-ST</SymbolTable>
  <SymbolTableVersion>*</SymbolTableVersion>
  <Owner2>EXAMPLE</Owner2>

```

```

<SymbolTable2>E20-ST-VSE</SymbolTable2>
<SymbolTableVersion2>*</SymbolTableVersion2>
<Show>A</Show>
<Email>noreply@softwareag.com</Email>
<Report ID="3190061"/>
</Command>

<Command Object="Reports" Name="Add">
  <Type ShortType="CNV">Compare Networks</Type>
  <Owner>EXAMPLE</Owner>
  <Network>B60-FLOWFX</Network>
  <NetworkVersion>*</NetworkVersion>
  <Owner2>EXAMPLE</Owner2>
  <Network2>B60-FLOW38</Network2>
  <NetworkVersion2>*</NetworkVersion2>
  <Show>A</Show>
  <Email>noreply@softwareag.com</Email>
  <Report ID="3190068"/>
</Command>

```

Command - Object: Reports - Name: Delete

The `Delete` command remove reports from the report list. Command is best used in conjunction with the `List` command. The `List` can be used to generate a `VariableArray` that contains the IDs of the report list which can be deleted with the `Delete` command.

The `<Parameter>` tag defines the parameter for the command:

Parameter	Format	Value	Explanation
ID	A	A FixedID, a VariableName, or a VariableArrayName.	ID of a report given as variable name or as fixed value. This parameter can be specified multiple times or <code>VariableArray</code> can be used.

Example 1:

```

<Command Object="Reports" Name="Delete">
  <Parameter Name="ID">3092265</Parameter>
  <Parameter Name="ID">3092267</Parameter>
  <Parameter Name="ID">3092268</Parameter>
  <Parameter Name="ID">3092271</Parameter>
</Command>

```

Output:

```
<Command Object="Reports" Name="Delete">
  <ReportDeleted ID="3092265"/>
  <ReportDeleted ID="3092267"/>
  <ReportDeleted ID="3092268"/>
  <ReportDeleted ID="3092271"/>
</Command>
```

Example 2:

```
<Command Object="Reports" Name="List">
  <Parameter Name="Filter_Status">Completed</Parameter>
  <Parameter Name="returnID">all_Completed_Reports</Parameter>
</Command>

<Command Object="Reports" Name="Delete">
  <Parameter Name="ID">%all_Completed_Reports(*)%</Parameter>
</Command>
```

Output:

```
<Command Object="Reports" Name="Delete">
  <Filter Name="Status">Completed</Filter>
  <ReportList ID="2520730">
    <Status>Completed</Status>
    <Type ShortType="NS0"> Network Schedule Overview</Type>
    <Creation><Date Format="G">20131205</Date><Time>103305</Time></Creation>
  </ReportList>
  <ReportList ID="2520731">
    <Status>Completed</Status>
    <Type ShortType="AC0"> Activation Overview</Type>
    <Creation><Date Format="G">20131205</Date><Time>103307</Time></Creation>
  </ReportList>
</Command>

<Command Object="Reports" Name="Delete">
  <ReportDeleted ID="2520730"/>
  <ReportDeleted ID="2520731"/>
</Command>
```

Command - Object: - Name: DeterminationDate

The `DeterminationDate` command modifies the determination date e.g. used during the reports generation.

The `<Parameter>` tag defines the parameter for the command:

Parameter	Format	Default	Value	Explanation
Date	D	current date	A valid date with format YYYYMMDD.	Specify the determination date.

Example:

```
<Command Name="DeterminationDate">
  <Parameter Name="Date">20131126</Parameter>
</Command>
```

Command - Object: - Name: Language

The Language command set the new application language. The language is e.g. important during report generation to get localized reports.

The <Parameter> tag defines the parameter for the command.

Parameter	Format	Default	Value	Explanation
Language	N2	Current Application Language	1 for English 2 for German	Same as Natural system variable *LANGUAGE. Note: Only left-to-right single-byte languages with Latin lower case

Example:

```
<Command Name="Language">
  <Parameter Name="Language">2</Parameter>
</Command>
```

Command - Object: - Name: Stop

The Stop command is used to terminate the execution of the batch client command xml file.

One or more Stop commands may be inserted.

Example:

```
<Command Name="Stop"/>
```

Variable Handling

The batch command client provides a simple parameter handling. Parameters with the prefix `return` provide the option to save the data as a variable. These variables can be used by other commands if a value accepts data from the type variable. If a parameter value is retrieved from a variable, the name of the variable has to be enclosed in percent (%) signs. If a variable contains a list, the variable name must end with (*).

Example to list all reports with status Completed and then delete all these reports:

```
<Command Object="Reports" Name="List">
  <Parameter Name="Filter_Status">Completed</Parameter>
  <Parameter Name="returnID">all_Completed_Reports_List</Parameter>
</Command>

<Command Object="Reports" Name="Delete">
  <Parameter Name="ID">%all_Completed_Reports_List(*)%</Parameter>
</Command>
```

In the example above, the command `List` uses the parameter `returnID` to return a list of report IDs. The command `Delete` accepts the list of report IDs for the `ID` parameter.

Example to add a report, wait for generation completion and save as file:

```
<Command Object="Reports" Name="Add">
  <Parameter Name="ShortType">LGT</Parameter>
  <Parameter Name="Owner">EXAMPLE</Parameter>
  <Parameter Name="returnID">new_ID_LGT</Parameter>
</Command>

<Command Object="Reports" Name="List">
  <Parameter Name="WaitID " ">%new_ID_LGT%</Parameter>
</Command>

<Command Object="Reports" Name="SaveToFile">
  <Parameter Name="ID">%new_ID_LGT%</Parameter>
  <Parameter Name="OutputType">HTML</Parameter>
  <Parameter Name="FileName">c:\LGT.html</Parameter>
</Command>
```

Explanation:

1. The command `Add` creates a report of the type `LGT` for the owner `EXAMPLE`. The parameter `returnID` returns the ID for the report and fills the variable `new_ID_LGT`.
2. The command `List` uses the parameter `WaitID` required to ensure that the report is completely generated before the data is written to the file. `WaitID` uses the same variable as `returnID` to select the previously generated report.
3. The command `SaveAsFile` writes the generated report to an HTML file. The parameter `ID` uses the same variable as `returnID` to select the same report previously generated.

Relative Date Feature

For parameters of type date a special feature for relative time setting is available. Based on the current date, the relative date feature gives you the possibility to specify dates in past or future.

Instead of specifying a specific date (N8, Format YYYYMMDD) an offset $+n$ or $-n$ days can be specified, where n is of type (I2).

Example:

```
<Command Object="Reports" Name="List">
  <Parameter Name="Filter_Date">-1</Parameter>
  <Parameter Name="returnID">all_Yesterday_Reports</Parameter>
</Command>

<Command Object="Reports" Name="Delete">
  <Parameter Name="ID">%all_Yesterday_Reports(*)%</Parameter>
</Command>

<Command Object="Reports" Name="Add" DeterminationDate="+7">
  <Parameter Name="ShortType">NS0</Parameter>
  <Parameter Name="Owner">EXAMPLE</Parameter>
  <Parameter Name="returnID">new_Next_Week_NS0</Parameter>
  <Parameter Name="ScheduleType">schedule network time</Parameter>
</Command>
```

V Using Owners

10 Using Owners

- Available Functions: Owner 164
- Owner at Logon 165
- Linking Additional Owners 165
- Owners Granted Access to Individual Networks 166

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.

Available Functions: Owner

➤ To list all available functions for the Owner metanode

- In the object workspace, select the **Owner** metanode and open the context menu.

The context menu provides the following functions:

Function	Shortcut	Explanation
List	F8	Lists all owners linked to your user ID.
Refresh	F5	See <i>Refreshing Object Lists</i> .
Filter	F3	Selection criteria for listing network masters: see <i>Filtering Objects</i> .
Export	---	Export Objects window to export all items of the metanode Owner : see <i>Exporting Objects</i> in the <i>Import/Export Functions</i> documentation.
Set Drag And Drop Function	---	See <i>Drag & Drop</i> .

➤ To list all available functions for an Owner instance

- In the object workspace, select an owner from the **Owner** metanode and open the context menu.

The context menu provides the following functions:

Function	Explanation
Next Activations	See Displaying Next Network Starts - Next Activations in the section <i>Schedule Maintenance</i> .
List Active Jobs	See List Active Jobs .
Browse Log	See Browsing the Log .
Export	Opens the Export Objects window to export an owner: see <i>Exporting Objects</i> in the <i>Import/Export Functions</i> documentation.
Set Drag And Drop Function	See Drag & Drop .

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

- On the **Main** page of the **Maintenance User** window (see the *Administration* documentation), replace the name of the owner entered in the **Owner at logon** box.

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

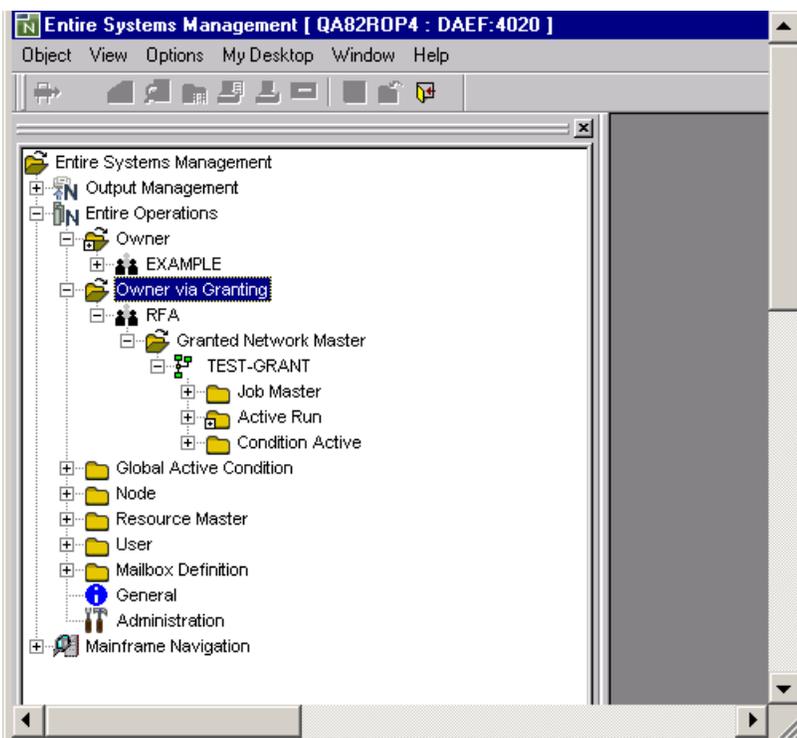
You can grant a user or an owner unrestricted or restricted access to a selected network by using the **Granting Definition** option of the network maintenance function described in [Granting Definition: Authorizing Other Users or Owners to Access a Network](#) in the section *Network Maintenance*.

Listing Granted Owners

The **Owner via Granting** metanode lists all owners who granted network access for your user ID.

The hierarchy below the **Owner via Granting** metanode shows networks granted to your user ID or to an owner linked to your user ID if the owner of these networks is not linked to your user ID. A granted owner who is also linked in your user profile is only listed in the **Owner** node.

All of these networks are separated in the tree view hierarchy by their respective owner as shown in the example below:



The context menus of the **Owner via Granting** nodes provide the following functions:

Function	Shortcut	Explanation
List	F8	Depending on the node selected, lists all owners or objects contained in the node.
Refresh	F5	See <i>Refreshing Object Lists</i> .
Filter	F3	Selection criteria for listing granted network masters: see <i>Filtering Objects</i> .
Set Drag And Drop Function	---	See <i>Drag & Drop</i> .

The functions available for a granted network master correspond to the functions available for a network master: see *Available Functions: Network Master*.

VI 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

Viewing and Maintaining a Job Network Diagram

Checking for a Loop in a Job Network

Maintaining the Usage of Network Versions

Applying Network Defaults to Jobs (Mass Update)

Activating a Job Network Manually

11

Maintaining Job Network Definitions

▪ Listing all Network Definitions	172
▪ Available Functions: Network Master	173
▪ Displaying a Network Definition	175
▪ Modifying a Network Definition	175
▪ Adding a Network Definition	178
▪ Fields: Network Definition	179
▪ OS Specials - Operating System and Environment Defaults	181
▪ Message and Message Recipients - Specifying Recipients for Network Messages	186
▪ Granting Definition: Authorizing Other Users or Owners to Access a Network	187
▪ Long Description - Documenting Your Networks	191
▪ Deleting a Job Network	192

Note: The **Active Run** subnode of the **Network Master** node correlates with the **Active Run** subnode of the **Network Active** node. Both subnodes control the same active runs. If you use both subnodes to change active runs during a session, consider refreshing them manually to make sure that you view the latest status.

Listing all Network Definitions

➤ To list all networks defined for an owner

- In the object workspace, select the **Network Master** node of an owner instance and choose **List** from the context menu.

Or:

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

A **Network Master List** window similar to the example below opens:

Network	Loop	Node	Description
B60-FLOW		55522	BS2000 Job Flow
E01-CONTI		148	Completion-Codes, Job Duration
E02-IOC-01		148	I/O Conditions without events
E03-IOC-02		148	I/O Conditions with code events
E04-IOC-03		148	I/O Conditions with string events
E05-IOC-04		148	I/O Conditions with user routine event
E10-PAR-01		148	Parallel & sequential job execution
E20-DYN-01		148	Dynamic JCL generation
E40-REC-01		148	Cause a recovery situation
E40-REC-02		148	Recovery and Reschedule
E50-USRT		148	User Routine Usage
E51-MAIL		148	Mailbox Usage
E52-RES		148	Resources

The window lists all networks defined for the selected owner.

For further information, see:

- [Columns: Network Master List](#)
- [Available Functions: Network Master](#)

Columns: Network Master List

The **Network Master List** window contains the following columns:

Column	Description
Network	User-defined network name.
Loop	blank column No loop was found during a loop test for the corresponding network.
	Yes A loop was found during a loop test for the corresponding network.
Node	Default execution node for the jobs in the network.
Description	Short description of the network.

Available Functions: Network Master

➤ To list all available functions for the Network Master metanode

- In the object workspace, select a **Network Master** metanode and open the context menu.

The context menu provides the following functions:

Function	Shortcut	Description
List	F8	See <i>Listing All Network Definitions</i> .
New	CTRL+N	Creates a new network master instance: see <i>Adding a Network Definition</i> .
Refresh	F5	See <i>Refreshing Object Lists</i> .
Filter	F3	Selection criteria for listing network masters: see <i>Filtering Objects</i> .
Paste data	CTRL+V	See <i>Copying Job Network Definitions; Cloning Network Versions</i> .
Export	---	See <i>Exporting Objects</i> in the <i>Import/Export Functions</i> documentation.
Set Drag And Drop Function	---	See <i>Drag & Drop</i> .

➤ To list all available functions for a Network Master instance

- In the object workspace, select a **Network Master** instance and open the context menu.

The context menu provides the following functions:

Function	Shortcut	Description
Open	CTRL+O	Modifies the selected network definition.
Display	CTRL+D	Displays the selected network definition.
Diagram	---	Displays an overview of the job flow within this network. See Viewing and Maintaining a Job Network Diagram .
Delete	DELETE	Deletes the selected network , including all jobs and all other definitions for this network.
Copy data	CTRL+C	Copies all definitions of the selected network to paste them as a new network: see Copying Job Network Definitions; Cloning Network Versions .
Copy Defaults	---	Copies network attribute definitions of the selected network as the default for all jobs contained in the network: see Applying Network Defaults to Jobs (Mass Update) .
Activate Network	---	Activates the selected network manually.
History	---	Displays the execution history (previous network runs).
Display schedule	---	Displays the schedule for a job network.
Next Activations	---	See Displaying Next Network Starts - Next Activations in the section <i>Schedule Maintenance</i> .
Usable Symbol Tables	---	Opens the Usable Symbol Tables window to view or modify all available symbol table definitions. See also Listing Usable Symbol Tables in the section <i>Symbol Table and Symbol Maintenance</i> .
Check for loop	---	Checks for loops in the network.
Version Usage	---	Maintains the usage of network versions.
List Active Jobs	---	Lists active jobs defined for this network: see Listing Active Jobs .
Browse Log	---	See Displaying Logged Information - Browse Log Function .
Export	---	See Exporting Objects in the <i>Import/Export Functions</i> documentation.
Add to Workplan	---	Stores a reference to the currently selected object in a list of activities to be done. See also Add to Workplan .
Set Drag And Drop Function	---	See Drag & Drop Function .

Displaying a Network Definition

Displays job value, operating system special, symbol prompting and granting definition information for the currently selected Network Master.

The **Display** function invokes a dialog that displays the information for the currently selected network definition. You can only view the information, not edit it. If you want to edit the item, you have to use the **Open** function.

» To display a network definition

- 1 In the object workspace, select an owner from the **Owner** node and then a network definition from the **Network Master** node.
- 2 Invoke the context menu and choose the **Display** function.

Or:

Press CTRL+D.

A **Network Master** window opens that displays all available information for the selected network definition.

The fields and tabs of the **Network Master window** are explained in *Fields: Network Definition*.

Modifying a Network Definition

Opens job value, operating system special, symbol prompting and granting definition information for the currently selected network master in editing mode.

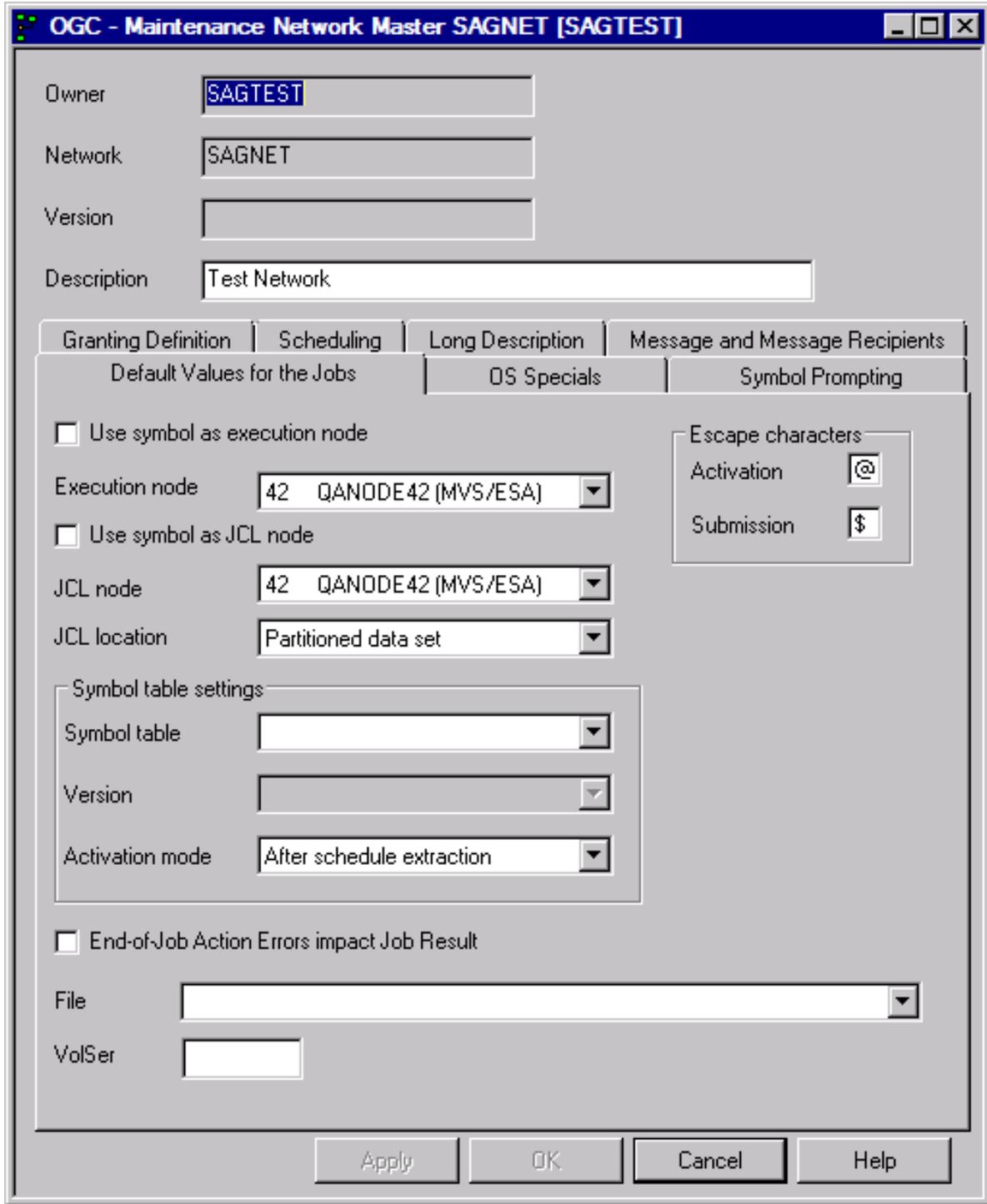
» To modify a network definition

- 1 In the object workspace, select a **Network Master** instance.
- 2 Invoke the context menu and choose the **Open** function.

Or:

Press CTRL+O.

A **Maintenance Network Master** window similar to the example below opens:



3 Now you can edit the fields you want to change.

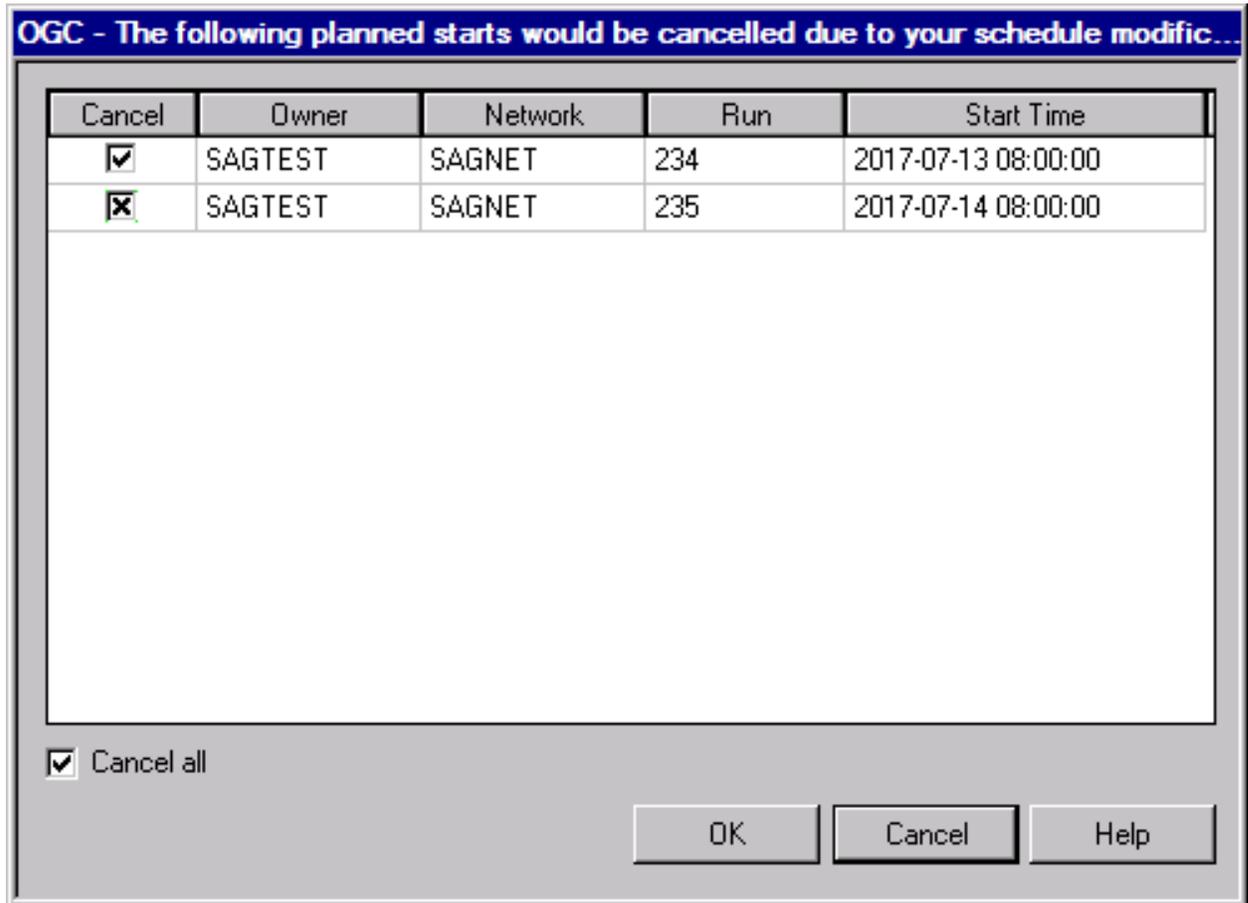
The fields and tabs of the **Maintenance Network Master** window are explained in [Fields: Network Definition](#).

4 Select **OK**.

Your changes are saved.

If your modification of a network schedule affects any planned starts, you have the option to cancel the starts or to keep them active.

If you select **Cancel**, a dialog similar to the example below appears:



You can select the following functions:

1. Cancel all starts - by checking the **Cancel all** box.
2. Cancel selected – by selecting certain planned runs in the table.
3. Do not cancel – by selecting the **Cancel** button. (Keeps all listed starts active, regardless of your schedule modifications).

Adding a Network Definition

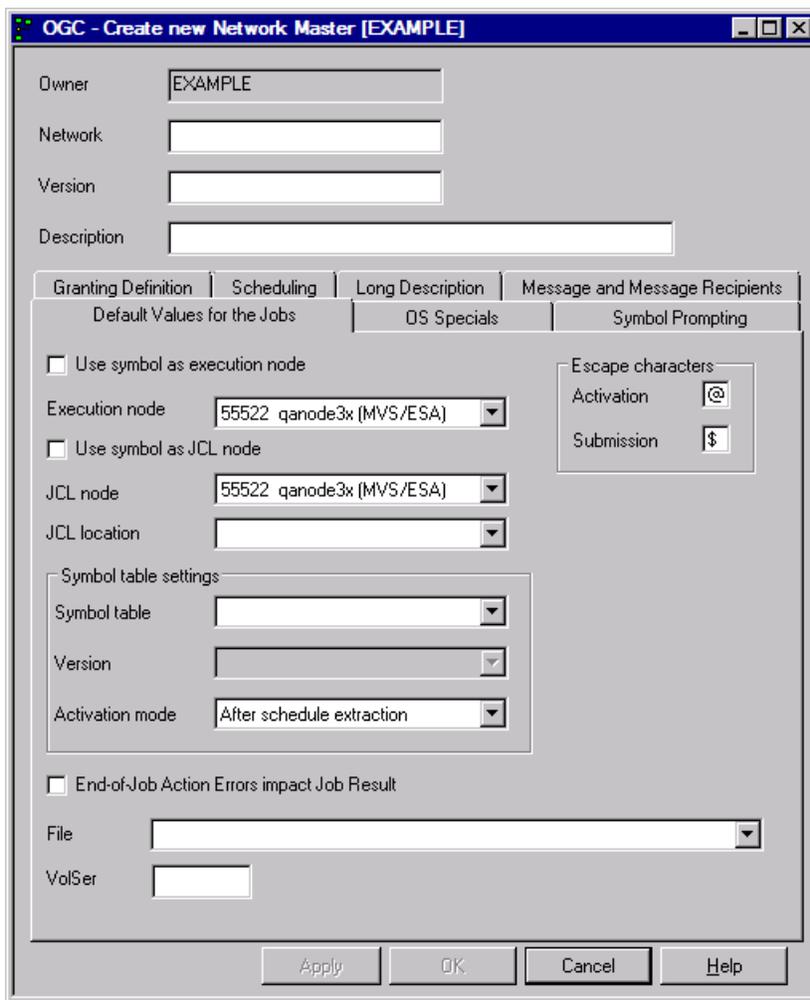
➤ To create a new network master instance

- 1 In the object workspace, select a **Network Master** instance.
- 2 Invoke the context menu and choose the **New** function.

Or:

Press CTRL+N.

A **Create new Network Master** window similar to the example below opens:



- 3 Now, you can insert the missing information into the fields.

The fields and tabs of the **Create new Network Master** window are explained in [Fields: Network Definition](#).

- 4 Select **OK**.

Your changes are saved.

Fields: Network Definition

The fields and tabs of a **Network Master** window are described in the following table:

Field/Tabbed Page	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 in the Network List window.</p> <p>A longer description of the network can be added using the editor (see Documenting Your Networks).</p>
Default Values for the Jobs	The input fields on this tabbed page 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.
Use symbol as execution node	Mark this option if you want to use a suffix symbol: see the field Suffix symbol in the section <i>Job Maintenance</i> .
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. The drop-down list box shows all nodes available for selection. The operating system type appears after a valid node number.
Use symbol as JCL node	Mark this option if you want to use a suffix symbol: see the field Suffix symbol in the section <i>Job Maintenance</i> .
JCL node	<p>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.</p> <p>The context menu of the selected node is the same as for Execution node.</p>

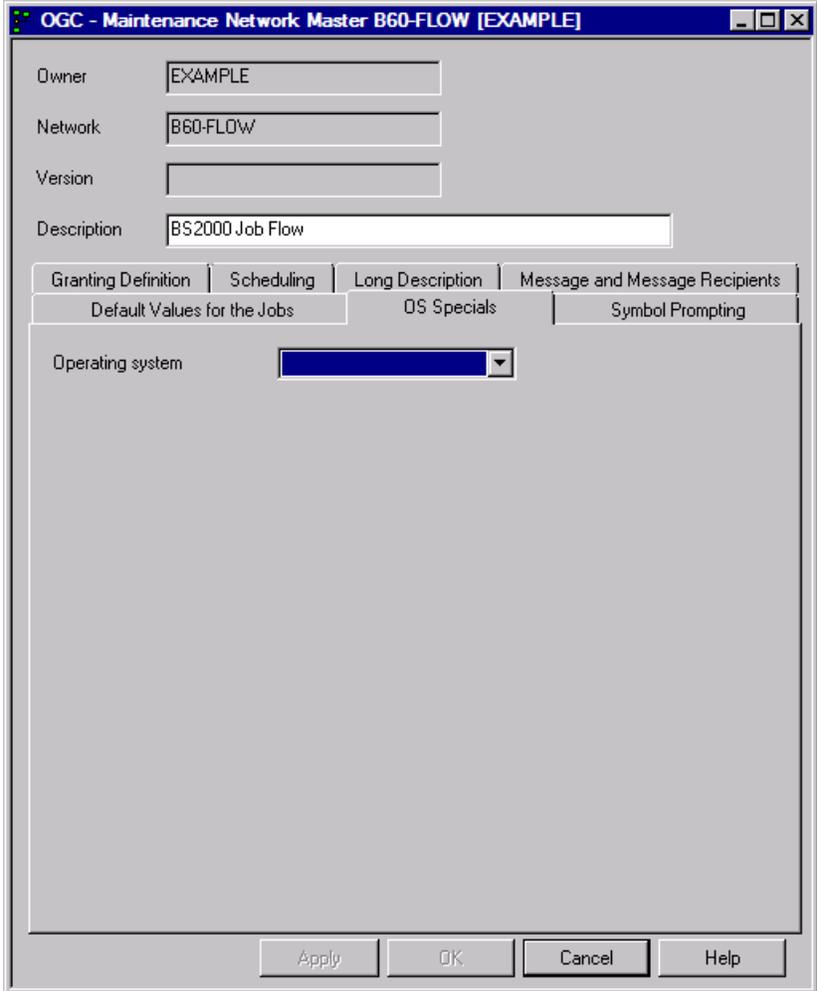
Field/Tabbed Page	Description
JCL location	<p>Type of storage for the JCL: see List of JCL Locations.</p> <p>The default is used in new job definitions and can be overridden there.</p>
Symbol table	<p>Default symbol table for those jobs in the network that use the dynamic JCL generation facility. Can be overridden at the job level and is therefore optional here. The drop-down list box shows all symbol tables available for selection.</p>
Version (symbol table)	<p>You can maintain several versions of a symbol table. The drop-down list box shows all symbol table versions available for selection.</p> <p>Reserved names:</p> <p><i>blank</i> Unknown (implicit) version.</p> <p>(current) Is replaced by the current version for the activation date or determination date.</p> <p>(nv) 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>
Activation mode (symbol table)	<p>From the drop-down list box you can choose either of the following:</p> <ul style="list-style-type: none"> ■ During the network activation. <p>No symbol prompting is possible.</p> <ul style="list-style-type: none"> ■ After schedule extraction. <p>Symbol prompting can be used for scheduled networks. This is the default.</p>
Activation (escape characters)	<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>
Submission (escape characters)	<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	<p>If you select this check box, 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 <code>not ok</code>. If the job was already set to <code>not ok</code>, it will remain <code>not ok</code>, regardless of the definition here.</p>

Field/Tabbed Page	Description
Impact Job Result	<p>Note:</p> <ol style="list-style-type: none"> 1. Setting this flag may 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. <p>If the check box is not selected, 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.</p>
File	<p>Name of the file or Natural library according to the value of the JCL Location field.</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>
VolSer	Volume serial number of data set (if data set is not cataloged).
OS Specials	This tabbed page is described in <i>OS Specials - Operating System and Environment Defaults</i> .
Symbol Prompting	This tabbed page is described in <i>Symbol Prompting during Network or Job Activation</i> in the section <i>Symbol Table and Symbol Maintenance</i> .
Granting Definition	This tabbed page is described in <i>Granting Definition: Authorizing Other Users or Owners to Access a Network</i> .
Scheduling	This tabbed page is described in <i>Scheduling a Network</i> in the section <i>Schedule Maintenance</i> .
Long Description	This tabbed page is described in <i>Long Description - Documenting Your Networks</i> .
Message and Message Recipients	This tabbed page is described in <i>Message and Message Recipients - Specifying Recipients for Network Messages</i> .

OS Specials - Operating System and Environment Defaults

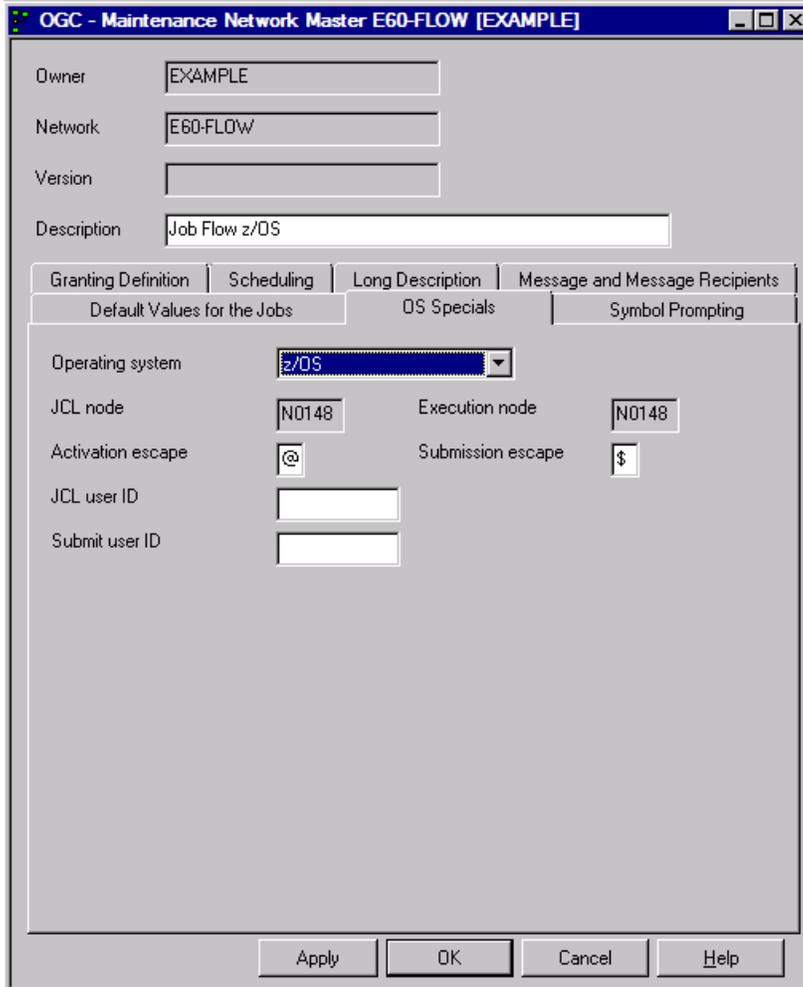
➤ To specify operating system specific defaults

- 1 Select the **OS Specials** tab.



2. Select the required operating system.

Depending on the operating system/environment selected, different input fields are provided on the page as shown in the example of z/OS below:



For information on all fields available, see [Input Fields: Network Defaults](#).

- 3 Select **OK**.

This section covers the following topic:

Input Fields: Network Defaults

The input fields available on the tabbed page **OS Special** depend on the operating system selected as indicated in the following table:

Field	Description
Operating system	Operating system/environment: BS2000, z/OS, z/VSE, UNIX/Windows or SAP R/3.

Field	Description
Activation escape	<p>Activation Escape Character = Network default value specific for operating system</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>
Submission escape	<p>Submission Escape Character = Network default value specific for operating system</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>
JCL User ID	<p>(Not applicable to z/VSE)</p> <p>The user ID to be used for JCL loading</p> <hr/> <p>BS2000:</p> <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> <hr/> <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> <hr/> <p>UNIX and Windows:</p> <p>With the authorization of this user ID, the Entire Operations Monitor loads the JCL of type TXT.</p> <hr/> <p>See also the default setting User ID Definition (<i>Defaults for Other Settings, Administration</i> documentation), <i>Operating System User IDs</i> and <i>Default User ID Determination</i>.</p>
Submit User ID	<p>(Not applicable to z/VSE)</p> <p>The user ID to be used at job start</p> <hr/> <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>

Field	Description
	<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: 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: 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>Defaults for Other Settings, Administration</i> documentation), Operating System User IDs and Default User ID Determination.</p>
Applies to BS2000 only:	
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	<p>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.</p> <p>See also the default setting User ID Definition (<i>Defaults for Other Settings, Administration</i> documentation), Operating System User IDs and Default User ID Determination.</p>
SYSOUT Cat ID	<p>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.</p>
Applies to UNIX and Windows only:	
JCL Group	<p>Applies to UNIX only (optional)</p> <p>If this field is left blank, the standard group of the UNIX user ID is used as defined under <code>/etc/passwd</code>. Otherwise, this field must contain one of the groups issued by the UNIX command groups.</p>

Field	Description
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 <code>/etc/passwd</code> . Otherwise, this field must contain one of the groups issued by the UNIX command <code>groups</code> .
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).

Message and Message Recipients - 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

- 1 Select the **Message and Message Recipients** tab.

OGC - Maintenance Network Master B60-FLOW [EXAMPLE]

Owner: EXAMPLE

Network: B60-FLOW

Description: Job Flow, BS2000

Default Values for the Jobs | OS Specials | Symbol Prompting

Granting Definition | Scheduling | Long Description | Message and Message Recipients

Text: (Network-related Messages)

Destination	Type	Node
SN		N0146 EOR Dev F-MC (MVS/ES)

Buttons: Add, Delete, Clear All

Buttons: Apply, OK, Cancel, Help

Insert a text and a destination for the message.

For further information, refer to [Fields and Columns: Message and Message Recipients](#) in the section [Message Sending](#).

- 2 Select **OK**.

Granting Definition: Authorizing Other Users or Owners to Access a Network

Entire Operations allows you to authorize other users and owners to access a network that belongs to you.

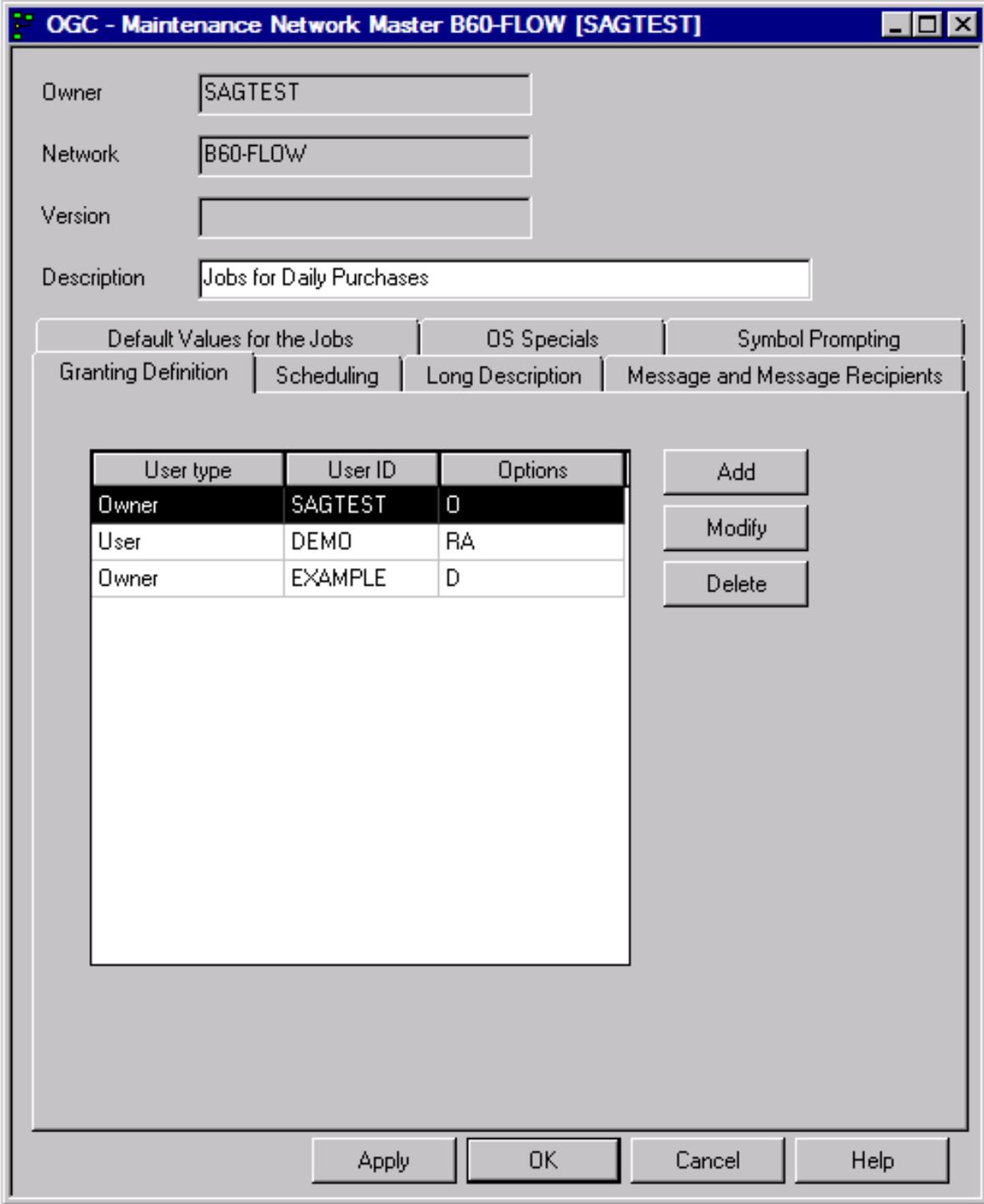
Users grouped under the owner SYSDBA have unlimited access to all networks in the system. They do not need the authorization described in this section.

Users and linked owners (lower than administrator) have limited access rights to granted networks in accordance with the rights specified in their user profiles and the restrictions specified in their individual network granting definition.

The **Owner via Granting** metanode lists all owners who granted network access for your user ID: see *Listing Granted Owners* in the section *Using Owners*.

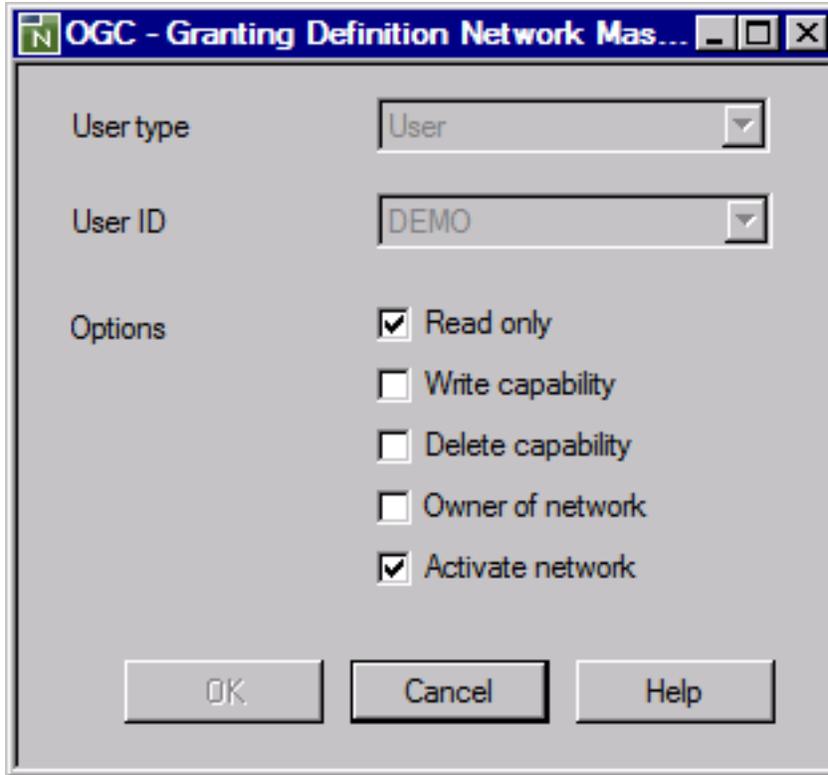
» **To add or modify a granting definition**

- 1 In the **Network Master** window, open the tabbed page **Granting Definition**.



- 2 Select a user type.
- 3 Select the **Add** button.

A **Granting Definition** window similar to the example below opens:



- 4 Specify the rights for the previously selected user. The fields and options available are described in *Fields: Granting Definition*.
- 5 Select **OK**.

» **To delete a granting definition**

- 1 On the **Granting Definition page**, select the table row that contains the user or owner whose granting definition you want to remove and choose **Delete**.

A window prompts you to confirm the deletion.

- 2 Choose **Yes** to confirm the deletion (**No** cancels the action) and close the window.
- 3 Choose **OK** to save your changes.

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 on the [Granting Definition page](#) and the corresponding fields and options in the [Granting Definition window](#) are described in the following table:

Column/Field	Description	
User Type	Specifies the object to be authorized. Possible options:	
	Owner	All users linked to the Entire Operations owner specified in the User ID field.
	User	A defined user.
User ID	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. You can select a user from the drop-down list box.	
Options	Level of access granted to a user or owner. You can select one or more of the following authorization levels:	
	Read only	Read access to the network (no maintenance).
	Write capability	Read and write access to the network (maintenance allowed except delete network).
	Delete capability	Read, write and delete access to the network.
	Owner of network	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.
	Activate network	User is allowed to activate the network.

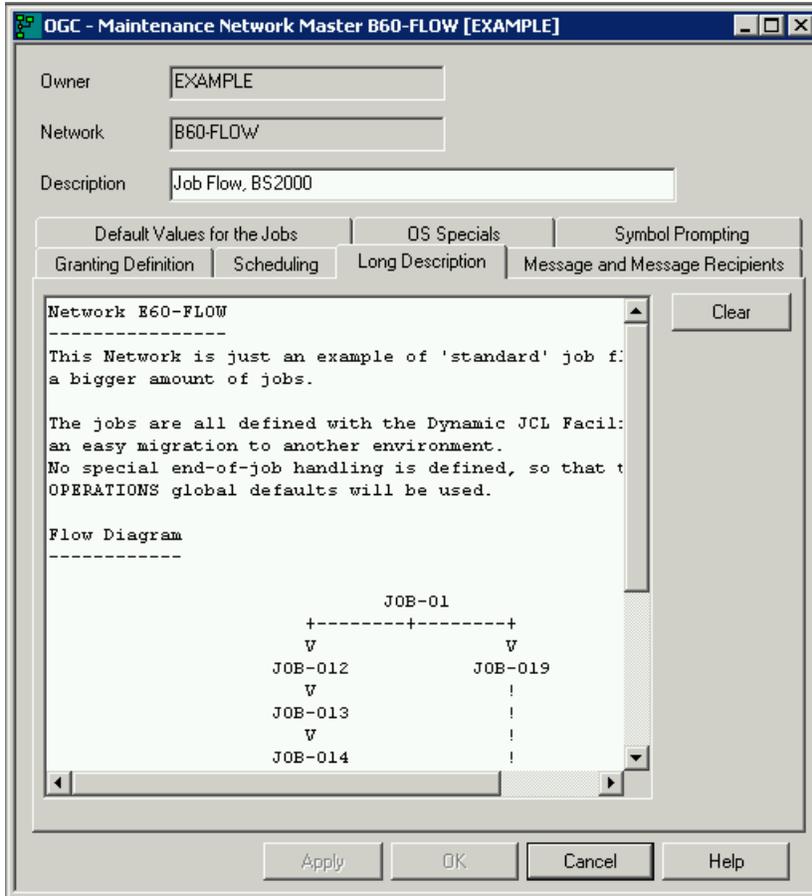
Long Description - Documenting Your Networks

You can add a [brief description](#) of a job network when defining a network in the **Network Master** window. This short description appears in the list of networks in the [Network Master List](#) window.

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 master**

- 1 Select the **Long Description** tab.



- 2 Write an explanation concerning the network master.
- 3 Select **OK**.

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 In the object workspace, select an owner from the **Owner** node and then a network definition from the **Network Master** node.

- 2 Invoke the context menu and choose the **Delete** function.

Or:

Press Delete.

A dialog prompts you to confirm the delete request.

- 3 Choose **Yes**.

If the delete is rejected, you receive an appropriate error message on the reason.

12 Copying Job Network Definitions; Cloning Network

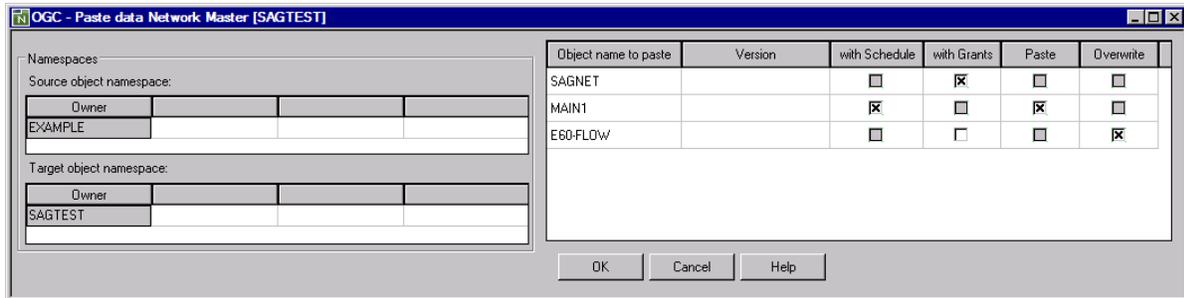
Versions

- Fields: Paste Data Network Master 197

➤ **To copy a network master**

- 1 From a **Network Master** node in the object workspace, select the network you want to copy and choose **Copy data** from the context menu, or press you CTRL+C.
- 2 Select the **Network Master** where you want to add the new network and choose **Paste data** from the context menu, or press CTRL+V.

A **Paste data Network Master** window similar to the example below opens:



The owner of the network(s) to be copied (here: EXAMPLE) and the owner of the **Network Master** node selected for copying (here: SAGTEST) are listed in the left-hand tables.

The network(s) to be copied are listed in the right-hand table. In the example above the network:

- SAGNET is not copied at all,
- MAIN1 is copied (including its schedule definition) if no network of the same name exists in the **Network Master** target node, and
- E60-FLOW is only copied if a network of the same name already exists in the target node.

- 3 Change the entries as required. The fields available in the window are described in [Fields: Paste Data Network Master](#).

When you are finished, choose **OK**.

The new network master(s) are copied to the selected **Network Master** metanode.

Fields: Paste Data Network Master

The fields of the **Paste data Network Master** 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	
Namespaces	The fields in the Namespaces section contain the name of the owner from which the new network is copied.	
Object name to paste	The name to be used for the new network.	
Version	The network version to be used for the new network. To clone a network version, make sure that owner names in the Source/Target object namespaces are identical, and enter a different network version.	
with Schedule	Possible check box settings:	
	<i>unchecked</i>	The schedule linked to the source network will not be copied (default).
	<i>checked</i>	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 day until the end of the current year. The execution history will not be copied.
with Grants	<i>unchecked</i>	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 Granting Definition: Authorizing Other Users or Owners to Access a Network .
	<i>checked</i>	Other users/owners granted access to the source network will be copied to the new network, in addition to the current user/owner.
Paste	<i>unchecked</i>	A new network is not pasted. See also Unchecked Paste and Overwrite .
	<i>checked</i>	Pastes the new network into the Network Master node of the owner entered in Namespaces (default).
Overwrite	<i>unchecked</i>	An existing network with the same name entered in Object name to paste will not be replaced (default). See also Unchecked Paste and Overwrite .
	<i>checked</i>	Replaces an existing network with the same name entered in Object name to paste .

Unchecked Paste and Overwrite

If neither the **Paste** nor the **Overwrite** check box is marked, the copy function is not executed for the respective target network. This is useful, for example, if you specified multiple networks in **Object name to paste** and want to skip the copy function for a single network.

See also the [example of a Paste data Network Master](#) window.

13

Viewing and Maintaining a Job Network Diagram

- Explanations of Diagram Symbols 200
- Maintenance Functions for Diagrams 204
- Editing and Navigating in the Network Diagram 209
- Examples of Diagrams 211

Diagrams are provided for instances of a network master or an active network/job run. Diagrams can be used to view and edit a defined network or define a new network.

 **Note:** You can customize the diagram view by changing the **Diagram** settings in your user profile described in the *Administration* documentation.

➤ To view a network diagram

- 1 In the object workspace, select a **Network Master** instance or an **Active Run** instance.
- 2 Open the context menu and choose the **Diagram** function.

Depending on the node instance selected, a **Diagram Network Master** or **Diagram Active Run** window (see *Examples of Diagrams*) opens with a graphical view of the selected network.

The symbols used to represent the network components are described in *Explanations of Diagram Symbols*.

You can view and rearrange all network components as described in *Maintenance Functions for Diagrams*.

Explanations of Diagram Symbols

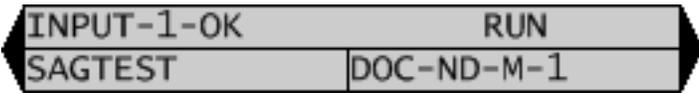
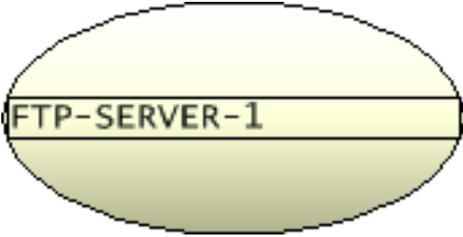
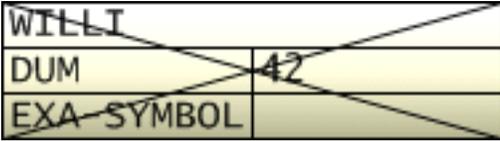
This section describes the symbols available in a diagram and indicates when a symbol only applies to an active job run.

The display of the symbols in the diagram graphic depends on whether the **Use new design** option is set (default) in the user profile (see the **Diagram** page described in *Administration* documentation). In general, the new design is characterized by gradient symbol colors and icons that identify specific reasons for an active job run that ended `not ok`.

The information provided with a diagram symbol can exceed the size available to display text. In addition, a complex network can contain very small symbols impossible to read without **zooming** because the size of the diagram is adjusted to the size of the network. Therefore, **tooltips** for annotations are provided for all symbols so you can always view the complete information in a readable form.

Grafiksymbol	Beschreibung						
<table border="1"> <tr> <td colspan="2">JOB-01</td> </tr> <tr> <td>JOB/C</td> <td>42</td> </tr> <tr> <td>EXAM-TABLE</td> <td></td> </tr> </table>	JOB-01		JOB/C	42	EXAM-TABLE		<p>Job.</p> <p>Übliche Einträge sind:</p> <ul style="list-style-type: none"> ■ Jobname (hier: JOB-01) ■ Jobtyp (hier: JOB)
JOB-01							
JOB/C	42						
EXAM-TABLE							

Grafiksymbol	Beschreibung																						
	<ul style="list-style-type: none"> ■ Job special type (here: C for a cyclic job) and/or milestone specification ■ Execution node number (here: 42) ■ Symbol table name (here: EXA-SYMBOL). 																						
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="2" style="text-align: left;">JOB-01</td> </tr> <tr> <td style="width: 70%;">JOB</td> <td style="text-align: center;">42</td> </tr> <tr> <td>EXA-SYMBOL</td> <td>MULT-01</td> </tr> </table>	JOB-01		JOB	42	EXA-SYMBOL	MULT-01	<p>Multiple suffix job (here: suffix symbol MULT-01 in the symbol table EXA-SYMBOL).</p>																
JOB-01																							
JOB	42																						
EXA-SYMBOL	MULT-01																						
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="2" style="text-align: left;">READ-DATA</td> </tr> <tr> <td style="width: 70%;">JOB/D/B</td> <td style="text-align: center;">146</td> </tr> <tr> <td> </td> <td> </td> </tr> </table>	READ-DATA		JOB/D/B	146			<p>Milestone job that performs as the first network job.</p> <p>In addition to the job special type, a milestone job has one or more (separated by slashes) of the following specifications:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="2">User-defined milestone:</td> </tr> <tr> <td style="width: 20%;">B</td> <td>Milestone job, network begin</td> </tr> <tr> <td>E</td> <td>Milestone job, network end</td> </tr> <tr> <td>I</td> <td>Milestone job, other</td> </tr> <tr> <td colspan="2">System milestone:</td> </tr> <tr> <td colspan="2">(used in active jobs only)</td> </tr> <tr> <td>1</td> <td>Milestone job, network begin</td> </tr> <tr> <td>2</td> <td>Milestone job, network end</td> </tr> </table>	User-defined milestone:		B	Milestone job, network begin	E	Milestone job, network end	I	Milestone job, other	System milestone:		(used in active jobs only)		1	Milestone job, network begin	2	Milestone job, network end
READ-DATA																							
JOB/D/B	146																						
User-defined milestone:																							
B	Milestone job, network begin																						
E	Milestone job, network end																						
I	Milestone job, other																						
System milestone:																							
(used in active jobs only)																							
1	Milestone job, network begin																						
2	Milestone job, network end																						
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="2" style="text-align: left;">PROCESS-1</td> </tr> <tr> <td style="width: 70%;">JOB/D/I</td> <td style="text-align: center;">146</td> </tr> <tr> <td> </td> <td> </td> </tr> </table>	PROCESS-1		JOB/D/I	146			<p>Milestone job that performs in no particular job order.</p>																
PROCESS-1																							
JOB/D/I	146																						
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="2" style="text-align: left;">DISTRIBUTE</td> </tr> <tr> <td style="width: 70%;">FTP/D/E</td> <td style="text-align: center;">146</td> </tr> <tr> <td> </td> <td> </td> </tr> </table>	DISTRIBUTE		FTP/D/E	146			<p>Milestone job that performs as the last network job.</p>																
DISTRIBUTE																							
FTP/D/E	146																						
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="2" style="text-align: left;">SUBNETJOB1</td> </tr> <tr> <td style="width: 70%;">NET</td> <td style="text-align: center;">42</td> </tr> <tr> <td>EXA-SYMBOL</td> <td> </td> </tr> </table>	SUBNETJOB1		NET	42	EXA-SYMBOL		<p>Job with type Subnetwork (NET), here named SUBNETJOB1.</p>																
SUBNETJOB1																							
NET	42																						
EXA-SYMBOL																							

Grafiksymbol	Beschreibung
	<p>Condition.</p> <p>Indicates the condition name (here: READ-DATA-OK) and the condition reference (here: RUN).</p>
	<p>External condition (new design) from another network.</p> <p>Common entries are:</p> <ul style="list-style-type: none"> ■ Condition name (here: INPUT-1-OK) ■ Condition reference (here: RUN) ■ Network owner (here: SAGTEST) ■ Network name (here: DOC-ND-M-1)
	<p>External condition (old design).</p>
	<p>Resource (here named FTP-SERVER-1).</p>
	<p>Reset condition.</p>
	<p>Set condition.</p>
	<p>Successfully ended job.</p>
	<p>Job waiting for execution.</p> <p>(active run only)</p>
	<p>Running job.</p> <p>(active run only)</p>

Grafiksymbol	Beschreibung
	<p>Job in hold or suspended after changes to the job definition.</p> <p>(active run only)</p>
	<p>Job ended not ok.</p> <p>(active run only)</p>
	<p>Job ended not ok for one of the following reasons:</p> <ul style="list-style-type: none"> ■ Job missed the latest start time and/or deadline defined in the Scheduling Parameters of the job. ■ Latest start time and/or deadline defined in the Network schedule does not allow job execution. ■ Job exceeded latest start time for one of the above reasons. <p>(shown in the new design and for the active job only)</p>
	<p>Job ended not ok because network activation was canceled by the user.</p> <p>(shown in the new design and for the active job only)</p>
	<p>Job ended not ok for any reason except and explained above.</p> <p>(shown in the new design and for the active job only)</p>

The background colors used for the symbols denote the following:

Color	Explanation
yellow	<p>The job waits for its execution. If the allowed execution time is exceeded, the job is considered as ended not ok and the symbol changes to red.</p> <p>(active run only)</p>
green	<p>The job is currently executing.</p> <p>(active run only)</p>

Color	Explanation
light green	The active job is affected by <ul style="list-style-type: none"> ■ an explicit hold function ■ an active regenerated JCL ■ a file/member that is changed in the active job definition. (active run only)
light yellow	The job ended ok.
red	The job ended not ok, for example, due to a JCL load error or an execution timeout. (active run only)
light grey	The condition is set.
dark grey	The condition is not set.

Maintenance Functions for Diagrams

This section describes the functions available in a network diagram and indicates whether a function can only be used in a master network (master only) or an active run.

The following functions are available within the free space area of the diagram:

Function	Description
Arrange	Rearranges the objects of the diagram automatically.
Refresh	Refreshes the presentation of the diagram. You can also use F5.
Auto Refresh	You can: <ul style="list-style-type: none"> ■ start/stop automatic refresh ■ change the interval of automatic refresh <p>Note: Changes to the refresh settings remain valid for the diagram in the active window only. For persistent changes, modify the Automatic Refresh settings in your user profile (see the <i>Administration</i> documentation).</p>
Object Filter	You may switch on/off: <ul style="list-style-type: none"> ■ Condition nodes ■ Resource nodes <p>Note: Changes to the filter settings remain valid for the diagram in the active window only. For persistent changes, modify the Filter settings in your user profile (see the <i>Administration</i> documentation).</p>

Function	Description
Tooltip	<p>You can customize the information displayed in tooltip:</p> <ul style="list-style-type: none"> ■ For Job it is possible to switch the display on/off for Name, Type, Status (only in active run), Node, Symbol Table, Suffix symbol, Subnetwork, JobID (only in active run). ■ For Condition it is possible to switch the display on/off for Name, Status (only in active run), Reference, Type, Suffix symbol, External network. ■ For Resource it is possible to switch tooltip on/off. ■ For Connector it is possible to switch the display on/off for source and target. <p>Note: Changes to the tooltip settings remain valid for the diagram in the active window only. For persistent changes, modify the Diagram settings in your user profile (see the <i>Administration</i> documentation).</p>
Find	<p>Opens a search dialog where you can enter a search string. You can also use CTRL+F.</p> <p>See also To search for a string.</p>
Find Next	<p>Finds the next occurrence of a search string entered in the search dialog. You can also use F3.</p> <p>See also To search for a string.</p>
Create Job	<p>Adds a new job to the network. A Create new Job Master (see Adding a Job Definition) or Create new Job Active (active run only; see Adding a New Job to the Active Network) window opens to add a new job to the network.</p>
Create Condition (master only)	<p>Adds a new condition. Proceed as described in To add and link a condition.</p>
Create Active Condition (active run only)	<p>Opens the Condition Active window to add a new active condition.</p>
Paste (master only)	<p>Paste a job object in the current network. See Pasting Objects.</p>
Browse Log	<p>Opens the Entire Operations log for the network or active run shown in the active diagram window or a symbol selected in the active diagram. The log only contains data relevant for this network or run.</p> <p>See also the Browse Log function that can be performed on a selected node in the tree view of the object workspace.</p>
Page Setup	<p>Opens a dialog where you can specify the page setup for printing the current diagram.</p>
Printer Setup	<p>Opens a dialog where you can specify your printer settings.</p>
Print Preview	<p>Opens a print preview where you can check if the diagram is laid out for printing according to your preferences.</p>
Print	<p>Opens a dialog where you can send the diagram to a printer.</p>

Function	Description
Zoom	<p>You have the following possibilities to zoom the diagram:</p> <ul style="list-style-type: none"> ■ in: makes the diagram larger ■ out: makes the diagram smaller ■ selection: zooms in/out single or multiple diagram symbols selected in the diagram. You can select single or multiple elements with CTRL+LEFT MOUSE BUTTON or by dragging the left mouse button over a group of adjacent symbols. ■ complete: shows the complete diagram on the screen ■ original: shows the diagram in its original size
Show World View	Opens a separate window with a minimized view of the complete diagram for easy navigation in complex diagrams. The green frame outlines the current viewport in the associated diagram. You can move the green frame to change the viewport in the diagram.
Show Legend View	Opens a separate window with a minimized view of the symbols used in the diagram for a quick overview of the objects (for example, resources and conditions) used in the job network.
Export Graphic	Opens a dialog where you can save the diagram on your hard disk.

The following functions are available for a job symbol selected in the diagram:

Function	Description
Open	Displays the currently selected job (box) and opens the Job Master or Job Active (active run only) window. Here, you can perform changes on the job.
Display	Displays the currently selected job (box) in the Job Master or Job Active (active run only) window.
New	Creates a new job (box) and opens the Create new Job Master (see <i>Adding a Job Definition</i>) or Create new Job Active (active run only; see <i>Adding a New Job to the Active Network</i>) window.
Delete (master only)	Deletes the currently selected job (box). See <i>Deleting Objects</i> .
Copy data (master only)	Copies the currently selected job (box). See <i>Copying Objects</i> .
Paste data (master only)	Pastes the currently selected job (box). See <i>Pasting Objects</i> .
Usable Symbol Tables	Lists all symbol tables that can be used by the job. See also <i>Listing Usable Symbol Tables</i> in the section <i>Symbol Table and Symbol Maintenance</i> .

Function	Description
Deactivate (active run only)	You can cancel the planned activation of a job in a planned network run that is waiting for activation.
Hold (active run only)	If the job has already been submitted to the operating system, it will be held in the operating system.
Release (active run only)	Releases a job from hold status.
Cancel (active run only)	The active job is cancelled.
Resubmit (active run only)	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.
Reactivate (active run only)	Reactivates a job.
Browse Master JCL (master only)	Browses the master JCL. See Displaying Master JCL .
Browse Active JCL (active run only)	A window opens where, you can see 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.
Edit Master JCL (master only)	Edits the master JCL. See Editing Master JCL and Natural Sources .
Edit Active JCL (active run only)	A window opens where you can view and edit the active JCL of the job.
Regenerate Active JCL (active run only)	Regenerates the JCL while the job is in the active database.
Stop cyclic execution (active run only)	Stops execution of currently running cyclic job.
Modify Latest Start (active run only)	See Modifying the Latest Start Time for an Active Run in the section <i>Active Job Networks</i> .
Release edit lock (active run only)	Removes a lock from active JCL: see Release edit lock in the section <i>Active Job Networks</i> .
Pregenerate Active JCL	See Pregenerating Active JCL .

Function	Description
(master only)	
Remove pregenerated Active JCL (master only)	See Removing Pregenerated JCL .
Edit pregenerated JCL (master only)	See Editing Pregenerated JCL .
Browse SYSOUT (active run only)	A dialog opens where you can view the SYSOUT of an active job. See Viewing Job SYSOUT .
Extended Log / JCL (active run only)	A screen opens, where you can see the JCL which was submitted for this job.
Extended Log / SYSOUT (active run only)	A screen opens, where you can see the SYSOUT of the job.
SYSOUT Messages (active run only)	A screen opens, where you can see the SYSOUT messages of the job (z/OS only).
Activate Job (master only)	Opens the Activation dialog where you can activate the currently selected job (box).
Open Subnetwork (active run only)	Opens the subnetwork Network Master dialog for a subnetwork (Job with type Subnetwork).
Subnetwork diagram	Opens the Network diagram for a subnetwork (Job with type Subnetwork).
Zoom Subnetwork	Opens a list of jobs for a subnetwork (Job with type Subnetwork).
Waiting for (active run only)	See Displaying Prerequisites for Active Jobs: Waiting for in the section <i>Active Job Networks</i> .
List Active Jobs	See Listing Active Jobs .
Browse Log	Opens the log for the currently selected job (box). See Displaying Logged Information - Browse Log Function .
Export (master only)	See Exporting Objects in the <i>Import/Export Functions</i> documentation.
Add to Workplan	Adds an object to the workplan. See also Add to Workplan .

Connecting Lines

The following functions are available for the relationships (connecting lines) within the diagram:

Function	Description
Set Input Condition (active run only)	Sets the input condition.
Reset Input Condition (active run only)	Resets the input condition.
Delete (master only)	Deletes the currently selected relationships between the objects.
Open Network (master only)	Opens the corresponding network maintenance dialog (see Modifying a Network Definition) for an external input condition.
Network Diagram (master only)	Opens the corresponding network diagram and to position diagram view to the proper condition for an external input condition.
Open Input Condition (master only)	Opens the Condition dialog. Here, you can perform changes on the currently selected input condition.
Open Output Condition (master only)	Opens the Condition dialog. Here, you can perform changes on the currently selected output condition.

Editing and Navigating in the Network Diagram

This section provides instructions for using the diagram editor functions to maintain objects for a network and find a particular object in the diagram.

» To add and link a condition

- 1 From the context menu in the **Diagram Network Master** window, choose **Create Condition**.

A square symbol appears indicating the position where the new condition will be placed in the diagram.

- 2 In the free space of the diagram, left-click on the position where you want to place the new diagram symbol.

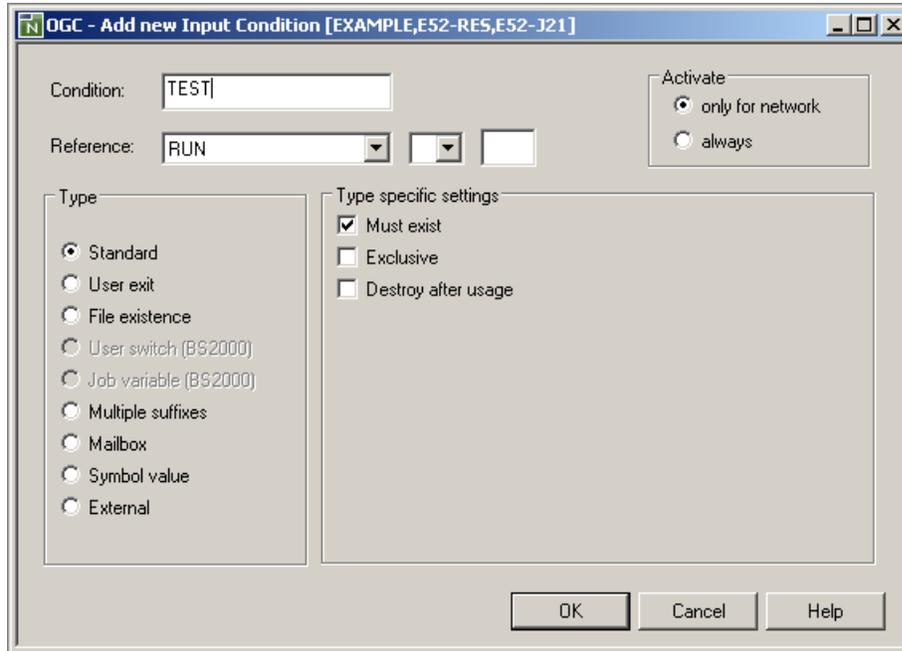
A **Create Condition** dialog appears.

- 3 Enter a valid condition name and choose **OK**.

The new condition is added to the diagram as a temporary (TMP) object, without any logical job links as shown in the tooltip information for the condition symbol.

- 4 Drag the selected condition and drop it onto the job to which you want to link the new condition as a logical input condition.

An **Add new Input Condition** window similar to the example below opens:



- 5 Enter the required definitions. The fields and options in the window are described in *Fields and Selection Options: Input Condition* in the section *Job Maintenance*.
- 6 When you are finished, choose **OK**.

The new condition is linked to the job as indicated by the connecting line between the job and the condition symbol and the tooltip information for the symbols.

➤ To create an output condition

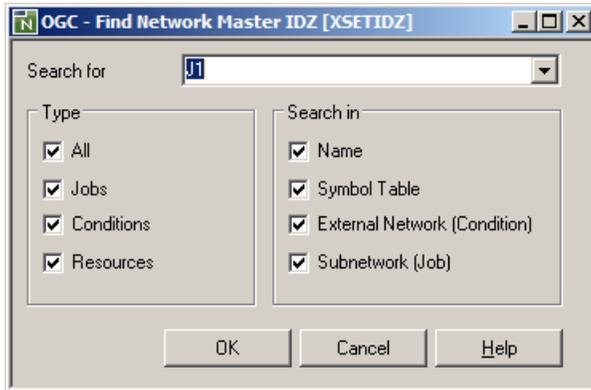
- 1 Select a job.
- 2 Drag the job and drop it onto a condition.
- 3 Select a branch (**All Checks ok** or **Any Check not ok**) to which the condition should be assigned.

➤ To link two jobs

- 1 Select a job.
- 2 Drag the job and drop it onto a job.
- 3 Select a branch (**All checks ok** or **Any check not ok**) to which the condition should be assigned.

➤ **To search for a string**

- 1 Call the search dialog by pressing CTRL+F or by choosing the **Find** function from the context menu:



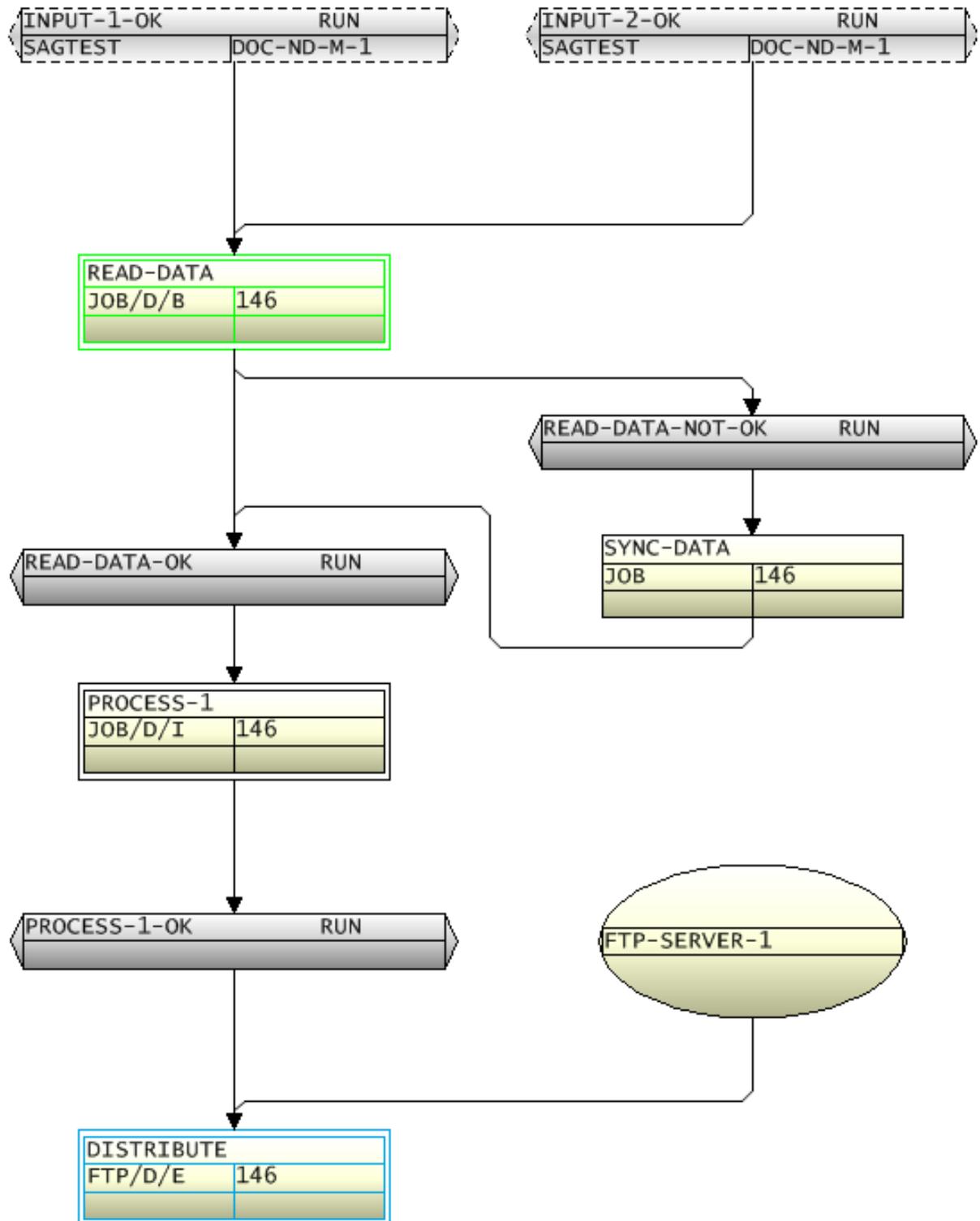
- 2 Fill in your search criteria. You can specify the search string, object type where the string is searched as well as the attribute of the object where to be searched. An asterisk ("*") is not interpreted as a wild card.
- 3 Choose **OK**.

Examples of Diagrams

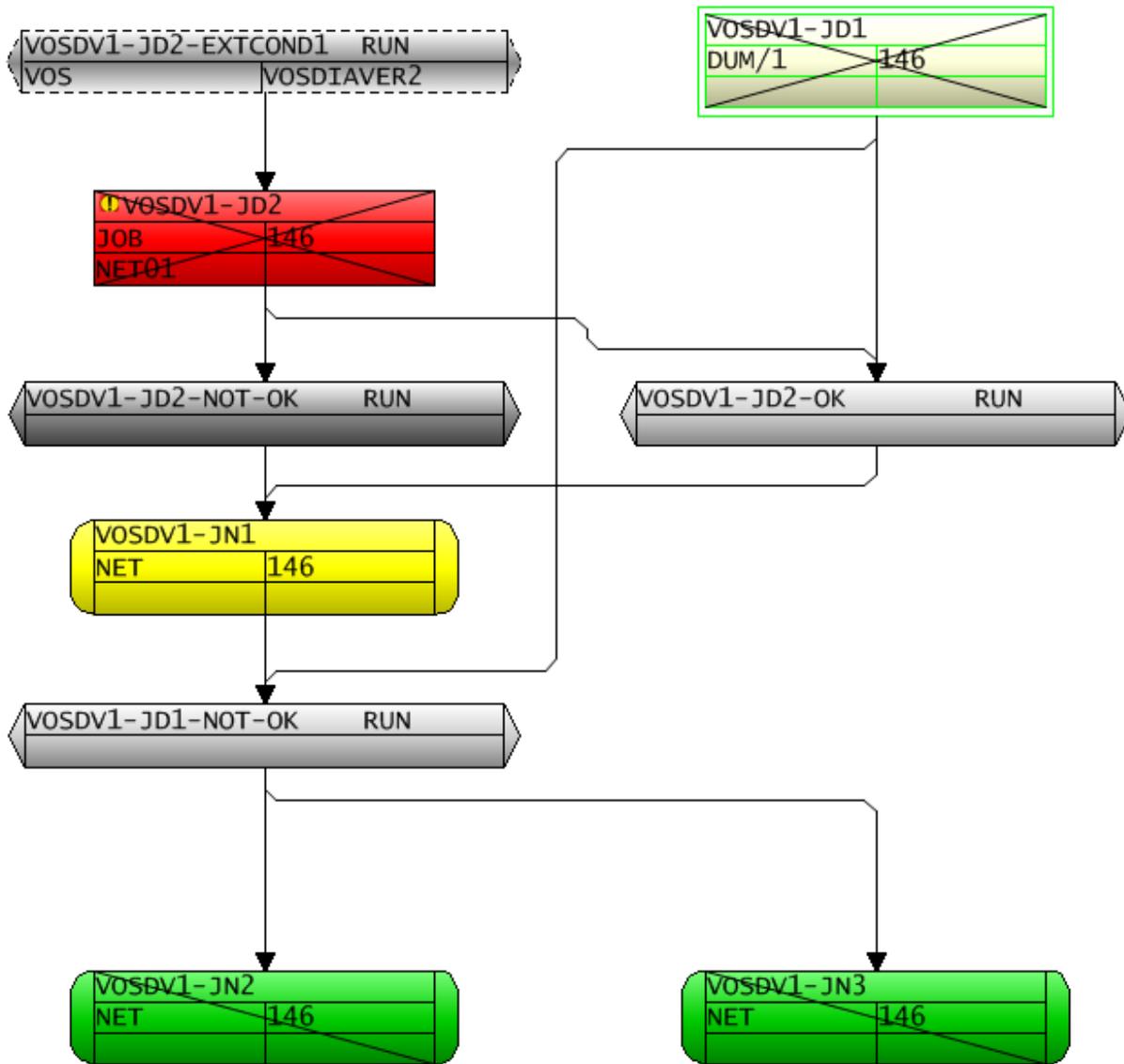
- [Example of a Diagram Network Master \(New Design\)](#)

- [Example of a Diagram Active Run \(New Design\)](#)

Example of a Diagram Network Master (New Design)



Example of a Diagram Active Run (New Design)



14

Checking for a Loop in a Job Network

» **To check for a loop in a network**

- 1 In the object workspace, select a **Network Master** instance.
 - 2 Invoke the context menu and choose **Check for loop**.
- 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:

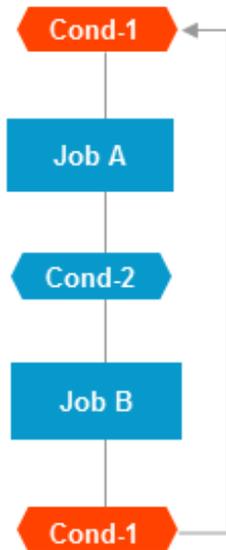
```
No Loop in Network xxx
```

- 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

- Handling Network Version Usage Definitions 218

Handling Network Version Usage Definitions

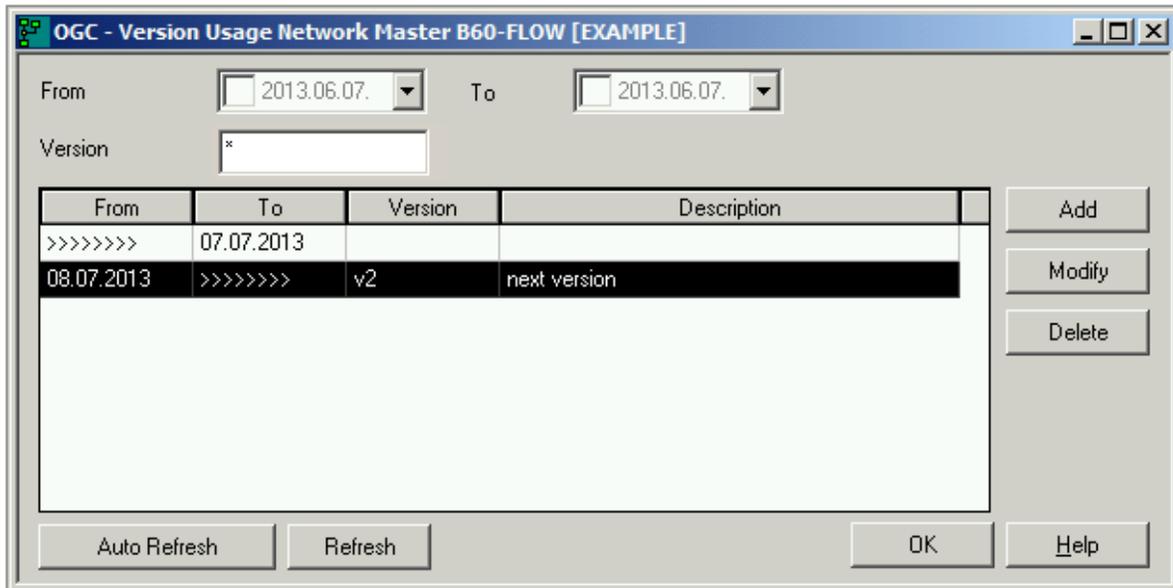
You can to define the maintenance of date ranges, in which network versions are to be used for scheduled activations.

The function Version Usage opens a list of date ranges, in which network versions are to be used for scheduled activations.

➤ **To list, add, modify or delete a network version usage definition**

- 1 In the object workspace, select a network from the **Network Master** node and choose **Version Usage** from the context menu.

A **Version Usage Network Master** window similar to the example below opens:



- 2 You can refresh the list by choosing the **Refresh** button.
- 3 You can modify or delete a defined version usage by selecting the respective row and choosing **Modify** or **Delete**, respectively.

Or:

You can add a version usage definition by choosing **Add**.

If you add or modify a defined network version usage, a window similar to the example below opens:

The screenshot shows a dialog box titled "OGC - Modify Network Version Usage". It contains the following elements:

- A "Network Version" dropdown menu set to "v2".
- A checkbox labeled "is to be used for scheduled activations" which is checked.
- A "From" date field set to "2013.07.08." with a checkmark icon to its left.
- A "To" date field set to "2013.06.07." with an unchecked checkbox icon to its left.
- A "Version Usage Description" text box containing the text "next version".
- Buttons for "OK", "Cancel", and "Help" at the bottom.

Add or modify the network version usage definition for scheduled activations as required. You can select a start or an end date only. If you omit the end date, the time is unlimited (infinite).



Note: The date ranges must not overlap.

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 In the object workspace, select a **Network Master** instance and choose **Copy Defaults** from the context menu.

A **Copy Defaults** window similar to the example below opens:

Copy	Attribute	Old	New
	Execution node	N0031	42
	JCL node	N0031	42
	JCL location		NAT
	File name		SYSEORU
	Symbol table		
	Symbol table version		
	JCL user ID		
	JCL group		
	Submit user ID		NOP
	Submit group		
	BS2000 default user ID		NOP
	BS2000 job class		
	BS2000 account number		
	SYSOUT user ID		NOP
	SYSOUT cat ID		
	Job priority		
	Run priority		
	Activation escape		@
	Submission escape		\$
	Modification user	(all)	NATQA5

The **Attributes** column lists all attribute definitions available for a network by default. The attribute values relevant for a mass update of jobs are listed in the **Old** and **New** columns.

All possible network attributes definitions are described in [Fields and Commands: Network Definition](#).

- From the drop-down list box in the **Copy** column, next to the attribute definition you want to copy, select one of the following options:

- **A: Apply to All**

The specified value overwrites the corresponding values in all jobs that belong to the network.

- **S: Apply to Specific**

Update only if the old value is the same as the old default value. This option ensures that no individual definitions are overwritten.

- If you do not want to copy a specified value, select the blank value from the list box.

If the **Copy** column is empty, no job modifications are performed.

- 3 When you are finished, choose **OK**.

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

Activating a Job Network Manually

- Fields: Network Activation 227

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 Master** node.

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

 **Note:** Only authorized users can activate job networks manually.

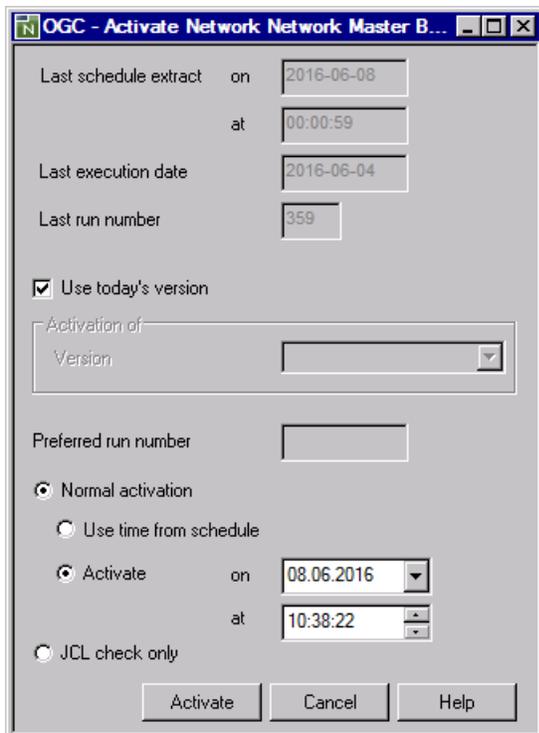
There are several situations in which you may wish to activate a job network manually:

- 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 In the object workspace, select a **Network Master** instance.
- 2 Invoke the context menu and choose **Activate Network**.

An **Activate Network** window similar to the example below opens:



3 Change the input fields as required. All fields contained in the window are explained in *Fields: Network Activation*.

4 Choose **Activate** when you are finished.

If symbol prompting has been defined for the network, a **Symbol Prompting window** opens 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*.

5 Choose **Continue** when you are finished.

If the activation is successful, the activated network gets a new run number as returned in a message, and its execution can be tracked in the GUI Client.

If a workplan has been defined for the network, a **Workplan window** opens with a list of objects accessed during the current Entire Operations session.

6 In the **Workplan window**, change the entries as required. For further explanations, see *Show Workplan* in the section *Using Entire Operations GUI Client*.

7 Choose **OK** when you are finished.

Fields: Network Activation

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

Field/Option	Description
Use today's Version	Version of the network to be activated.
Activation of Version	<p>If a current version exists for the current date, it will be preset as default value.</p> <p>If more than one version exists for a network, select Use today's Version (default) or Activation of Version as needed. If a version is not defined, then Activation of Version must be used.</p> <p>If no version for a network exists, then Use today's Version and Activation of Version have no function.</p>
Last Schedule Extract at	Date and time of the last automatic schedule extraction for this network by the Entire Operations Monitor.
Last Execution Date	Date of the last activation.
Last Run Number	Run number of the last activation.

Field/Option	Description
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 Allow setting of preferred run number during activation in the section <i>Defaults for Network Options</i> in the <i>Administration</i> documentation.</p>
Normal activation: Use Time in Schedule	<p>Select this option 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>
Normal activation: 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>
JCL Check only	<p>If this option is selected:</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 windows and lists. <hr/> <p>If this option is not selected:</p> <p>Normal activation (job submission) is initiated.</p> <p>This is the default.</p>

VII

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

18

Defining and Managing Jobs

▪ Listing Jobs	232
▪ Available Functions: Job Master	234
▪ Displaying and Modifying a Job Definition	235
▪ Adding a Job Definition	242
▪ Writing and Viewing Online Documentation for a Job	243
▪ Deleting a Job Definition	245
▪ Activating a Single Job Manually	245
▪ Displaying Job Dependencies	247
▪ Defining Job-Specific Log Information	248

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 In the object workspace, select the **Job Master** node of the required network.
- 2 From the context menu, choose **List**, or press F8.

All jobs defined for the selected network are listed in the **Job Master List** window as shown in the following example:

Job	Type	Loc	Spec.Type	Description	File or Library	Member
J2-SUB	NET			Subnetwork Job for Subnetwork SAGNETSUB2	SAGTEST/SAGNETSUB2	(D)
JOB-01	SRV			Where it all starts		
JOB-012	JOB	BS2		Depending on Job-01	\$NOP.SYSEORU	
JOB-013	JOB	MAC		Depending on JOB-012	SYSEORU	B60-M01
JOB-015	DUM			Depending on JOB-014		
JOB-019	JOB	MAC		Depending on JOB-01	SYSEORU	B60-M01
JOB-02	JOB	MAC		Dep. JOB-15, JOB-19	SYSEORU	B60-M02
JOB-03	NAT	NAT		Depending on JOB-02	SYSEORU	B60-P01
JOB-04	JOB	MAC		Depending on JOB-03	SYSEORU	B60-M01
JOB-05	DUM			Depending on JOB-04		
JOB-06	JOB	MAC		Where it all ends	SYSEORU	B60-M02
SUBNETJOB1	NET			Subnetwork job for subnetwork SAGNETSUB1	SAGTEST/SAGNETSUB1	(D)

The window columns are described in [Columns: Job Master List](#).

This section covers the following topics:

Available Functions: Job Master

➤ To list all functions available for a Job Master metanode

- In the object workspace, select a **Job Master** metanode and open the context menu.

The following functions are available:

Function	Shortcut	Description
List	F8	See <i>Listing Jobs</i> .
New	CTRL+N	See <i>Adding a Job Definition</i> .
Refresh	F5	See <i>Refreshing Object Lists</i> .
Filter	F3	See <i>Filtering Objects</i> .
Paste data	CTRL+V	See <i>Pasting Objects</i> .
Export	---	See <i>Exporting Objects</i> in the <i>Import/Export Functions</i> documentation.
Set Drag And Drop Function	---	See <i>Drag & Drop</i> .

➤ To list all functions available for a Job Master instance

- In the object workspace, select a **Job Master** instance and open the context menu.

The following functions are available:

Function	Shortcut	Description
Open	CTRL+O	See <i>Displaying and Modifying a Job Definition</i> .
Display	CTRL+D	See <i>Displaying and Modifying a Job Definition</i> .
Network Diagram	---	This function is used for navigation purposes. It opens the diagram of the network that contains the job. You can work with the diagram as described in <i>Viewing and Maintaining a Job Network Diagram</i> in the section <i>Network Maintenance</i> .
Open Subnetwork	---	Opens the Maintenance Network Master window for the network that is linked to a job of the type Subnetwork (NET).
Zoom Subnetwork	---	Opens a list of jobs associated with the network to which a job of the type Subnetwork (NET) is linked. See also <i>Listing Jobs of a Subnetwork</i> .

Function	Shortcut	Description
Subnetwork Diagram	---	This function is used for navigation purposes. It opens the network diagram for the network linked to a job of the type Subnetwork (NET). You can work with the diagram as described in Viewing and Maintaining a Job Network Diagram in the section <i>Network Maintenance</i> .
Delete	DELETE	See Deleting a Job Definition .
Copy data	CTRL+C	See Copying Objects .
Usable Symbol Tables	---	Opens the Usable Symbol Tables window to view or modify symbol table definitions available for the job. See also Listing Usable Symbol Tables in the section <i>Symbol Table and Symbol Maintenance</i> for details.
Browse Master JCL	---	See Displaying Master JCL .
Edit Master JCL	---	See Editing Master JCL and Natural Sources .
Activate Job	---	See Activating a Single Job Manually .
Pre-generate Active JCL	---	See Pregeneration of Active JCL .
Remove pre-generated Active JCL	---	See Removing Pregenerated JCL .
Edit pre-generated JCL	---	See Editing Pregenerated JCL .
List Active Jobs	---	Opens a list of active jobs for the selected job. See Listing Active Jobs in the section <i>Active Job Networks</i> .
Browse Log	---	See Displaying Logged Information - Browse Log Function .
Export	---	See Exporting Objects in the <i>Import/Export Functions</i> documentation.
Add to Workplan	---	See Add to Workplan .
Set Drag And Drop Function	---	See Drag & Drop .

Displaying and Modifying a Job Definition

» To view a job definition

- 1 In the object workspace, select a **Job Master** instance.
- 2 Open the context menu and choose **Display**, or press CTRL+D.

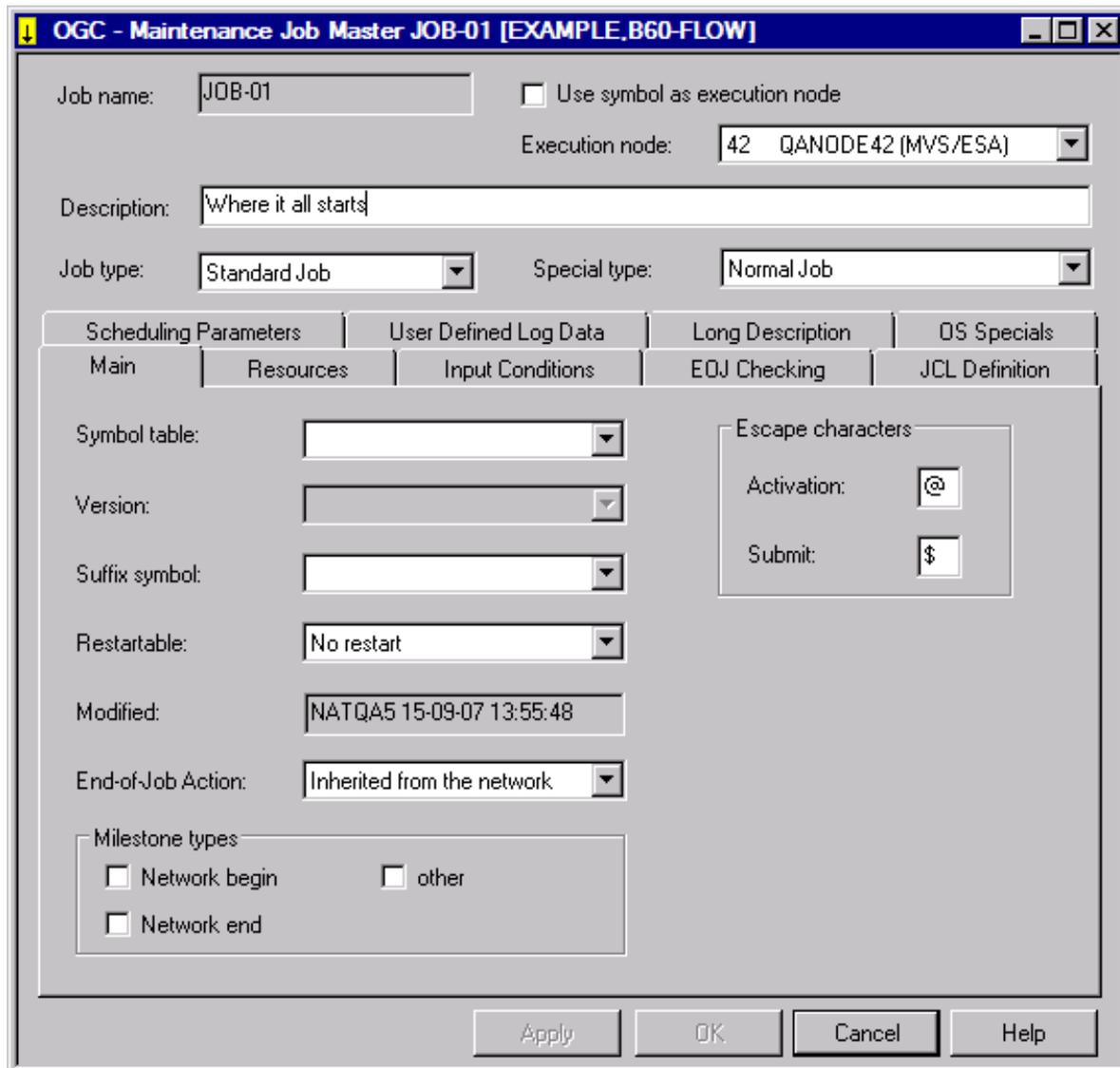
A **Display Job Master** window opens.

The protected fields and tabbed pages in the window correspond to the fields and pages of the **Maintenance Job Master window**. They are explained in *Fields: Job Definition (Master)*.

➤ **To modify a job definition**

- 1 In the object workspace, select a **Job Master** instance.
- 2 Open the context menu and choose **Open**, or press CTRL+O.

A **Maintenance Job Master** window similar to the example below opens:



- 3 Edit the fields you want to change. The fields are explained in *Fields: Job Definition (Master)*.
- 4 When you are finished, choose **OK**.

Your changes are saved.

This section covers the following topics:

- [Fields: Job Definition \(Master\)](#)

Fields: Job Definition (Master)

The fields in the **Job 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/Tabbed Page	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 in the Job Master List window.</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>You can select one of the following types from the list box:</p> <ul style="list-style-type: none"> ■ Normal Job Normal job without special condition. ■ 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/Tabbed Page	Description	
	<ul style="list-style-type: none"> ■ 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. 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. ■ Stop Job Job type Started Task: Stops a started task. Job type Windows Service: Stops a Windows service. ■ 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>). 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. 	
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.</p> <p>Note: In this field, you can also enter PMPA preceded by an escape character, for example, \$PMPA. See \$PMPA in <i>Predefined Symbols</i> in the section <i>Symbol Table and Symbol Maintenance</i>).</p>	
Use symbol as execution node	Select this check box if you want to use the suffix symbol entered in the Suffix symbol field.	
Main page:		
Symbol table	<p>Name of the symbol table to be referenced for substitution of variables in dynamic JCL. Select a name from the list. Alternatively, you can enter a new name 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>	
Version	Version of the symbol table to be used.	
(Symbol table)	Reserved names (will be replaced).	
	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)	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	

Field/Tabbed Page	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 PARA, and the symbol contains 001, 003, 012, then the active jobs PARA001, PARA003 and PARA012 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 selection options:</p> <ul style="list-style-type: none"> ■ Restart The job is to be restarted without any recovery after a system crash. ■ No restart No automatic restart after a system crash. ■ 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 /LOGON instruction:</p> <pre style="background-color: #f0f0f0; padding: 2px;">/REMARK EOR-SYSOUT-DIRECT=LATER</pre> <p>Later, the following line must appear:</p> <pre style="background-color: #f0f0f0; padding: 2px;">/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 Monitor JV must contain \$R. The system session number at the job check time must be different from the session number at submission time.</p>
Modified	User ID, date and time of the last modification of the job definition.

Field/Tabbed Page	Description
End-of-Job Action	<p>Job output conditions. Possible selection options:</p> <ul style="list-style-type: none"> ■ No impact on the job result Errors during End-of-Job action processing will have no impact on the job result. ■ 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 <code>not ok</code>. If the job was already set to <code>not ok</code>, it will remain <code>not ok</code>, 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"> ■ 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.
Milestone types	<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 selection options for user-defined milestones:</p> <ul style="list-style-type: none"> ■ Network Begin The milestone job performs as the first network job. ■ Network End The milestone job performs as the last network job. ■ 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"> ■ First Job The milestone job performs as the first network job. ■ Last Job The milestone job performs as the last network job. <p>You can use the application programming interface <code>NOPUMI1N</code> (see the section <i>API Routines</i>) to define your own milestones.</p>

Field/Tabbed Page	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 Symbol Escape Characters (<i>Symbol Maintenance</i>) and Notes about Escape Characters .
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 Symbol Escape Characters (<i>Symbol Maintenance</i>) and Notes about Escape Characters .
Other pages:	
Resources	This tabbed page is used to define a prerequisite resource for a job master. The fields and options on this page are explained in Handling Prerequisite Resources for a Job .
Input Conditions	This tabbed page is used to define input conditions for a job master. The fields and options on this page are explained in Defining and Managing Job Input Conditions .
EOJ Checking	This tabbed page is used to add, modify and delete an End-of-Job check for a job master and define a user message for the check. Furthermore, you can define an output condition for the End-of-Job check. The fields and options on this page are explained in End-of-Job Checking and Actions .
JCL Definition	This tabbed page is not available for all types of jobs. The fields and options on this page are explained in Defining JCL for a Job .
FTP Definition	This tabbed page is not available for all types of jobs. The fields and options on this page are explained in Defining Parameters for an FTP Job .
Scheduling Parameters	This tabbed page allows you to define a schedule for the job. The fields and options on this page are explained in Scheduling a Job in the section <i>Schedule Maintenance</i> .
User Defined Log Data	The fields and options on this tabbed page are explained in Defining Job-Specific Log Information .
Long Description	This tabbed page provides the option to write a long descriptive text for a job, in addition to the short job description you can enter in the Description field. See also Writing and Viewing Online Documentation for a Job .

Field/Tabbed Page	Description
Subnet	This tabbed page is only available for jobs of the type Subnetwork (NET). The fields and options on this page are described in <i>Fields: Job Type Specific Execution Features</i> in the section <i>Defining a Subnetwork</i> .
OS Specials	This tabbed page is not available for all types of jobs. The fields and options available on this page are described in <i>Fields: Operating System Specific Execution Features</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*.

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 In the object workspace, select a **Job Master** node and choose **New** from the context menu, or press CTRL+N.

Or:

Open the **Job Master List** window and choose the **New** context function from within the list table, or press CTRL+N.

A **Create new Job Master** opens. The fields contained in the window correspond to the fields in the **Maintenance Job Master window**. They are described in *Fields: Job Definition (Master)*.

- 2 Enter the required values.
- 3 Choose **OK** to save your entries.

The new job master is created and ready to use within the job network.

➤ **To copy a job definition**

- In the object workspace, select the job you want to copy from **Job Master** node and proceed as described in *Copying Objects*.

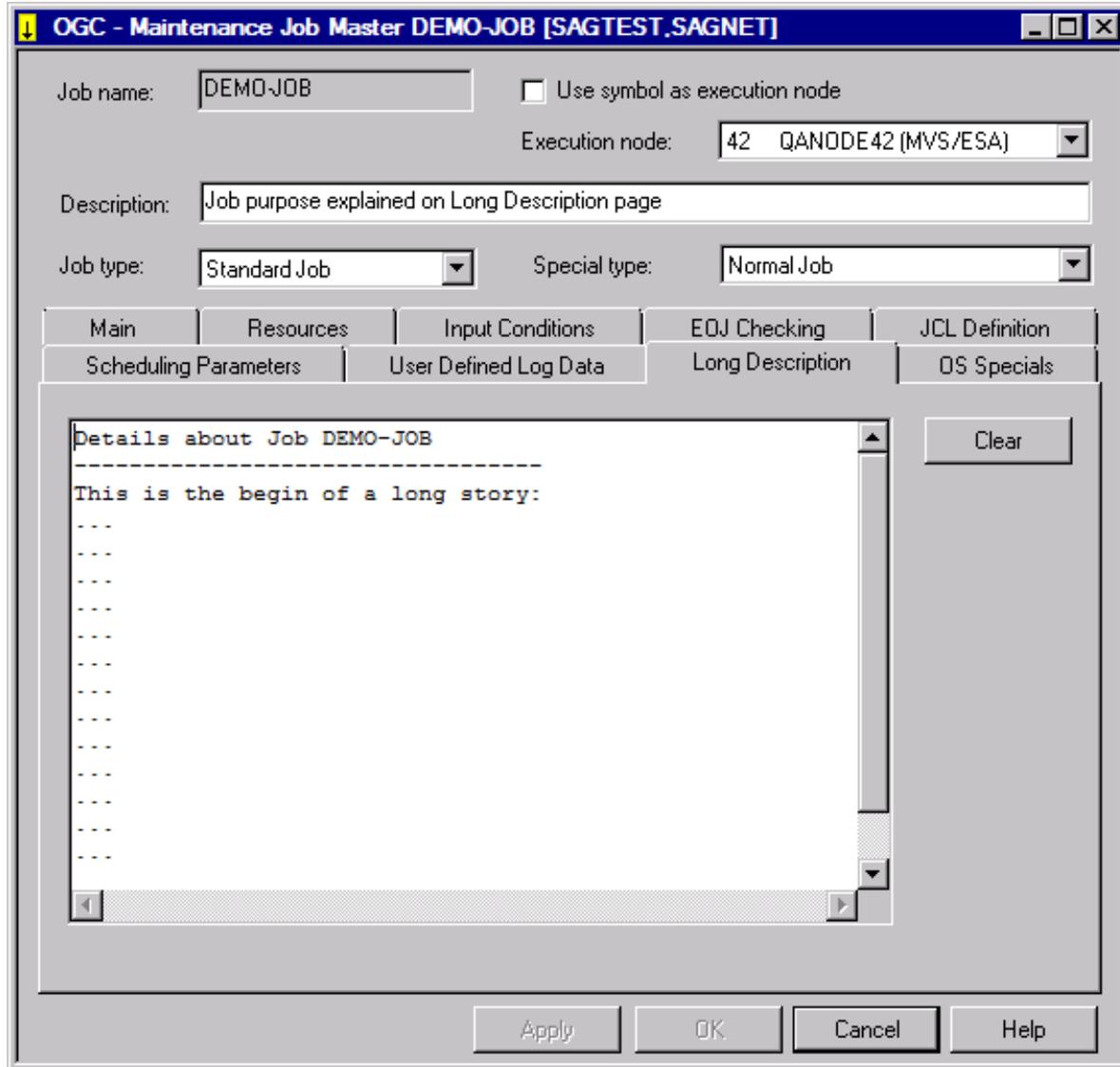
Writing and Viewing Online Documentation for a Job

You can add a short description of a job when defining a job in the **Maintenance Job Master window**. This short description appears in the list of jobs in the **Job Master List window**.

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 In the **Maintenance Job Master window**, open the tabbed page **Long Description** similar to the example below:



2 In the editor area, write new or replace existing text as required.

Choose **Clear** if you want to remove the entire text contained in the editor area.

3 Choose **OK** when you are finished.

Once online documentation is written, it can be read by any user who is authorized to access the network to which the job belongs.

You can also display or print online documentation by using the **Reporting** function described in the section [Reporting](#). If long job descriptions exist, they are included in the **Network Description (detailed)** report as indicated in 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 In the object workspace, select the job you want to delete from the **Job Master** node.
- 2 From the context menu, choose **Delete**.

Or:

Press DELETE.

A dialog opens where you have to confirm the deletion.

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:

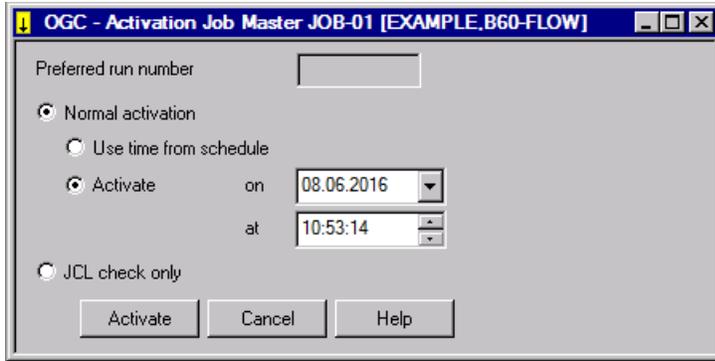
- no check for defined input conditions for the job is performed;
- no output conditions are set by the job.

All other End-of-Job actions defined for the job are executed.

➤ To activate a job manually

- 1 In the object workspace, select a **Job Master** instance and choose **Activate Job** from the context menu.

An **Activation Job Master** window similar to the example below opens:



The window title indicates the job (here: JOB-01) to be activated and the owner and network (here: EXAMPLE, B60-FLOW) of the job.

- 2 Make your definitions. The fields contained in the window are explained in [Fields: Job Activation](#).
- 3 When you are finished, choose **Activate**.

The job is activated and listed in the **Active Run** node.

This section covers the following topics:

- [Fields: Job Activation](#)

Fields: Job Activation

The fields of the [Activation Job Master window](#) are described in the following table:

Field	Description
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 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 Allow setting of preferred run number during activation in the section <i>Defaults for Network Options</i> in the <i>Administration</i> documentation.</p>
Normal activation: Use Time in Schedule	<p>Select this option 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 Activating a Job Network Manually in the section <i>Network Maintenance</i>.</p>

Field	Description
Normal activation: Activate at Date/Time	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. Note: No symbol prompting is performed for later activations.
JCL Check only	<p>If this option is selected:</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 windows and lists. <p>If this option is not selected:</p> <p>Normal activation is used.</p> <p>An actual JCL check is not performed. This is the default.</p>

Displaying Job Dependencies

The **Network Diagram** function 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.

Displaying Job Dependencies for a Specific Job

➤ To view job dependencies in a network

- 1 In the object workspace, select a job from the **Job Master** or **Job Active** node.
- 2 Open the context menu and choose the **Network Diagram** function.

Depending on the node selected, a **Diagram Network Master** or **Diagram Active Run** window (see the examples in the section *Network Maintenance*) opens with a graphical view of the selected network.

The symbols used to represent the network components and the functions available from the context menu in the diagram, are described in *Explanations of Diagram Symbols* and *Maintenance Functions for Diagrams* in the section *Network Maintenance*.

In a complex network, you can use the **Find** function to locate the required job.

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 In the [Maintenance Job Master window](#), open the tabbed page **User Defined Log Data**:

Fields: User-Defined/Extended Log

The input fields of the **User Defined Log Data** page are described in the following table:

Field/Column	Description	
Log SYSOUT	Specifies logging of job SYSOUT files after job termination. Possible check box settings:	
	<i>checked</i>	Logs SYSOUT.
	<i>unchecked</i>	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>Defaults for Network Options</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.	
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 check box settings:	
	<i>checked</i>	Logs JCL.
	<i>unchecked</i>	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.	
Log the selected message if it contains one of the following strings	Specifies additional selection strings for the message entered and selected in the Code column. The message will be logged if no additional string is specified, or if at least one specified string will be found in the 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.

19

Defining Job Types and Job Execution Features

- Available Job Types 252
- Defining Job Type Specific Execution Features 254
- Defining Operating System Specific Execution Features 257

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 **Maintenance Job 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 selected in the **Maintenance Job Master window**, additional pages are provided to define individual execution features 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
<p>DAT</p> <p>Text File</p>	<p>A UNIX, Windows or DAT text file.</p> <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 Windows Service job, and define the special type Stop Job described in <i>Fields: Job Master 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 Started Task job, and define the special type Stop Job described in <i>Fields: Job Master 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

- In the **Maintenance Job Master window**, open the extra page that appears when selecting one of the following job types in the **Job type** field:

For the job type **File Transfer Job** (FTP), the tabbed page **FTP Definition** is provided.

For the job type **Subnetwork** (NET), the tabbed page **Subnet** is provided.

For the job type **Text File** (DAT), the job type dependent fields are contained on the tabbed page **OS Specials**.

All type specific fields are described in [Fields: Job Type Specific Execution Features](#).

This section covers the following topics:

- [Fields: Job Type Specific Execution Features](#)

Fields: Job Type Specific Execution Features

The job types for which type specific fields are provided on additional pages of the **Job Master** window including the **OS Specials page** are described in the following table.

Job Type	Field	Description						
NET Subnetwork (tabbed page Subnet)	JCL Node	Node where the JCL of the job (if defined) is located.						
	Subnetwork owner	Owner of the subnetwork. Enter a name or select a name from the drop-down list box.						
	Subnetwork name	Name of the subnetwork. Enter a name or select a name from the drop-down list box.						
	Subnetwork version	Version of the subnetwork if defined.						
	Activation mode	<p>Activation time: see <i>Time of Activation of a Subnetwork</i> in the <i>System Overview</i>.</p> <p>Possible selection options:</p> <table border="1"> <tr> <td>At activation time</td> <td>At the moment of the activation of the calling job.</td> </tr> <tr> <td>At submission time</td> <td>At the moment of the start of the calling job.</td> </tr> <tr> <td>Use global defaults</td> <td>Use global default values as set in the Entire Operations defaults for Network Options (see the <i>Administration</i> documentation).</td> </tr> </table>	At activation time	At the moment of the activation of the calling job.	At submission time	At the moment of the start of the calling job.	Use global defaults	Use global default values as set in the Entire Operations defaults for Network Options (see the <i>Administration</i> documentation).
	At activation time	At the moment of the activation of the calling job.						
	At submission time	At the moment of the start of the calling job.						
	Use global defaults	Use global default values as set in the Entire Operations defaults for Network Options (see the <i>Administration</i> documentation).						
	Append multiple suffixes	<p>Select this check box to append the suffix of the calling multiple job to the name of the subnetwork.</p> <p>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.</p>						
	Use subnetwork time frames	<p>Possible check box settings:</p> <table border="1"> <tr> <td><i>unchecked</i></td> <td>Use the time frame from the calling job of the type NET (default).</td> </tr> <tr> <td><i>checked</i></td> <td>Use the time frame defined for the subnetwork itself.</td> </tr> </table>	<i>unchecked</i>	Use the time frame from the calling job of the type NET (default).	<i>checked</i>	Use the time frame defined for the subnetwork itself.		
	<i>unchecked</i>	Use the time frame from the calling job of the type NET (default).						
	<i>checked</i>	Use the time frame defined for the subnetwork itself.						

Job Type	Field	Description	
DAT Text File (tabbed page OS Specials)	Submit User ID	The user ID which will be used to write a DAT text output file.	
	Target Location	(No JCL required)	Generate active JCL only.
		SAM or ISAM file	BS2000 text file.
		LMS library	BS2000 LMS file.
		Natural source	Natural source object.
		Macro Natural source	Natural source object for macro JCL.
		Partitioned data set	z/OS partitioned data set.
		Text file	UNIX or Windows text file.
		z/VSE sublibrary	z/VSE sublibrary member with type.
	Target overwrite	This option is available only for some target locations.	
Possible check box settings:			
<i>unchecked</i>		Append the output to target file.	
	<i>checked</i>	Overwrite target file with the output.	
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.		
input field/selection box for member type	Additional selection box or input field for z/VSE or BS2000, respectively. BS2000, LMS: The LMS member type. For possible selection options, 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 (tabbed page SAP Definitions)	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.		

Job Type	Field	Description
	SAP User ID	User ID (for the Job) in the SAP system.
	Password	Password in the SAP system. The ciphering is to be performed with the jexa4S utility encryptpw.
	External User ID	External user for SAP jobs. Default: Submit User ID of the Entire Operations job.
SRV Windows Service (tabbed page OS Specials)	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

- In the **Job Master** window, open the tabbed page **OS Specials** shown in the following example of a BS2000 job:

The input fields provided depend on the operating system (here: BS2000) of the execution node.

(The **OS Specials** page is only available if execution features can be specified for the operating system of the selected execution node.)

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.

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	<p>Valid logon user ID defined for BS2000.</p> <p>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>
	Submit User ID	<p>Jobs in BS2000 are submitted under this user ID by the Entire Operations Monitor.</p> <p>This ID has no meaning for other operating systems.</p> <p>Note: The user ID TSOS can be defined only if the user defining the ID is working under TSOS.</p> <p>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.</p> <p>If this field is left blank, then the default user ID is inserted at job activation.</p> <p>See also the default setting User ID Definition (<i>Defaults for Other Settings, Administration</i> documentation), and the sections Operating System User IDs and Default User ID Determination.</p>
	Account Number	<p>Account number to be used for the BS2000 Submit User ID.</p> <p>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).</p> <p>Symbol replacement is performed in this field if the activation escape character is used.</p>
	BS2000 Job Class	<p>Valid job class for job submission defined for BS2000.</p> <p>Symbol replacement is performed in this field if the activation escape character is used.</p>
	Job Priority	<p>If not empty, this job priority will be used during submission and will override a possible setting in the LOGON statement.</p> <p>The default value on the network level will be used for new job definitions.</p>

Operating System	Field	Description
	Run Priority	<p>If not empty, this run priority will be used during submission and will override a possible setting in the LOGON statement.</p> <p>The default value on the network level will be used for new job definitions.</p>
	Share SYSOUT	<p>Make BS2000 SYSOUT file shareable.</p> <p>If you select this check box, the internal temporary SYSOUT file can be accessed from other BS2000 user IDs.</p>
	Submit Password	<p>Password for job submission. (Required only for BS2000 password encryption where submit node Entire System Server version is less than Version 3.4.1).</p>
	SYSOUT User ID	<p>User ID under which internal SYSOUT files are created by Entire Operations. If you leave this field blank, the Submit User ID is used.</p> <p>See also the default setting User ID Definition described in <i>Defaults for Other Settings</i> in the <i>Administration</i> documentation.</p>
	SYSOUT Cat ID	<p>Enter the SYSOUT catalog ID.</p> <p>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.</p>
	Collect SYSLST	<p>If you select this check box, 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.</p>
	Monitor JV	<p>Name of the BS2000 Monitor Job Variable to be used at job submission.</p> <p>If you leave this field blank, Entire Operations generates an internal unique name if the product JOB VARIABLES is available.</p>
	Password	<p>Password for the defined BS2000 Monitor Job Variable.</p> <p>Only alphanumeric passwords are supported.</p>
z/OS	Submit User ID	<p>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.</p> <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>Defaults for Other Settings, Administration</i> documentation), and the sections <i>Operating System User IDs</i> and <i>Default User ID Determination</i>.</p>

Operating System	Field	Description	
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>Defaults for Other Settings, Administration</i> documentation), and the sections <i>Operating System User IDs</i> and <i>Default User ID Determination</i>.</p>	
	Use Symbol as SYSOUT Node	Select this check box if you want to use the suffix symbol for the SYSOUT node.	
	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>	
	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>	
	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>Defaults for Other Settings, Administration</i> documentation), and the sections <i>Operating System User IDs</i> and <i>Default User ID Determination</i>.</p>	
	Command Line obligatory	<i>checked</i>	The existence of the symbol <code>CMDLINE - job</code> 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> .
		<i>unchecked</i>	A command line is not obligatory for the job (default).

20

Using a Dummy Job

- Dummy Job Execution by the Monitor 264
- Permanent Dummy Jobs 264
- Temporary Dummy Jobs 264
- Excluding a Job from Actual Execution 265
- End-of-Job Actions after Execution as a Temporary Dummy Job 266
- Supervision of Running Jobs 267

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 **Dummy Job** (`DUM`) 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 **Standard Job** (`JOB`) have a master JCL definition (see *Defining and Managing JCL for a Job*).

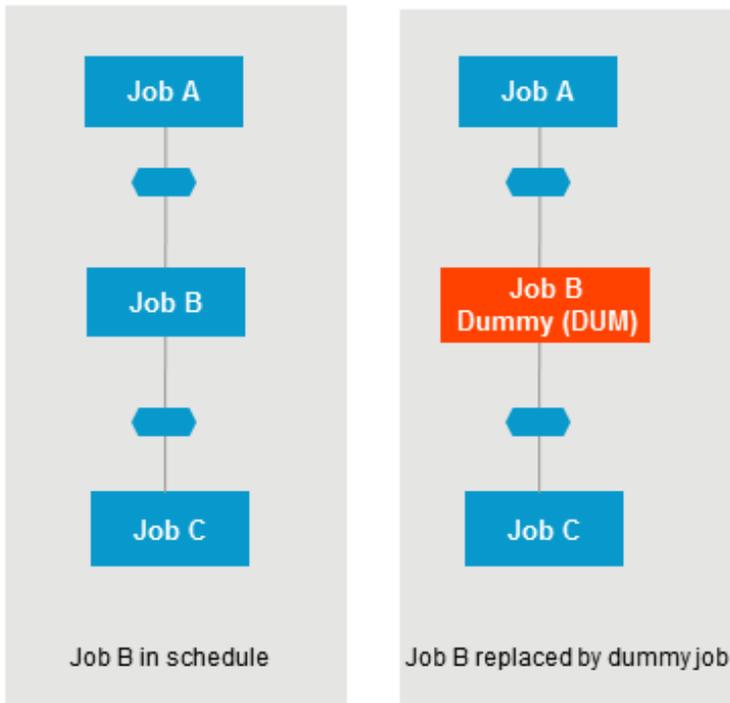
Temporary dummy jobs always have the execution time zero, regardless of which estimated elapsed time is defined for the job.

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 Execute as dummy (D) 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.
JCL check	Execution of the job control in the mode <i>JCL check</i> . Thus, jobs (see Fields: Job Activation) or Job Networks (see Fields: Network Activation) 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:



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 temporary 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:

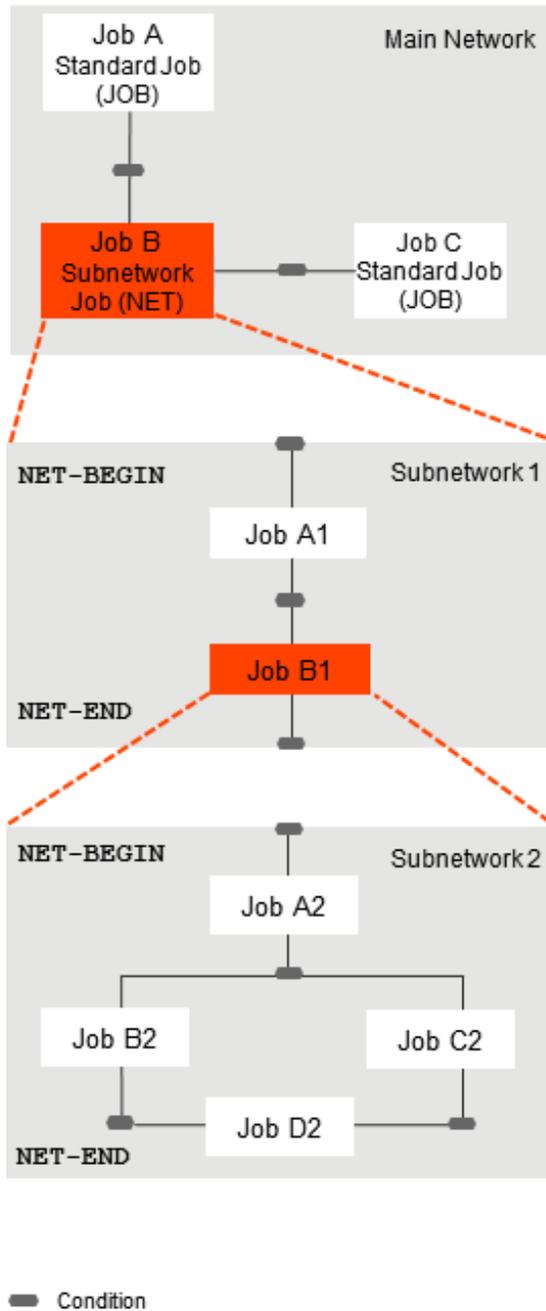
- is sent to all mailboxes linked to the network;
- is sent to Con-nect (if linked);
- is written to the log;
- appears in the **List Active Jobs**, **Active Job List** or **List Active Jobs Job Active** window.

21

Defining a Subnetwork

▪ Link to the Main Network	271
▪ Subnetwork Activation and Execution	272
▪ Restrictions for Subnetworks	273
▪ Adding or Modifying a Subnetwork Definition	274
▪ Listing Jobs of a Subnetwork	276

The job type **Subnetwork** (NET) 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.

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 **Subnetwork** (NET) 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 *Defaults for Network Options* in the *Administration* documentation).
- In the definition of the calling job of the type NET (see **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 display or modify a subnetwork definition

- 1 From a **Job Master** node in the object workspace, select a job of the type **Subnetwork** (NET), and choose **Display** or **Open** from the context menu.

Or:

From the table in a **Job Master List** window, select a job of the **Type** NET, and choose **Display** or **Open** from the context menu.

A **Maintenance Job Master window** opens.

- 2 Open the **tabbed page Subnet**.

The fields on the page are described in *Fields: Job Type Specific Execution Features*.

➤ To add a subnetwork definition

- 1 From the **Job Master window**, select **Subnetwork** from the **Job type** drop-down list box.

The tab **Subnet** appears.

- 2 Open the tabbed page **Subnet** similar to the example below:

OGC - Maintenance Job Master SUBNETJOB1 [SAGTEST.B60-FLOW]

Job name: SUBNETJOB1 Use symbol as execution node

Execution node: 42 QANODE42 (MVS/ESA)

Description: Subnetwork job for subnetwork SAGNETSUB1

Job type: Subnetwork Special type: Normal Job

Main Resources Input Conditions EOJ Checking Scheduling Parameters

User Defined Log Data Long Description Subnet OS Specials

Subnetwork owner: SAGTEST

Subnetwork name: SAGNETSUB1

Subnetwork version:

Activation mode: Use global defaults

Append multiple suffixes

Use subnetwork time frames

Apply OK Cancel Help

- 3 Make your definitions. The input fields available on the page are described in *Fields: Job Type Specific Execution Features*.
- 4 Choose **OK** when you are finished.

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 the jobs of a subnetwork**

- From a **Job Master** or **Job Active** node, select a job of the type **Subnetwork (NET)**, and choose **Zoom Subnetwork** from the context menu.

Or:

From the table in a **Job Master List** or an **Active Job List** window, select a job of the **Type NET**, and choose **Zoom Subnetwork** from the context menu.

The jobs of the subnetwork defined for the selected **NET** job are listed in a (separate) **Job Master List** or **Active Job List** window, as shown in the following example:

Job	Type	Loc	Spec.Type	Description	File or Library	Member
JOB-1	JOB	NAT		Where it all starts	SYSEXP	JCLJOB
SUBNETJOB2	NET			Subnetwork Job for Subnetwork SAGNETSUB2	SAGTEST/SAGNETSUB2	(D)

22

Defining Parameters for an FTP Job

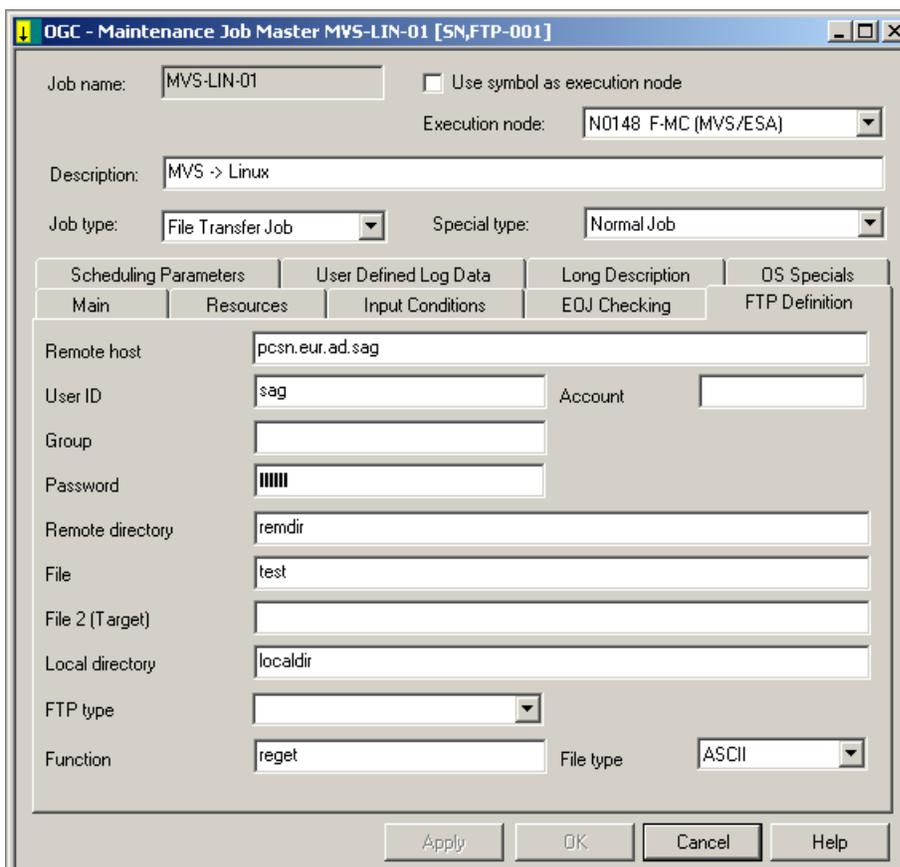
- Adding an FTP Parameter Definition 278
- JCL Generation for FTP Jobs 280

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 the FTP definition for a job

- 1 In the [Maintenance Job Master window](#), select [File Transfer Job](#) from the [Job type drop-down list box](#) and open the tabbed page [FTP Definition](#) similar to the example below:



The screenshot shows a window titled "OGC - Maintenance Job Master MVS-LIN-01 [SN,FTP-001]". The "Job type" is set to "File Transfer Job" and the "Special type" is "Normal Job". The "FTP Definition" tab is active, showing fields for Remote host (pcsn.eur.ad.sag), User ID (sag), Group, Password (masked), Remote directory (remdir), File (test), File 2 (Target), Local directory (localdir), FTP type, Function (reget), and File type (ASCII). Buttons for Apply, OK, Cancel, and Help are at the bottom.

- 2 Make your definitions.

The input fields are explained in [Fields: FTP Parameter Definition](#).

- 3 When you are finished, choose **OK** to save your entries.

This section covers the following topics:

- Fields: FTP Parameter Definition

Fields: FTP Parameter Definition

The fields on the [FTP Definition page](#) 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.
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.
Account	Account for the FTP login on the remote host. Required for some platforms only. Symbols can be used.
Group	Group or domain, respectively, for the FTP login on the remote host. Required for some platforms only. Symbols can be used.
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.
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 set to Standard FTP .	
Function	File transfer function to be used. Examples: get, put, mget, mput	
File Type	File type. Possible selection options:	
	ASCII	ASCII for text files.
	binary	Binary for other files.

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*.

23

Defining and Managing JCL for a Job

▪ Job Control for Jobs under BS2000	284
▪ Job Control for Jobs under UNIX	284
▪ Job Control for Jobs under Windows	285
▪ Defining Master JCL for a Job	288
▪ Displaying Master JCL	292
▪ JCL Locations	293
▪ Defining Operating System Dependent JCL Specifications	297
▪ Handling JCL during Job Submission	298
▪ Symbol Replacement in JCL	301
▪ Pregenerating Active JCL	301

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 In the **Maintenance Job Master window**, open the tabbed page **JCL Definition**:

OGC - Maintenance Job Master TEST-JOB [SAGTEST.SAGNET]

Job name: TEST-JOB Use symbol as execution node

Execution node: 42 QANODE42 (MVS/ESA)

Description: Job for JCL Test

Job type: Standard Job Special type: Normal Job

Scheduling Parameters User Defined Log Data Long Description OS Specials

Main Resources Input Conditions EOJ Checking JCL Definition

Use symbol as JCL node

JCL node: 42 QANODE42 (MVS/ESA) JCL load mode: At activation

JCL location: Macro Natural source

File/Natlib: SYSEORU

Member: E51-M01 Password: Edit

OS Specials

JCL user ID:

Apply OK Cancel Help

- 2 Make your definitions. The input fields and special functions available are explained in [Fields: Master JCL Definition](#) and [Functions: Master JCL Definition](#).
- 3 Choose **OK** to save your definitions.

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](#)

▪ [Functions: Master JCL Definition](#)

Fields: Master JCL Definition

The fields on the [JCL Definition page](#) are described in the following table:

Field	Description			
Use symbol as JCL node	Select this check box if you want to use a symbol as a JCL node.			
JCL Location	Type of JCL storage location. For selection options, see JCL Locations . In a network definition, this field serves as a default for the entire network. See also the section . 			
JCL Node	JCL for the job is read from this node. The job is started on the execution node . The default is the value specified in the job network definition (see Fields: Network Definition), which can be overwritten here. 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> .			
JCL Load Mode	Possible selection options:			
	<table border="1"> <tbody> <tr> <td>At activation</td> <td>Load JCL at activation time (default).</td> </tr> <tr> <td>At submission</td> <td>Load JCL at submission time. Use this only if prerequisites for JCL are not available at activation time. Note: 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.</td> </tr> </tbody> </table>	At activation	Load JCL at activation time (default).	At submission
At activation	Load JCL at activation time (default).			
At submission	Load JCL at submission time. Use this only if prerequisites for JCL are not available at activation time. Note: 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.			

Field	Description
	<p>6. For other types of JCL, we still recommend submission time symbol replacement, which causes less system overhead.</p>
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 select the required library or file from the drop-down list box.</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 select the required member from the drop-down list box.</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. 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). ■ at JCL loading (permanently, using the active symbol table). <p>This applies to all JCL locations which support members: see <i>List of JCL Locations</i>.</p>

Field	Description
	<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	<p>Volume serial of the file.</p> <p>Only required if the file has not been cataloged (z/OS only).</p>
Password	The file password if the file has been protected by a password.
OS Specials	Operating system-dependent input fields: see Fields: Operating System Specials for JCL .

Functions: Master JCL Definition

The following special function is available on the [JCL Definition page](#):

Function	Description
Edit	<p>Opens the editor for the source entered in the Member field or the file entered in the File/Natlib field depending on the specified JCL location.</p> <p>This command is only available if you are authorized to edit the specified source and if the Apply and OK buttons are disabled in the window.</p> <p>This command corresponds to the Edit Master JCL or Edit Active JCL function available from the context menu of a selected job instance.</p> <p>See also Editing Master JCL and Natural Sources.</p>

Displaying Master JCL

➤ To display the master JCL source defined for a job

- In the object workspace, select a **Job Master** instance and choose **Browse Master JCL** from the context menu.

A **Browse** window similar to the [Edit window](#) opens displaying the source of the JCL.

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 select for JCL from the **JCL location** drop-down list box on the [JCL Definition page](#).

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/Selection 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/Selection 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 <i>Restriction for Job Types</i> .
PDS Partitioned data set	z/OS	Member of a partitioned data set. See also <i>Restriction for Job Types</i> .
PRC Callable procedure	BS2000	BS2000 procedure. See <i>JCL Location PRC: BS2000 Procedure</i> 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 **FTP Definition page**.

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

- **On the JCL Definition page**, enter valid values in the input fields provided for the relevant operating system in the **OS Specials** section.

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 **OS Specials** section of the **JCL Definition page** 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>Defaults for Other Settings</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>Defaults for Other Settings, Administration</i> documentation), and the sections <i>Operating System User IDs</i> and <i>Default User ID Determination</i>.</p>

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 SNNOPLEX (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 SNNPEX (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](#)

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.

➤ To pregenerate active JCL

- In the object workspace, select a **Job Master** instance and choose **Pre-generate Active JCL** from the context menu.

If no generation errors occur, a message indicates that active JCL is pregenerated for the job.

- [Editing Pregenerated JCL](#)
- [Removing Pregenerated JCL](#)

Related Topic:

- [Regenerating Active JCL](#) in the section *Active Job Networks*

Editing Pregenerated JCL

This function allows editing the pregenerated JCL.

➤ **To edit pregenerated JCL**

- 1 In the object workspace, select a **Job Master** instance for which pregenerated active JCL exists.
- 2 Choose **Edit pre-generated JCL** from the context menu.

An **Edit window** opens where you can change the pregenerated JCL as required.

See also [Editing Master JCL and Natural Sources](#).

Removing Pregenerated JCL

➤ **To remove pregenerated active JCL**

- 1 In the object workspace, select a **Job Master** instance for which pregenerated active JCL exists.
- 2 Invoke the context menu and choose the **Remove pre-generated Active JCL** function.

The message `Pregenerated JCL deleted` indicates that the pregenerated JCL has been removed.

24

Editing Master JCL and Natural Sources

- General Editing Considerations 304
- UNIX and Windows: Handling of Tab Characters (H'09') within JCL Lines 305
- Usage of Text Objects in JCL 305
- Using the Editor 305
- Editing Macro Sources for Dynamic JCL Generation 310
- Locking of Natural Sources 319

JCL and Natural source objects can be edited with the internal 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 GUI Client*.

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. See [File Names in Job Control for Jobs under Windows in the section System Overview](#).

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 In the object workspace, select a **Job Master** instance and choose **Edit Master JCL** or **Edit Active JCL** from the context menu.

Or:

On the **JCL Definition page** in the **Maintenance Job Master** or **Maintenance Job Active** window, choose **Edit** (see also *Functions: Master JCL Definition*).

An **Edit** window similar to the following examples opens.

(If the source of the specified JCL member does not yet exist, the **Edit** window is empty.)

- Example of JCL for the **location Partitioned data set** (PDS):

```

OGC - Edit JOB-PDS [SAGTEST.SAGNET]
//NOPE02J1 JOB @UID,CLASS=@CLASS,MSGCLASS=@MSGCLASS,MSGLEVEL=(1,1) 00000010
//* ***** 00000050
//* IF STEP02 IS NOT EXECUTED 00000090
//* THEN CONDITION 'E02-J01-OK' ==> JOB 'E02-J02' WILL BE EXECUTED 00000090
//* ELSE CONDITION 'E02-J01-NOK' ==> JOB 'E02-J03' WILL BE EXECUTED 00000090
//* END-IF 00000090
//* ***** 00000050
//JOBLIB DD DSN=@JOBLIB,DISP=SHR 00000090
//STEP01 EXEC PGM=NOPCONTI 00000090
//*STEP02 EXEC PGM=NOPCONTI,PARM='U202' 00000090
//
    
```

The example above shows the JCL member NOPE02J3 of the job JOB-PDS.

- Example of JCL for the **location Macro Natural source** (MAC):

```

OGC - Edit E51-M01 [SYSEORU]
@ DEFINE DATA PARAMETER USING NOPXPL-A
@ LOCAL /* LOCAL VARIABLES START HERE
@ 1 #JOBNAME (A10)
@ 1 #DATE (D)
@ 1 #WEEKDAY (A10)
@ END-DEFINE
@ * -----
@ MOVE P-JOB TO #JOBNAME
@ EXAMINE #JOBNAME FOR '-' DELETE
@ MOVE P-ACTIVATION-TIME TO #DATE
@ MOVE EDITED #DATE (EM=N(10)) TO #WEEKDAY
//EORE51J1 JOB ,@P-OWNER,MSGCLASS=@MSGCLASS,CLASS=@CLASS
//*
//STEP01 EXEC PGM=NOPCONTI,PARM='C=0000,T=00:00:10'
//STEPLIB DD DISP=SHR,DSN=@STEPLIB
@ IF #WEEKDAY = 'Wednesday'
//* -- THIS STEP IS EXECUTED ON WEDNESDAYS ONLY --
//WEDNESDY EXEC PGM=NOPCONTI,PARM='C=0004,T=00:10:00'
//STEPLIB DD DISP=SHR,DSN=@STEPLIB
@ END-IF
//*
    
```

The example above shows the macro Natural source E51-M01 contained in the Natural library SYSEORU.

- 2 Edit the source as required.

You can use the editor functions provided in the context menu of the **Edit** window (or the toolbar of the **main application window**) to find and change text, navigate in the source or perform special functions for macro sources. See *Available Functions: Edit Window* and *Editing Macro Sources for Dynamic JCL Generation*.

➤ **To edit the source of a user exit**

- 1 In the object workspace, select a **Job Master** instance and open the **Input Conditions** page.

Select an input condition of the type **Exit**.

Choose **Edit**.

Or:

In the object workspace, select a **Job Master** instance and open the **EOJ Checking** page.

Select an action user exit (indicated by an **X** in the **Action** column) or a checking user exit and choose **Edit Action Exit** or **Edit User Exit**, respectively.

An **Edit** window similar to the following opens:

```

OGC - Edit BS2EJA01 [SYSEORU]
DEFINE DATA PARAMETER USING NOPXPL-A
LOCAL      /* ... ALL LOCALS SHOULD START WITH 'L-'
1 L-DUMMY          (A1)
1 #USERID          (A8)
1 #DBENV           (A10) CONST <' (THIS) '>
1 #RUN-I4          (I4)
1 #USE-NAT-MSG-CACHE (L)  CONST <FALSE>
1 #LOG-STATUS      (N4)
1 #LOG-MESSAGE     (A70)
END-DEFINE
* -----
WRITE *PROGRAM *TIME
 / '=' P-JOB-ID-10
#USERID := *USER
#RUN-I4 := P-RUN
MOVE 8001 TO #LOG-STATUS
COMPRESS P-CALL-PLACE 'Job Id' P-JOB-ID-10
 INTO #LOG-MESSAGE
CALLNAT 'NOPULW8N'
 #USERID (AD=O) #LOG-STATUS (AD=O) #LOG-MESSAGE (AD=O)
 #DBENV (AD=O)
 P-OWNER (AD=O) P-NETWORK (AD=O) #RUN-I4 (AD=O) P-JOB (AD=O)
 P-JOB-ID-10 (AD=O)
 'NOP' #USE-NAT-MSG-CACHE (AD=O)|
RESET P-RC
MOVE #LOG-MESSAGE TO P-RT
END

```

(If the source of the selected user exit has not yet been created and cataloged, the **Edit** window only contains an incomplete `DEFINE` sample statement.)

- 2 Edit the source as required.

In order to use a user exit, it must be saved and cataloged as a Natural object in a Natural library.

For information on coding action user exits, see also [User Exits for End-of-Job Checking and Actions](#) in the section *User Exits*.

You can use the editor functions provided in the context menu of the **Edit** window (or the toolbar of the **main application window**) to find and change text, navigate in the source or perform special functions for macro sources. See [Available Functions: Edit Window](#).

This section covers the following topics:

- Available Functions: Edit Window

Available Functions: Edit Window

The following functions are provided from the context menu in an **Edit window** or the toolbar in the **main application window**:

Function	Shortcut	Description
Undo	CTRL+Z	Undo the last action.
Redo	CTRL+Y	Redo the last action.
Cut	CTRL+X	Cut the selected text string to the clipboard.
Copy	CTRL+C	Copy the selected text string to the clipboard.
Paste	CTRL+V	Paste the text string previously cut or copied to the clipboard.
Delete	DELETE	Delete the selected text string.
Select All	CTRL+A	Select the whole text contained in the Edit window.
Upper	CTRL+SHIFT+U	Convert the selected text string to uppercase letters.
Lower	CTRL+SHIFT+L	Convert the selected text string to lowercase letters.
Find	CTRL+F	Open the Find dialog to enter a search string and specify search options (case sensitive and/or whole word only) and the search direction backward (scrolls up in the source) or forward (scrolls down in the source). By default, the currently selected text string is entered in the Find dialog.
Find Next	F3	Find the next occurrence of the search string specified in the Find dialog.
Replace	CTRL+H	Open the Replace dialog to enter a Find search string and a Replace text string and specify search options and the search directory as described for the Find dialog. By default, the currently selected text string is entered in the Find dialog.
Replace Next	CTRL+F3	Replace the next occurrence of the search string specified in the Replace dialog.
Go To	CTRL+G	Open the Go To dialog to place the cursor on the source line entered in the dialog.
Print	CTRL+P	Output the text on a printer.
Check	---	Applies to Natural sources only. Check the syntax of the source.
Catalog	---	Applies to Natural sources only. Catalog the source.
Stow	---	Applies to Natural sources only. Check the syntax and catalog the source.

Function	Shortcut	Description
Macro	---	Applies to macro sources only. See Special Editor Functions for Macro Sources .
Test	---	Applies to macro sources only. See Special Editor Functions for Macro Sources .

Editing Macro Sources for Dynamic JCL Generation

Jobs with **JCL location Macro Natural source** (MAC) 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 function must be used to generate the final JCL. The **Test** editor function can be used to test the generated JCL.



Note: The **Test** function 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 Functions 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 Functions for Macro Sources

The following special editor functions are available for a macro source:

Function	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 function 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 function. 2. The generated object is not compiled with the Natural STOW command during execution of the Test function.

Run Number used for the Test Function

The reserved run number used during the **Test** function 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 **Macro Natural source** (MAC) 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 **Macro Natural source (MAC)**:

```

$ 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`.

25

Defining and Managing Job Input Conditions

▪ Listing Input Conditions	322
▪ Adding and Modifying a Master Input Condition	325
▪ Accessing another Network Defined for an Input Condition	329
▪ Input Condition References	330
▪ Global Conditions	333
▪ Input Condition: User Exit	333
▪ Input Condition: Multiple Suffixes	336
▪ Input Condition: File Existence	337
▪ Input Condition: Mailbox	339
▪ Input Condition: Symbol Value	340
▪ Input Condition: User Switch - BS2000	343
▪ Input Condition: Job Variable - BS2000	344
▪ Listing Jobs Linked to an Input Condition	346
▪ Deleting an Input Condition Definition	347

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*.

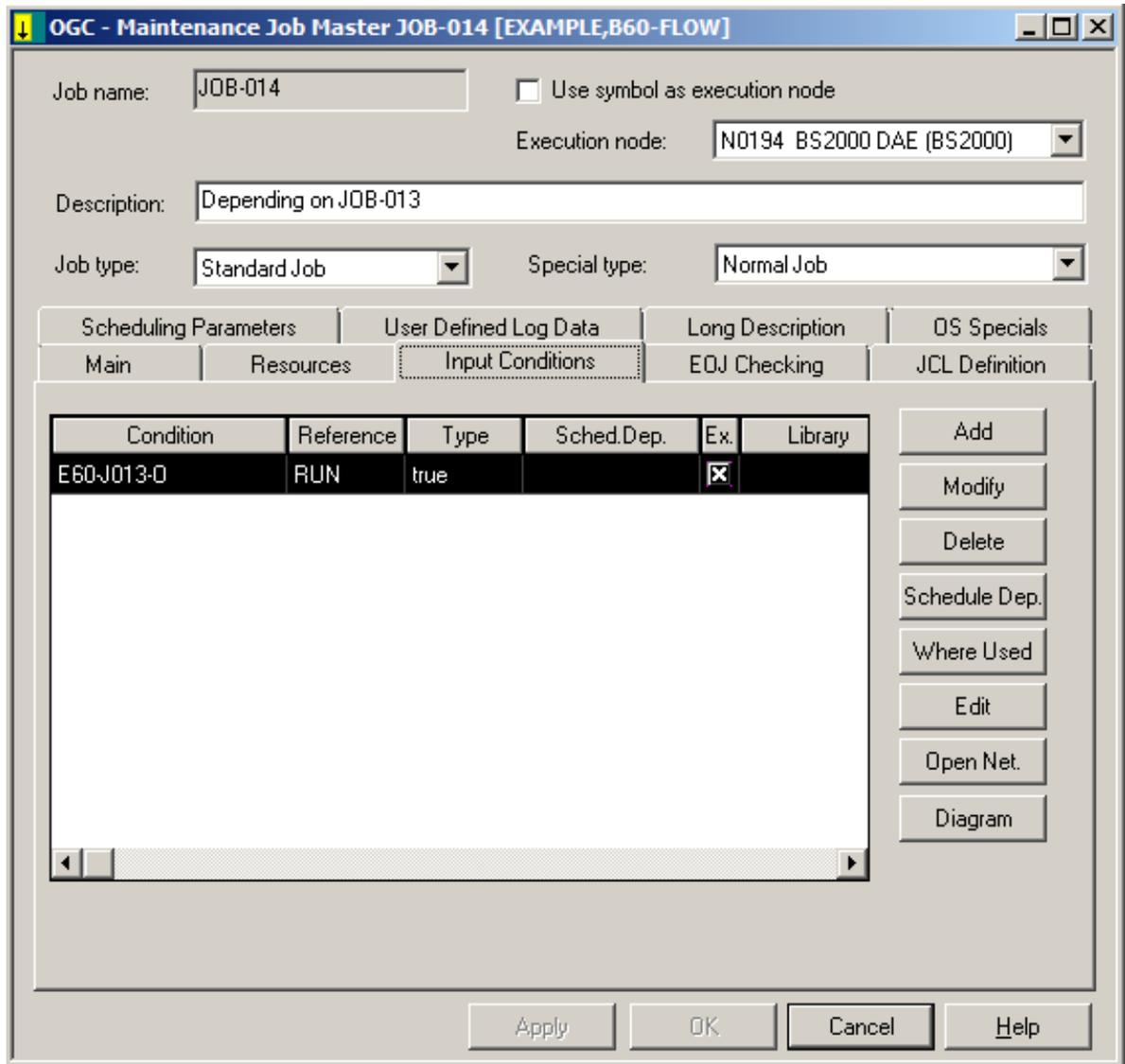
Related Topics:

- *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

- In the **Maintenance Job Master window**, open the tabbed page **Input Conditions** similar to the example below:



All input conditions defined for the job are listed on the page.

The columns and functions available on the page are explained in [Columns: Input Conditions Maintenance](#) and [Functions: Input Conditions](#).

This section covers the following topics:

- [Columns: Input Conditions](#)

▪ [Functions: Input Conditions](#)

Columns: Input Conditions

The following table explains the column headings for the data listed on the [Input Conditions page](#) of the **Maintenance Job Master** window:

Column	Description																																				
Condition	User-defined condition name. See also Restrictions for Condition Names and Global Conditions .																																				
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>mult.Sfx.</td> <td>Condition depends on multiple suffixes.</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> </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.	mult.Sfx.	Condition depends on multiple suffixes.	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).
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.																																				
mult.Sfx.	Condition depends on multiple suffixes.																																				
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 .																																				

Column	Description	
Ex.	x mark in the table	Condition must exist.
	no mark in the table	Condition must not exist.
Library	Natural library where a defined user exit resides.	

Functions: Input Conditions

The following functions are available on the [Input Conditions page](#):

Function	Description
Add	Add an input condition definition.
Modify	Modify the input condition definition.
Delete	Delete the input condition definition.
Schedule Dep.	Add or modify schedule dependency. See Defining Schedule Dependencies for an Input Condition in the section <i>Schedule Maintenance</i> .
Where Used	Display jobs for which the condition is used as an input or output condition. See Viewing the Usage of a Prerequisite Resource .
Edit	Edit user exit to set input condition.
Open Net.	Open the network definition for an external input condition. See Accessing another Network Defined for an Input Condition .
Diagram	View the network diagram for the input condition. For further information, see Viewing and Maintaining a Job Network Diagram .

Adding and Modifying a Master Input Condition

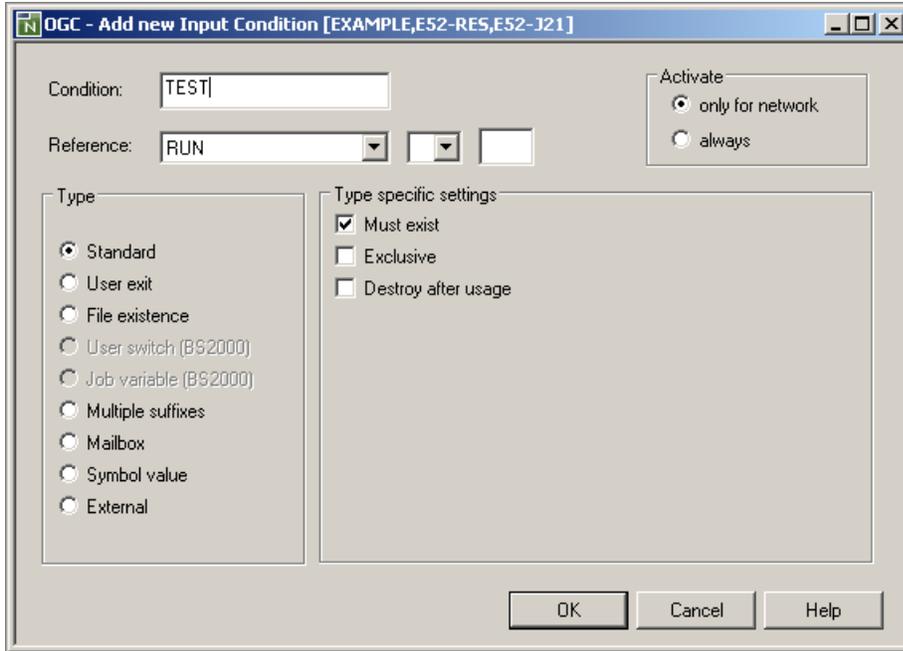
➤ To add or modify an input condition

- 1 On the [tabbed page Input Conditions](#), choose **Add** to create a new input condition.

Or:

On the [tabbed page Input Conditions](#) select an input condition from the table and choose **Modify** to change an input condition.

An **Input Condition** window similar to the example below opens:



- 2 Make your definitions. The fields and options in the window are described in *Fields and Selection Options: Input Condition*.
- 3 Choose **OK**.

The new input condition is now allocated to the job master.



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](#)

Fields and Selection Options: Input Condition

The input fields in the **Input Condition window** are described in the following table.

The fields provided in the **Type specific settings** section of the window depend on the **Reference** and **Type** selected for the condition.

Field	Description	
Condition	<p>Name assigned to the condition.</p> <p>The condition name and its reference date uniquely identify an active condition.</p> <p>See also Restrictions for Condition Names.</p>	
Run	Current run number (for active jobs only).	
Reference	<p>Reference date to specify which occurrence of this definition the job uses.</p> <p>For possible selection options, see Possible References for Input Conditions in the section Input Condition References.</p>	
Activate	<i>checked</i>	Input condition definition will always be activated (for job activations as well).
	<i>unchecked</i>	<p>Default.</p> <p>Input condition definition will be activated for network activations only.</p>
Type specific settings section:		
Must Exist	<i>checked</i>	Specifies that the condition must exist (be true) as a prerequisite to job submission.
	<i>unchecked</i>	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	<i>checked</i>	Specifies that when this condition is in use, no other job can access this condition until it is free (job finished).
	<i>unchecked</i>	<p>Any job can use the condition at any time. This feature is useful to prevent simultaneous execution of jobs with the same input conditions.</p> <p>Default.</p>
Destroy after usage	<i>checked</i>	Specifies that the condition is automatically reset after the job is submitted.
	<i>unchecked</i>	<p>Do not reset condition: later job runs can use this condition according to the Reference date.</p> <p>Default.</p>
Activate	<i>checked</i>	Input condition definition will always be activated (for job activations as well).
	<i>unchecked</i>	<p>Input condition definition will be activated for network activations only.</p> <p>Default.</p>

Field	Description
Type section:	
Standard	Select this radio button (default) to set the standard condition usage options (must exit, exclusive and/or destroy after usage).
User Exit	Select this radio button if the condition is to be set by a user exit. For further information and the fields/options available in the Type specific settings section, see Input Condition: User Exit .
File Existence	Select this radio button to define an input condition dependent on the existence or non-existence of a file. For further information and the fields/options available in the Type specific settings section, see Input Condition: File Existence .
User Switch (BS2000)	User switch (BS2000 only). Select this radio button to define an input condition dependent on the existence or non-existence of a user switch. For further information and the fields/options available in the Type specific settings section, see Input Condition: User Switch - BS2000 .
Job Variable (BS2000)	Job variable (BS2000 only). Select this radio button to define an input condition dependent on a comparison with the contents of a BS2000 job variable. For further information and the fields/options available in the Type specific settings section, see Input Condition: Job Variable - BS2000 .
Multiple Suffixes	Select this radio button to define a symbol to be used for the active condition name. For further information and the fields/options available in the Type specific settings section, see Input Condition: Multiple Suffixes .
Mailbox	Select this radio button to define a user prompt to a mailbox. For further information and the fields/options available in the Type specific settings section, see Input Condition: Mailbox .
Symbol Value	Select this radio button to define an input condition dependent on a comparison with the value of a symbol in a symbol table. For further information and the fields/options available in the Type specific settings section, see Input Condition: Symbol Value .
External	Select this radio button to define an input condition dependent on another network. See also Accessing another Network Defined for an Input Condition .

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 Master 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 .

Accessing another Network Defined for an Input Condition

➤ To access another network defined for an input condition

- From the table on the **tabbed page Input Conditions**, select an input condition of the **Type** *extern* and choose **Open Net.**

The **Maintenance Network Master** window of the network defined for the input condition opens.

If required, you can change the network definitions. The fields and tabs of the **Network Master window** are explained in *Fields: Network Definition* in the section *Modifying a Network Definition*.

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 select from the drop-down list box next to the **Reference** field of the [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 ABS (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.
Explicit date		A date entered in the format YYYYMMDD. 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>Defaults for Time Ranges</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>Defaults for Time Ranges</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.

In this case, additional fields appear next to the **Reference** field where you can select the required sign (+ or -) and enter a number.

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](#)
- [Example of a User Exit](#)

Defining and Editing an Input Condition of the Type User Exit

➤ To define an input condition that depends on a user exit

- 1 In the [Input Condition window](#), select **User exit** from the **Type** section.

In the **Type specific settings** section, 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](#).

- 2 When finished, choose **OK** to save your entries and close the window.

➤ To edit the user exit of an input condition

- 1 From the table on the [tabbed page Input Conditions](#), select an input condition of the type `Exit` and choose **Edit**.

An editor window opens containing the source of a Natural subprogram similar to the [Example of a User Exit](#).

(If no user exit is specified for the selected input condition, an appropriate message occurs instead.)

- 2 Modify the user exit as required.

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 **Type specific settings** section of the **Input Condition** 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 .

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 **Input Condition window**, select **Multiple suffices** from the **Type** section.
- 2 In the **Type specific settings** section, enter a symbol name and select a symbol table/version, if required. See also *Fields: Input Condition Multiple Suffixes*.
- 3 When finished, choose **OK** to save your entries and close the window.

This section covers the following topics:

- [Fields: Input Condition Multiple Suffixes](#)

Fields: Input Condition Multiple Suffixes

The input fields provided for multiple suffixes in the **Type specific settings** section of the **Input Condition window** are described in the following table:

Field	Description			
Always Job Table	Specifies whether the symbol table defined for the job is used.			
	Possible check box settings:			
	<table border="1"> <tbody> <tr> <td><i>checked</i></td> <td>The multiple suffix is always taken from the job symbol table. A local definition will be ignored (default).</td> </tr> <tr> <td><i>unchecked</i></td> <td>Use the symbol table defined here.</td> </tr> </tbody> </table>	<i>checked</i>	The multiple suffix is always taken from the job symbol table. A local definition will be ignored (default).	<i>unchecked</i>
<i>checked</i>	The multiple suffix is always taken from the job symbol table. A local definition will be ignored (default).			
<i>unchecked</i>	Use the symbol table defined here.			

Field	Description	
	Note: In the case of a job or network copy, it is recommended to select this check box. 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.	
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 check box settings:	
	<i>checked</i>	The multiple suffix is always taken from the job symbol table. A local definition will be ignored (default).
	<i>unchecked</i>	Use the symbol table defined here.
	Note: In the case of a job or network copy, it is recommended to select this check box. 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.	

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 **Input Condition window**, select **File existence** from the **Type** section.
- 2 In the **Type specific settings**, enter a file and a member name. The input fields and options are described in *Fields: Input Condition File Existence*.
- 3 When finished, choose **OK** to save your entries and close the window.

This section covers the following topics:

- [Fields: Input Condition File Existence](#)

- [Rules for File Names and File Checking](#)

Fields: Input Condition File Existence

The input fields provided for a file existence check in the **Type specific settings** section of the **Input Condition window** are described in the following table:

Field	Description				
File	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> . Note: When entering a file name, remember to observe the rules for upper and lower case which are specific to some operating systems. See also Rules for File Check .				
Member (optional)	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. Note: Only specify a member where necessary and possible. If this field is left blank, the existence of the whole file is checked. See also Rules for File Check .				
Must exist	Possible check box settings:				
	<table border="1"> <tr> <td><i>checked</i></td> <td>The file (or member) must exist as a prerequisite to job submission.</td> </tr> <tr> <td><i>unchecked</i></td> <td>The file (or member) must not exist as a prerequisite.</td> </tr> </table>	<i>checked</i>	The file (or member) must exist as a prerequisite to job submission.	<i>unchecked</i>	The file (or member) must not exist as a prerequisite.
<i>checked</i>	The file (or member) must exist as a prerequisite to job submission.				
<i>unchecked</i>	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.

Input Condition: Mailbox

Mailboxes are defined to the system and assigned to user IDs by using the appropriate context function of the **Mailbox Definition** metanode. 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](#)

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 asks a user to confirm completion of the action(s).

Each user linked to this mailbox sees this message.

A user can be associated with up to ten mailboxes.

Defining an Input Condition of the Type Mailbox

➤ **To send a message to a mailbox for an input condition that is not satisfied during network execution**

- 1 In the **Input Condition window**, select **Mailbox** from the **Type** section.
- 2 In the **Mailbox** field of the **Type specific settings** section, enter the name of the mailbox to which you want to send a message.

Or:

Select a name from the drop-down list box next to the **Mailbox** field.

- 3 When finished, choose **OK** to save your entry and close the window.

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

Defining an Input Condition of the Type Symbol Value

➤ To define an input condition that depends on a symbol value

- 1 In the **Input Condition window**, select **Symbol value** from the **Type** section.

In the **Type specific settings** section you can enter the symbol to be compared and further parameters to specify the symbol.

The input fields available are described in *Fields: Input Condition Symbol Value*.

- 2 When finished, choose **OK** to save your definitions and close the window.

Fields: Input Condition Symbol Value

The input fields of the **Type specific settings** for the type **Symbol value** in the **Input Condition window** are described in the following table:

Field	Description										
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.										
Symbol table	Enter the name of a valid symbol table. 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. See also <i>Symbol Table Types and Symbol Search Order</i> in the section <i>Symbol Table and Symbol Maintenance</i> .										
Version	Symbol Table Version. Possible version names: <table border="0" style="width: 100%;"> <tr> <td style="padding-left: 20px;">(current)</td> <td>Current version at determination date.</td> </tr> <tr> <td style="padding-left: 20px;">(nv)</td> <td>Same version as network version.</td> </tr> <tr> <td style="padding-left: 20px;">(svn)</td> <td>Symbol table version of network.</td> </tr> <tr> <td style="padding-left: 20px;">(svj)</td> <td>Symbol table version of job.</td> </tr> <tr> <td style="padding-left: 20px;">(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).										
Position	Enter the position of symbol substring to be checked. Possible values: 1 through 80. (Checked only if Format is set to A.)										

Field	Description												
Length	Enter length of symbol substring to be checked. Possible values: 1 through 80. (Checked only if Format is set to A.)												
Format	Specify the format in which the symbol substring is to be checked against the comparison string. Possible selection options: <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20%;">A</td> <td>Alphanumeric.</td> </tr> <tr> <td>D</td> <td>Date in the format YYYYMMDD. 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).												
Symbol is	This is the comparison operator. Specify a logical operator for the comparison of the defined symbol against the comparison string (see below). Possible selection options: <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20%;">= 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.												
compared to <i>comparison string</i>	Comparison string. In the input lines below compared to , enter the string or field to be compared with the symbol (substring). The strings are compared in the defined Format . Symbol replacement is possible in this field. <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.

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](#)

Defining an Input Condition of the Type User Switch

» To define an input condition that depends on a user switch

- 1 In the **Input Condition window**, select **User switch (BS2000)** from the **Type** section.

In the **Type specific settings** section, enter a user switch and BS2000 user ID.

The input fields available are described in [Fields: Input Condition User Switch](#).

- 2 When finished, choose **OK** to save your entries and close the window.

Fields: Input Condition User Switch

The input fields of the **Type specific settings** for the type **User switch (BS2000)** in the **Input Condition window** are described in the following table:

Field	Description		
User Switch	Number of a user switch.		
BS2000 User ID	BS2000 user ID to which the specified user switch belongs.		
Must exist	Possible check box settings:		
	<table border="1"> <tr> <td><i>checked</i></td> <td>The user switch must exist as a prerequisite to job submission.</td> </tr> </table>	<i>checked</i>	The user switch must exist as a prerequisite to job submission.
<i>checked</i>	The user switch must exist as a prerequisite to job submission.		

Field	Description
	<i>unchecked</i> The user switch must not exist as a prerequisite.

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 \(BS2000\)](#)
- [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 **Input Condition window**, select **Job variable (BS2000)** from the **Type** section.

In the **Type specific settings** section, enter a job variable and further parameters to specify the input condition.

The input fields are described in [Fields: Input Condition Job Variable \(BS2000\)](#).

- 2 When finished, choose **OK** to save your entries and close the window.

Fields: Input Condition Job Variable (BS2000)

The input fields provided for a job variable in the **Type specific settings** section of the **Input Condition window** are described in the following table:

Field	Description
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.

- 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.

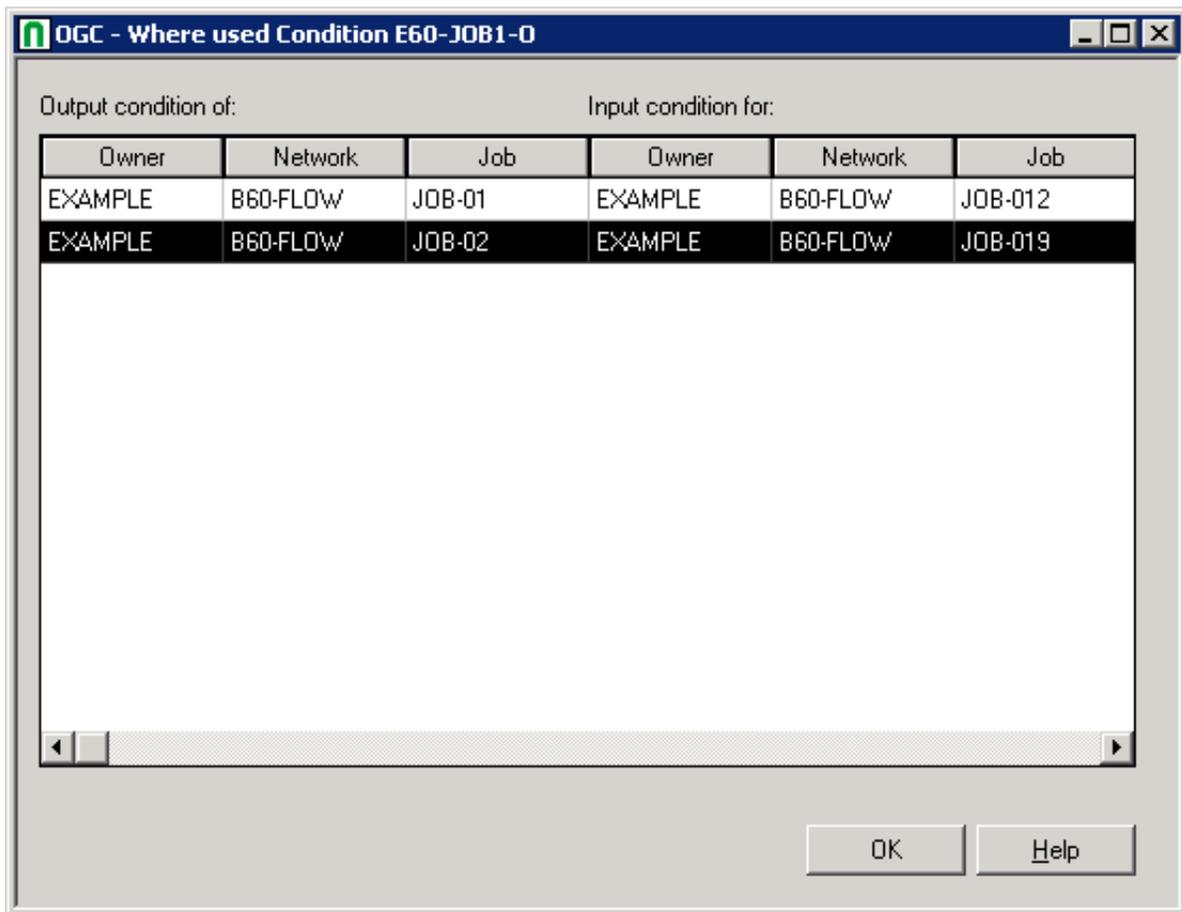
Listing Jobs Linked to an Input Condition

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 a condition

- From the table on the [tabbed page Input Conditions](#), select an input condition and choose **Where Used**.

A **Where used Condition** window similar to the example below opens:



The window displays the name of the selected condition and two lists of jobs:

- one lists the jobs which use the condition as input condition;
- the other lists the jobs which use the condition as output condition.

The jobs are listed according to owner, network and job name.



Note: For an active input condition, a similar window opens: see the [Active Usage Condition](#) window described in the section *Active Job Networks*.

Deleting an Input Condition Definition

> To delete an input condition

- 1 On the [tabbed page Input Conditions](#), select an input condition in the table and choose **Delete**.
- 2 Select an input condition in the table.
- 3 Choose the **Delete** button.

The input condition is now deleted for this job master.

26 Handling Prerequisite Resources for a Job

- Listing Prerequisite Resources Defined for a Job 350
- Viewing the Usage of a Prerequisite Resource 351
- Displaying, Modifying and Adding a Prerequisite Resource Definition 352
- Deleting a Prerequisite Resource Definition 353
- Columns and Fields: Prerequisite Resource Definitions 354

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 maintenance functions of the **Resource Master** metanode 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 all prerequisite resources defined for a job

- In the **Maintenance Job Master window**, open the tabbed page **Resources**:

OGC - Maintenance Job Master JOB-06 [EXAMPLE.B60-FLOW]

Job name: Use symbol as execution node

Execution node:

Description:

Job type: Special type:

Scheduling Parameters | User Defined Log Data | Long Description | OS Specials

Main | Resources | Input Conditions | EOJ Checking | JCL Definition

Resource	T	Quantity	D	DNO	Allocated
HUGO	Quantitative, n	5.00	After job	Y	
PARA-1	Quantitative, re	10.00	After ne	Y	
WILLI	Quantitative, n	20.00	Keep u	Y	

Add
Modify
Delete
Master
Where used

Apply OK Cancel Help

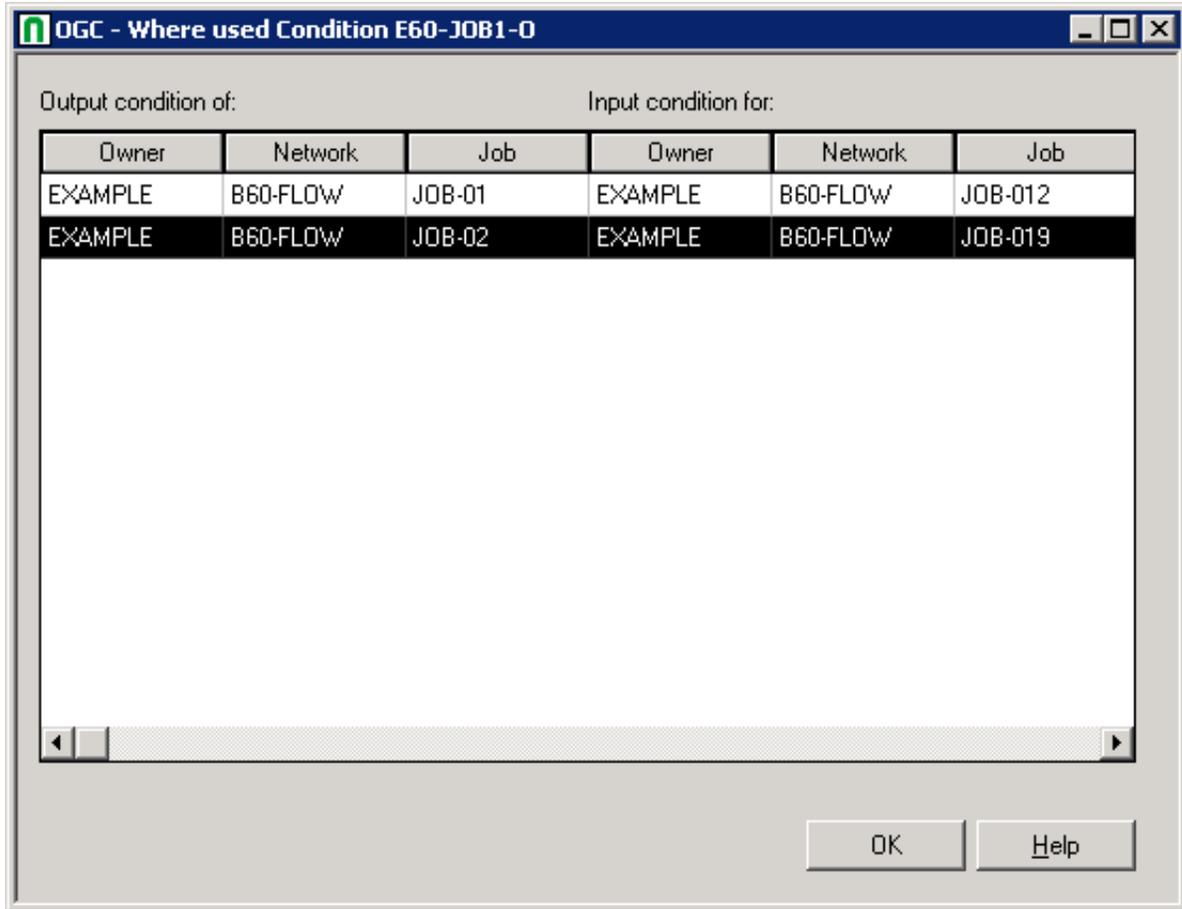
The table in the window lists all resources defined as prerequisites for the job. The table columns are described in [Columns and Fields: Prerequisite Resource Definitions](#).

Viewing the Usage of a Prerequisite Resource

➤ To find out where a predefined resource is used

- 1 From the table on the **Resources** page, select the required resource.
- 2 Choose **Where Used**.

A **Where used Condition** window similar to the example below opens:



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.

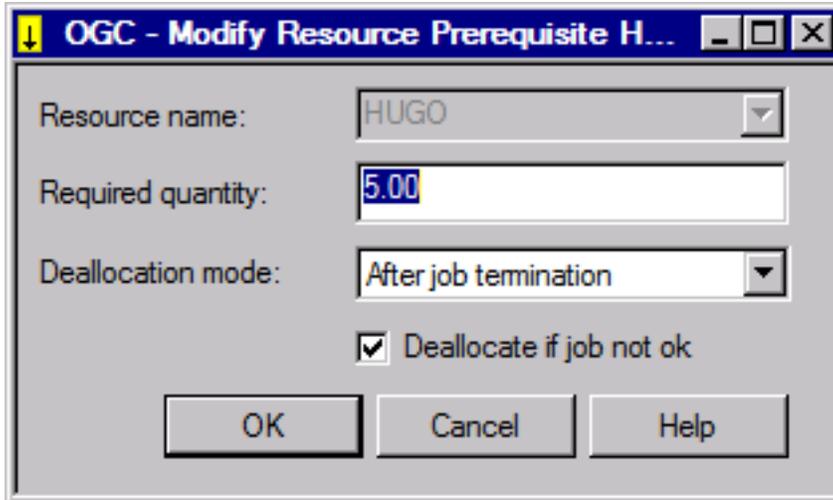
The fields and columns contained in the window correspond to the fields and columns of the **Defined in Jobs** page. They are described in *Field and Columns: Resource Defined in Jobs* in the section *Resources* in the *Administration* documentation.

Displaying, Modifying and Adding a Prerequisite Resource Definition

➤ To display or modify the prerequisite resource definition of a job

- 1 From the table on the **tabbed page Resources**, select the resource definition you want to view or change and choose **Modify**.

A **Resource Prerequisite** window similar to the example below opens:



- 2 Make your definitions. The input fields and selection options available in the window are described in *Columns and Fields: Prerequisite Resource Definitions*.
- 3 Choose **OK** when you are finished.

➤ **To add a prerequisite resource definition for a job**

- 1 On the **tabbed page Resources**, choose **Add**.

A **Resource Prerequisite window** similar to the previous example opens.

- 2 Make your definitions. The input fields and selection options available in the window are described in *Columns and Fields: Prerequisite Resource Definitions*.
- 3 Choose **OK** when you are finished.

Deleting a Prerequisite Resource Definition

➤ **To delete a resource definition**

- From the table on the **tabbed page Resources**, select the resource definition you want to remove and choose **Delete**.

The selected resource is deleted for this job master.

Resource Deallocation Modes

 **Note:** Resource allocations will be deallocated if the retention period for an active condition is reached.

Mode/Selection Option	Explanation
J	Release resource at job termination (default).
After job termination	The resource is released when the allocating job terminates.
N	Release resource at network termination.
After network termination	<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 <code>network ended ok</code>, you must set the reserved condition <code>NET-END-OK</code> at least once in your network.</p>
E	Release resource at network termination, but release earlier after an erroneous job.
Keep until network termination	The resource will be kept until network termination. If any job ends <code>not ok</code> , it will be released immediately after this job's termination.
K	Keep resource until manual release.
Keep until explicit release	<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 **Keep until explicit release** 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*.

VIII

■ 27 End-of-Job Checking and Actions	359
■ 28 End-of-Job Checking and Actions - Overview	361
■ 29 Adding and Modifying Events and Actions	365
■ 30 End-of-Job Checking and Actions: Columns EOJ Checking Page	369
■ 31 Adding Event Definition for z/OS or z/VSE Job	371
■ 32 Modifying Event Definition for z/OS or z/VSE Job	375
■ 33 Adding Event Definition for BS2000 Job	381
■ 34 Modifying Event Definition for BS2000 Job	385
■ 35 Adding Event Definition for UNIX or Windows Job	393
■ 36 Modifying Event Definition for UNIX or Windows Job	397
■ 37 EOJ Checking Defaults for Various Operating Systems	401
■ 38 Adding Output Condition Definitions	405
■ 39 Define Symbol Modification or Setting	409
■ 40 Editing End-of-Job Checking or Action User Exits	411
■ 41 Defining Activation of Jobs or Job Networks	415
■ 42 Defining Recovery Action	417
■ 43 SYSOUT Actions	421
■ 44 Message Sending	423
■ 45 Defining End-of-Job Action User Exit	429
■ 46 End-of-Job Action: Release Resource	431

27

End-of-Job Checking and Actions

[End-of-Job Checking and Actions - Overview](#)
[Adding and Modifying Events and Actions](#)
[End-of-Job Checking and Actions: Columns EOJ Checking Page](#)
[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](#)
[EOJ Checking Defaults for Various Operating Systems](#)
[Adding Output Condition Definitions](#)
[Define Symbol Modification or Setting](#)
[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](#)

Related Topics:

- *End-of-Job Checking and End-of-Job Actions in Concepts and Facilities.*
- *Passing Files to Entire Output Management*

28

End-of-Job Checking and Actions - Overview

- List of End-of-Job Checking and Actions 362
- Kinds of End-of-Job Actions 363
- End-of-Job Checks for Various Operating Systems 364

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

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.

29 Adding and Modifying Events and Actions

You can add, modify and delete an End-of-Job checking for a job master and define a user message for the check. Furthermore, you can define an output condition for the End-of-Job checking.

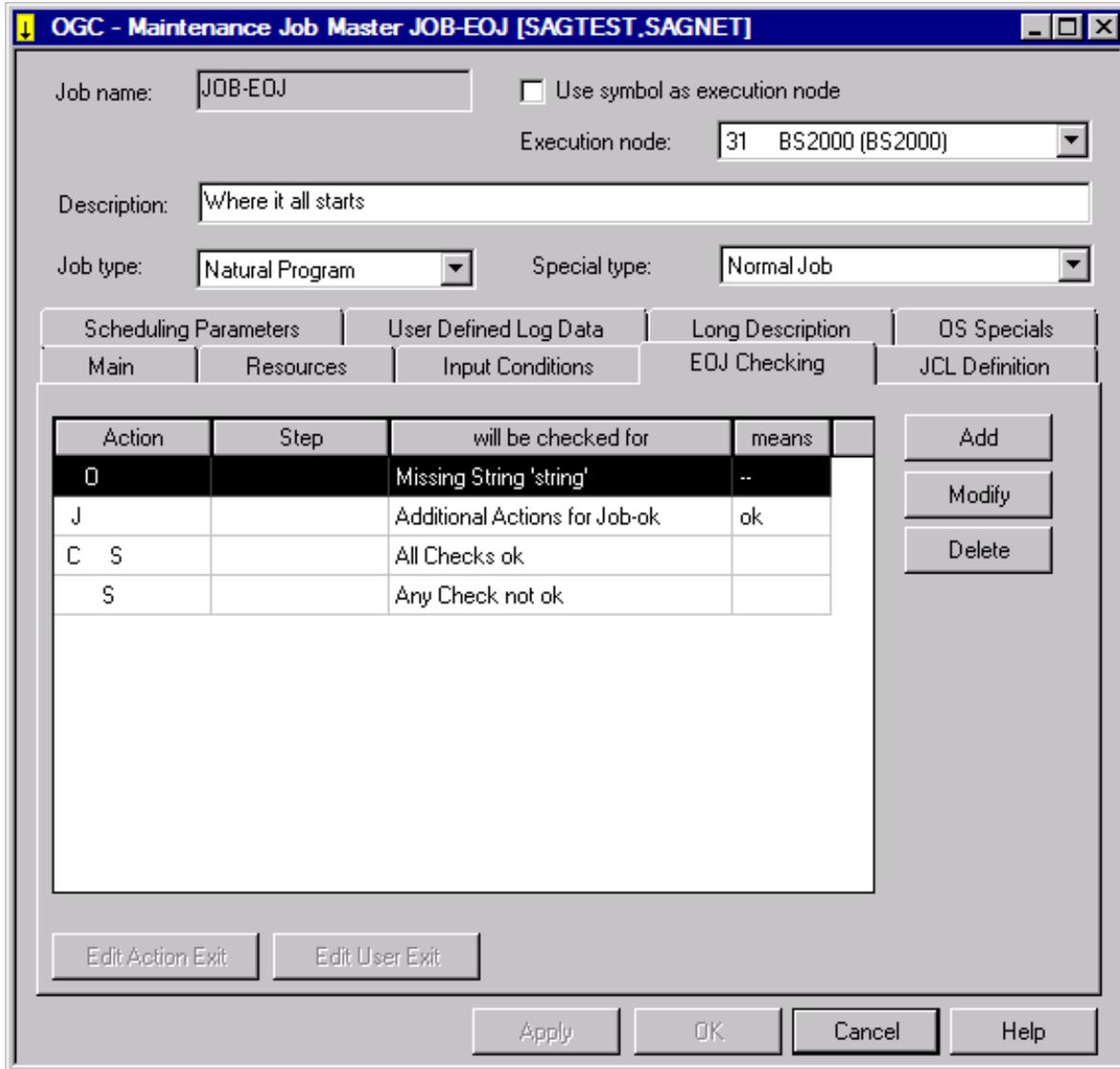
This feature is provided on the **tabbed page EOJ Checking** of the **Job Master** window.

For further information, see *Columns: End-of-Job Checking and Actions* in the section *End-of-Job Checking and Actions Facility*.

➤ To add (modify) an End-of-Job (EOJ) checking

- 1 Select the **EOJ Checking** tab.
- 2 Select the **Add** button.

An **EOJ Checking** page similar to the example below opens:



3 Make your definitions.

 **Note:** For further information, see *Field Descriptions: Job/Network Activation* in the section *Defining Activation of Jobs or Job Networks*.

4 Select **OK**.

The new check is defined.

➤ **To add an output condition to the End-of-Job (EOJ) checking**

- 1 Select the **EOJ Checking** tab.
- 2 Select the **Add** button.

An **Add new EOJ Checking** window opens.

- 3 Select the **Output Condition** tab.
- 4 Select the **Add** button.

The following dialog appears:



- 5 Make your definitions.



Note: For further information, see [Fields and Columns: Output Conditions](#).

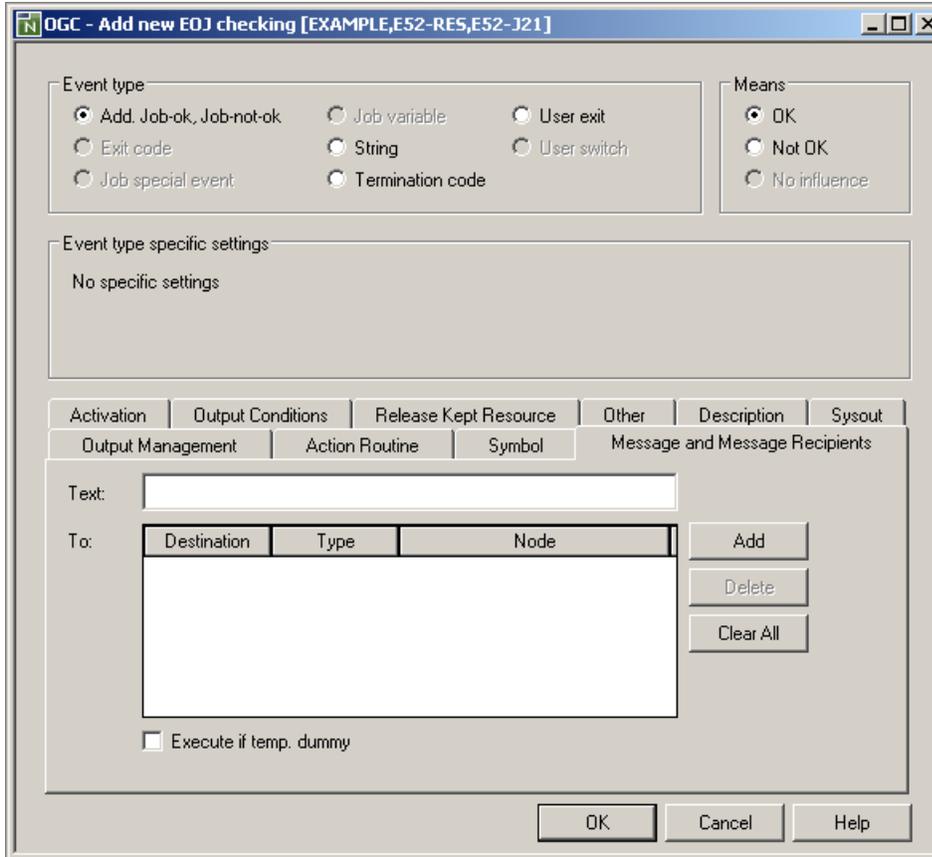
- 6 Select **OK**.

The new output condition is defined.

➤ To create a user message for End-of-Job checking

- 1 Select the **EOJ Checking** tab.
- 2 Select an EOJ Checking and select **Modify** button.
- 3 Select the **Message and Message Recipients** tab.

An **EOJ checking** window similar to the example below opens.



4 Insert a text and a destination for the message.

 **Note:** For further information, see *Fields and Columns: Message and Message Recipients* in the section *Message Sending*.

5 Select **OK**.

The new user message is defined.

➤ **To add an End-of-Job checking (EOJ) for Entire Output Management**

- Proceed as described in the section *Passing Files to Entire Output Management*.

30

End-of-Job Checking and Actions: Columns EOJ Checking

Page

■ Columns: End-of-Job Checking and Actions	370
--	-----

Columns: End-of-Job Checking and Actions

The columns on the **EOJ Checking page** of the **Maintenance Job Master** or **Maintenance Job Active** window are described in the following table:

Column	Description		
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	This column only appears on the EOJ Checking page in the Maintenance Job Active window of an active job.		
	This column only contains entries if an event occurred during job execution.		
	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:		
	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.	

31 Adding Event Definition for z/OS or z/VSE Job

- Field Descriptions: Add/Modify Event Definition 372

See also [z/OS Defaults for Event Checking](#) and [z/VSE Defaults for Event Checking](#).

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	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.
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.

Field	Description
	>> 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 C-type event) Specify a condition code to be compared with the return code. Possible values:
	Cxxxx Condition code xxxx (z/OS) Return code xxxx
	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:

Field	Description
	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 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.
	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 in the section Job Maintenance).	

32

Modifying Event Definition for z/OS or z/VSE Job

- Example of an A-type Event 376
- Example of a C-type Event 377
- Example of an R-type Event 378
- Example of an S-type Event 379

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](#) 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: Columns EOJ Checking Page* (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.

33

Adding Event Definition for BS2000 Job

- Field Descriptions 382

See also BS2000 Defaults for [Event Checking](#).

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.	

Field	Description	
(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.	
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: Columns EOJ Checking Page* 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: Columns EOJ Checking Page*. The new events are listed on this screen.

34

Modifying Event Definition for BS2000 Job

- Example of an S-type Event 386
- Example of a U-type Event 387
- Example of a V-type Event 388
- Adding or Modifying a Job Variable 390
- Example of a J-type Event 390

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 ! OA
_ !      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 !                                                    ! her
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.

Example of a V-type Event

Field Descriptions: Event Definition

(BS2000)

```

24.10.08          ***** Entire Operations *****          14:09:01
Ow +-----+
Ne !
-- !              Add Event Definition                          ! ---
C !              BS2000                                         !  OA
_ !      Event Type ==> v  R  User Exit          J  Job special event  !
_ !      +-----+
_ !      !
_ !      !              Event Definition: Job Variable Checking      ! !
_ !      !              BS2000                                         ! !
_ !      User   !   If Contents of Job Variable                       ! !
_ !      !   $DC1.JV.DEMO_____                                     ! !
_ !      Strin !   at Position 10_ in Length 20_ with Format A       ! !
** !   in Fi  !   is EQ                                             ! ! ***
_ !      !   Result is 100_____                                     ! !
_ !      !   _____                                             ! !
-- !   User   !   (Optional) Read Password ==>                       ! ! ---
A !      !
P !   Occur !   The successful Check means OK or NOT OK ==> OK     ! ! Rtn
!   !   Enter-PF1-----PF3-----PF5-----                       ! !
Co !   Enter- !   Help          End          Save                     ! !
Ent !      +-----+
+-----+
  
```

Explanation:

The event is ok if the job variable \$DC1.JV.DEMO contains:

```
Result is 100
```

at position 10 in length 20 in alphanumeric format.

Field Descriptions: Event Definition

Job Variable Checking - BS2000

Meaning of the input fields:

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. Symbol replacement is possible. The submission escape character is to be used.	
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
is xx	This is the comparison operator. Enter a logical operator for the comparison of the defined BS2000 job variable substring against the comparison string below. 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.
(Comparison string)	Enter the string or field to be compared with the job variable substring. 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. The comparison is made in the defined format. Symbol replacement is possible. The submission escape character is to be used.	
(Optional) Read Password	If the job variable is read password-protected, specify the password here.	

Adding or Modifying a Job Variable

- [Field Descriptions: BS2000 Job Variable](#)

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.	

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 !
_ !      +-----+ b-not-ok !
_ !      !
_ !      !          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----- !          ! her
P !      Occur !          Help      End      Save      !          ! Rtn
_ !      !      +-----+
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.

35 Adding Event Definition for UNIX or Windows Job

- Field Descriptions 394
- Exit Code Check Error Handling 395

See also [UNIX and Windows Defaults for Event Checking](#).

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).

Field	Description	
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.	
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.

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*.

36

Modifying Event Definition for UNIX or Windows Job

- Example of an S-type Event 398

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:								
	<table border="1"> <tr> <td>A</td> <td>Additional definition for job ok or job not ok. See Example of an A-type Event.</td> </tr> <tr> <td>J</td> <td>Job special event.</td> </tr> <tr> <td>R</td> <td>Job SYSOUT is to be checked by a user exit. See Example of an R-type Event.</td> </tr> <tr> <td>S</td> <td> 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 </td> </tr> </table>	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
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								

Field	Description	
		<p>SYSOUT file exists. This case will apply, if the job is submitted as a temporary dummy job.</p> <p>2. A string search in a non-existent file is handled like "string not found" in an existing file.</p>
	X	Exit code.
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 <i>Editing End-of-Job User Exits</i>.</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>	

Field	Description	
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 <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>	
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 in the section Job Maintenance).</p>		

37

EOJ Checking Defaults for Various Operating Systems

▪ Common Defaults for Event Checking	402
▪ z/OS Defaults for Event Checking	402
▪ z/VSE Defaults for Event Checking	403
▪ BS2000 Defaults for Event Checking	403
▪ UNIX and Windows Defaults for Event Checking	404

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*.

38

Adding Output Condition Definitions

- Fields and Columns: Output Conditions 406

You can define output conditions for an event or a job.

Fields and Columns: Output Conditions

- [Field Descriptions: Output Conditions](#)
- [Column Headings: Output Conditions](#)

Field Descriptions: Output Conditions

Field	Description
Condition	Condition name. Symbol replacement is possible. The submission escape character is to be used.
	<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	An output condition can be set with different references (usually the current network run number). Possible values:
	RUN Current run number of the job network is assigned. This is the default.
	ABS Absolute condition. Exists only once, because it is independent of run numbers.
	RCM Sets an active condition with owner, network, run number of calling network and appends multiple suffixes.
	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 .
	The reference is evaluated and set when the active condition is created by the Monitor according to the End-of-Job definition.
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 selection options:

39 Define Symbol Modification or Setting

- Field Descriptions: Symbol Modification 410

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.

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.
at Position	Enter position of symbol substring to be set. Possible values: 1 through 80. Default: 1.
in Length	Enter length of symbol substring to be set. Possible values: 1 through 80. Default: 80. Note: The effective value length may be shortened do to this definition.
with Format	Enter format in which the symbol substring is to be set. Possible values:
	A alphanumeric
	L alphanumeric (lower case)
	U alphanumeric (upper case)
to Value	N numeric
	The value to be set into the complete symbol or a substring of the symbol. Symbol replacement is possible. If the effective value length (after symbol replacement) is longer than in Length, the value will be shortened to this length.
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.

40

Editing End-of-Job Checking or Action User Exits

- Creating or Editing User Exits 412
- Example 412

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: Columns EOJ Checking Page*. 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
```

41 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*.

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.
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.



Notes:

1. **Wildcard Usage**
2. 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.

3. 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.
4. If you leave the Job field blank, the whole specified network is activated.
5. **Authorization**
6. 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*).

42 Defining Recovery Action

- Recovery Action Definition 418
- System Symbols for Recovery Actions 419

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](#)

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.

Field	Description	
	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 <code><jobname>-MAX-RETRY</code> 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.	

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.

43

SYSOUT Actions

- Defining Job SYSOUT Actions 422

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

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.

44

Message Sending

- Introduction 424
- Fields and Columns: Message and Message Recipients 424
- Message Destination Types 425
- Message Recipients 426
- E-Mail on Mainframes 426
- E-Mail on UNIX and Windows Systems 427

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.

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 Message Destination Types 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>

Field	Description
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> <pre>user(a)any.host</pre> <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).

Type	Description
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](#).

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.

45

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.

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.	

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*.

46

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.

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.

IX

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.

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

47

Maintaining Active Job Networks

- Available Functions: Network Active 436
- Listing Active Job Networks 436
- Deactivating Active Job Networks 437
- Viewing the Execution History of a Network Active 439
- Maintaining Active Runs 440

Available Functions: Network Active

➤ To list all functions available for a Network Active node

- In the object workspace, select a **Network Active** node and open the context menu.

The following functions are available:

Function	Shortcut	Description
List	F8	See Listing Active Job Networks .
Refresh	F5	See Refreshing Object Lists .
Filter	F3	See Filtering Objects .
Set Drag And Drop Function	---	See Drag & Drop .

➤ To list all functions available for a Network Active instance

- Select a **Network Active** instance and open the context menu.

The following functions are available:

Function	Shortcut	Description
History	CTRL+C	See Viewing the Execution History of a Network Active .
List Active Jobs	---	See Listing Active Jobs .
Browse Log	---	See Displaying Logged Information - Browse Log Function .
Set Drag And Drop Function	---	See Drag & Drop .

Listing Active Job Networks

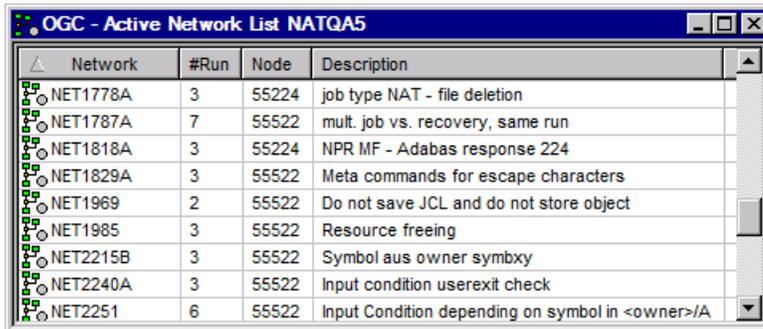
➤ To list all active job networks of an owner

- 1 In the object workspace, select the **Network Active** node of an owner.
- 2 From the context menu, choose **List**.

Or:

Press F8.

An **Active Network List** window similar to the example below opens:



Network	#Run	Node	Description
NET1778A	3	55224	job type NAT - file deletion
NET1787A	7	55522	mult. job vs. recovery, same run
NET1818A	3	55224	NPR MF - Adabas response 224
NET1829A	3	55522	Meta commands for escape characters
NET1969	2	55522	Do not save JCL and do not store object
NET1985	3	55522	Resource freeing
NET2215B	3	55522	Symbol aus owner symbxy
NET2240A	3	55522	Input condition userexit check
NET2251	6	55522	Input Condition depending on symbol in <owner>JA

- [Columns: Active Job Networks](#)

Columns: Active Job Networks

The following table explains the column headings for the data listed in the [Active Network List window](#):

Field	Description
#Run	Number of currently active runs of the network.
Network	Network name.
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.

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](#)

- [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.

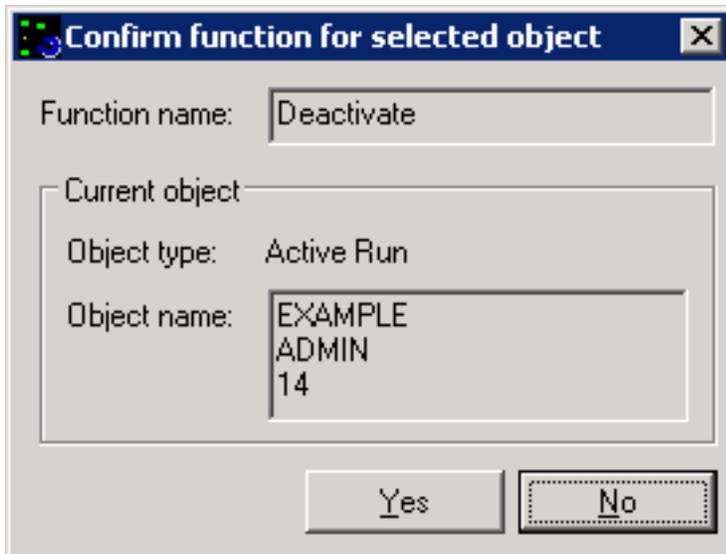
Deactivating Active Runs for a Network

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 single or multiple active runs

- 1 In the object workspace, select an **Active Run** and choose **Deactivate** from the context menu.

A confirmation dialog similar to the example below opens:



- 2 Choose **Yes**.

The currently selected active run is deactivated.

» To deactivate multiple active runs

- 1 In the object workspace, select an **Active Run** node and choose **List** from the context menu.

An **Active Run List** window opens with a table of all active runs.

- 2 Select multiple active runs and choose **Deactivate** from the table's context menu.

In the **confirmation dialog** that opens, choose **Yes** to deactivate one active run after the other, or choose **Yes to All** to delete all active runs at once.

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.

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](#)).

Viewing the Execution History of a Network Active

➤ To view the history of a Network Active instance

- 1 In the object workspace, select a **Network Active** instance and choose **History** from the context menu.

A **History Network Active** window opens for the selected network. This window corresponds to the **History Network Master window** described in [Displaying the Network Execution History](#) in the section *Schedule Maintenance*.

- 2 In the calendar, click on the scheduled date for which you want to view the history data.

If history data exists for the selected network and date, execution information on all network starts is contained in the right-hand table columns of the window. The columns are explained in [Columns: Network History](#) in the section *Schedule Maintenance*.

Maintaining Active Runs

An active run is identified by owner, network and run number. Run numbers are unique per network. If a single job is activated, it will get a run number of the network to which it belongs.



Note: You can maintain active runs from either the **Active Run** node under **Network Active** or the **Active Run** node under **Network Master**. Both nodes control the same active runs. If you use both nodes to change active runs during a session, consider refreshing them manually to make sure that you view the latest status.

This section covers the following topics

- [Available Functions: Active Run](#)
- [Using Filter Criteria to List Active Runs in a Network](#)
- [Viewing a Network Diagram](#)
- [Repeating an Active Run](#)
- [Modifying the Latest Start Time for an Active Run](#)

Available Functions: Active Run

➤ To list all functions available for an Active Run node

- In the object workspace, select an **Active Run** node and open the context menu.

The following functions are available:

Function	Shortcut	Description
List	F8	List all active runs in a network: see Listing Objects .
Refresh	F5	See Refreshing Object Lists .
Filter	F3	See Using Filter Criteria to list Active Runs in a Network .
Set Drag And Drop Function	---	See Drag & Drop .

➤ To list all functions available for an Active Run instance

- Select an **Active Run** instance and open the context menu.

The following functions are available:

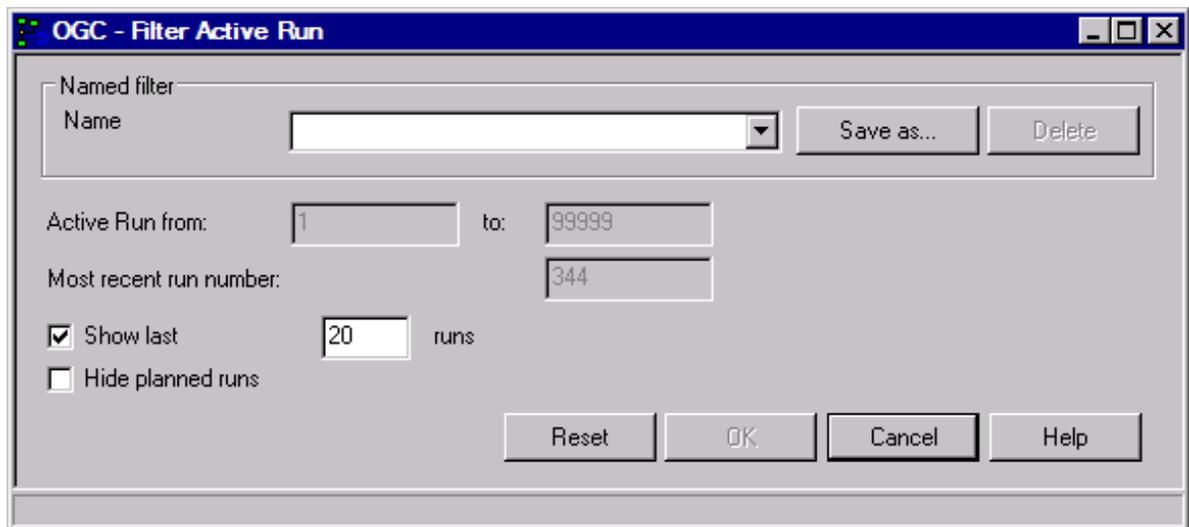
Function	Description
Diagram	See Viewing a Network Diagram .
Deactivate	See Deactivating Active Runs for a Network .
Release	Release job(s) from hold status for the selected active run. See Releasing an Active Job .
Repeat	See Repeating an Active Run .
Modify Latest Start	See Modifying the Latest Start Time for an Active Run .
List Active Jobs	See Listing Active Jobs .
Browse Log	See Displaying Logged Information - Browse Log Function .
Add to Workplan	See Add to Workplan .
Set Drag And Drop Function	See Drag & Drop .

Using Filter Criteria to List Active Runs in a Network

➤ To filter active runs

- 1 Select an **Active Run** node and choose **Filter** from the context menu.

A **Filter Active Run** window similar to the example below opens:



- 2 In the **Show last** input field, enter the number of latest active runs you want to view.

Or:

Remove the mark from the **Show last** check box and enter the range of run numbers you want to view in the **Active Run from/to** input fields (default is 1 to 99999). The most recent run number is indicated for orientation.

- 3 In addition, you can mark the **Hide planned runs** check box to hide planned runs from the display list.

You can set default values for an active run filter in the your user profile by using the **Filter** option (see the *Administration* documentation).

For information on naming a filter, see [Filtering Objects](#).

Viewing a Network Diagram

Diagrams are provided for network masters and active networks. In case of active networks, they can be used to define an active run within a network and to monitor the progress within the network.

› To view a network diagram

- 1 In the object workspace, select an instance of **Active Run** and choose **Diagram** from the context menu.

Or:

In the object workspace, select an instance of **Job Active** and choose **Network Diagram**.

- 2 Open the context menu and choose **Diagram** or **Network Diagram**, respectively.

A **Diagram Active Run window** with a network diagram opens similar to the example shown in the [Network Maintenance](#) section.

Here you can see all network components available for the currently selected **Active Run** in a graphical view.

For explanations of the symbols used in the diagram and the functions provided, see [Explanations of Diagram Symbols](#) and [Maintenance Functions for Diagrams](#) in the section *Network Maintenance*.

Repeating an Active Run

› To repeat an Active Run

- 1 In the object workspace, select an instance of **Active Run**.
- 2 Invoke the context menu and select **Repeat**.



- 3 You can adapt job time frames for the repetition.

If you select this option, then the time frames are 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

- 4 Choose **OK**.

The active run is repeated immediately.

Modifying the Latest Start Time for an Active Run

You can change the latest start time for all jobs of an active run where the active job status permits it. The latest start time may be changed for an active job that is not started or is already terminated.

➤ To modify the latest start for an active run

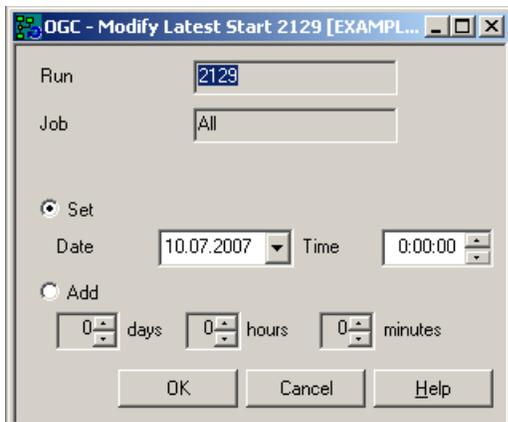
- 1 In the object workspace, select an instance of **Active Run**.
- 2 Invoke the context menu and select **Modify Latest Start**.

A window similar to the following opens:

You may choose only one of two methods to modify the latest start time:

- **Set:** Enter date and time values of new latest start;
- **Add:** Enter days, hours and minutes that shall be added to the latest start time value of active job.

The deadline value may get changed during this modification to ensure the following rule is honored: latest start time + estimated elapsed time <= Deadline.



All actions are logged in the Entire Operations log file.

48

Maintaining Active Jobs

- Available Functions: Job Active 446
- Listing Active Jobs 448
- Listing Jobs of an Active Subnetwork 453
- Adding a New Job to the Active Network 454
- Modifying the Latest Start Time for an Active Job 456
- Choosing the Job ID of a Job for Logging 457
- Canceling, Holding and Releasing Active Jobs 458
- Resubmitting an Active Job 460
- Deactivating a Job in an Active Network 462
- Reactivating an Active Job 465
- Displaying and Modifying an Active Job Definition 465
- Modifying EOJ Checking and Actions 467
- Viewing Long Descriptions of Active Jobs 468
- Displaying Prerequisites for Active Jobs: Waiting for 468
- Viewing and Modifying Resources Used by Active Jobs 470
- Browsing Active Job SYSOUT 471

Available Functions: Job Active

➤ To list all functions available for a Job Active node

- In the object workspace, select a **Job Active** node and open the context menu.

The following functions are available:

Function	Shortcut	Description
List	F8	See Listing Active Jobs .
New	CTRL+N	Creating a new Job Active : see Adding a New Job to the Active Network
Refresh	F5	See Refreshing Object Lists .
Filter	F3	Using selection criteria to list active jobs: see Filtering Objects .
Set Drag And Drop Function	---	See Drag & Drop .

➤ To list all functions available for a single Job Active instance

- Select a **Job Active** instance and invoke the context menu.

The following functions are available:

Function (or shortcut)	Description
Open (or CTRL+O)	Opens the Maintenance Job Active to modify all definitions of the specified job: see Displaying and Modifying an Active Job Definition .
Display (or CTRL+D)	Opens the Display Job Active window to view all definitions of the specified job. The read-only fields in the window correspond to the fields in the Maintenance Job Active window: see Displaying and Modifying an Active Job Definition .
Network Diagram	Opens the Diagram Active Run window displaying an overview of the active job runs within the network. See Viewing a Network Diagram .
Zoom Subnetwork	See Listing Jobs of an Active Subnetwork .
Subnetwork Diagram	This function is used for navigation purposes. Using it for a job of type NET, the network diagram for the network that is linked to this object is opened. See Viewing a Network Diagram .

Function (or shortcut)	Description
Usable Symbol Tables	Opens a list of symbol tables used by the active job. See also Listing Usable Symbol Tables in the section <i>Symbol Table and Symbol Maintenance</i> .
Deactivate	See Deactivating a Job in an Active Network
Hold	See Holding an Active Job .
Release	See Releasing an Active Job .
Cancel	See Canceling an Active Job .
Resubmit	See Resubmitting an Active Job .
Reactivate	See Reactivating an Active Job .
Regenerate Active JCL	See Regenerating Active JCL .
Stop cyclic execution	Stops a currently executing cyclic job.
Modify Latest Start	See Modifying the Latest Start Time for an Active Job .
Browse Active JCL	See Browsing Active JCL .
Edit Active JCL	See Editing Active JCL .
Release edit lock	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 Locking of Natural Sources in the section <i>Job Maintenance</i> .
Extended Log / Active JCL Changes	See Viewing the Extended Log with Active JCL Changes . See also Choosing the Job ID of a Job for Logging .
Browse SYSOUT	See Viewing Job SYSOUT .
Extended Log / JCL	Opens a window where you can see the JCL which was submitted for this job. For further information, see JCL Log in the section <i>Job Maintenance</i> . See also Choosing the Job ID of a Job for Logging .
Extended Log / SYSOUT	Opens a window where you can see the SYSOUT of the job.
SYSOUT Messages	Opens a window where you can see the SYSOUT messages of the job (z/OS only).
Waiting for	See Displaying Prerequisites for Active Jobs: Waiting for .
List Active Jobs	See Listing Active Jobs .
Browse Log	See Displaying Logged Information - Browse Log Function .
Add to Workplan	See Add to Workplan .

Function (or shortcut)	Description
Set Drag And Drop Function	See <i>Drag & Drop</i> .

Listing Active Jobs

You can access active runs of a network to list and maintain active jobs from different nodes in the object workspace. The result list of active jobs depends on the node and the node instance selected.

User Restrictions for Active Jobs Lists

Users can be restricted to view jobs in the **List Active Jobs** window.

If the user is of type **Administrator** (A) 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*.

➤ To list active jobs

- 1 In the object workspace, select an instance from a **Network Active** or an **Active Run** node to list all active jobs of this particular network or active run, respectively.

Or:

In the object workspace, select an instance from an **Owner** or a **Network Master** node to list all active jobs of this particular owner or network, respectively.

Or:

In the object workspace, select the **General** node to list all active jobs running in your environment.

- 2 Open the context menu and choose the **List Active Jobs** function.

A **List Active Jobs** window similar to the example below opens with a list of active jobs:

Owner	Network	Run	Job	Node	JobId	Time	Message
NATQA5	A-OGCDEMO2	325	J-OGCDEMO	55522		07-04 12:20	Latest Start 07-04 12:20 exce
NATQA5	N1459C	9717	JOB1	55522		07-04 11:33	Dummy Job terminated
NATQA5	N1459C	9716	JOB1	55522		07-04 10:33	Dummy Job terminated
INCIDENT	I5033788BA	22548	JOB1	0		07-04 10:20	Node N0146 - Evaluation Erro
NATQA5	NET1787A	192	MA04	55522	55741	07-04 09:45	Ended not ok - P0001 G C001
NATQA5	NET1787A	192	MA03	55522	55740	07-04 09:45	Ended not ok - P0001 G C001
NATQA5	NET1787A	192	MA02	55522	55739	07-04 09:45	Ended not ok - P0001 G C001
NATQA5	NET1787A	192	MA01	55522	55738	07-04 09:45	Ended not ok - P0001 G C001
NATQA5	N1459C	9715	JOB1	55522		07-04 09:32	Dummy Job terminated
NATQA5	N1459C	9714	JOB1	55522		07-04 08:32	Dummy Job terminated
INCIDENT	I5033788BA	22547	JOB1	0		07-04 08:10	Node N0146 - Evaluation Erro
NATQA5	N1459C	9713	JOB1	55522		07-04 07:39	Dummy Job terminated
NATQA5	N1459C	9712	JOB1	55522		07-04 07:39	Dummy Job terminated
NATQA5	N1459C	9711	JOB1	55522		07-04 07:39	Dummy Job terminated
NATQA5	N1459C	9710	JOB1	55522		07-04 07:39	Dummy Job terminated
NATQA5	NET1582B	280	D1	55522		07-04 07:36	Start Time 07-05 04:00
NATQA5	NET1582	369	D1	55522		07-04 07:36	C2 - 369 - RUN not found
INCIDENT	I5033788BA	22546	JOB1	0		07-04 07:36	Node N0146 - Evaluation Erro
NATQA5	N1459A	10015	-	0		07-04 07:36	Latest Start 07-04 07:30 exce
NATQA5	N1459A	10014	-	0		07-04 07:36	Latest Start 07-04 05:30 exce
NATQA5	NET1582A	369	D1	55522		07-04 07:36	Schedule dependency: Calen
NATQA5	NET1582B	279	D1	55522		07-04 07:36	C1 - 279 - RUN not found

This list displays all active jobs available showing the owner, network, run, job node, job ID, time and a message.

The columns and input fields are explained in [Columns and Filter Criteria: List Active Jobs](#).



Notes:

1. For performance reasons, the table view is filled dynamically with data if scrolled to the end of the table.
2. For a comprehensive overview of the job list, the **Latest Start**, **Deadline** and **Version** columns and the **Filter** section are hidden.

➤ To display the Latest Start and Deadline columns

- Select the **Show schedule** check box.

The columns with the latest start times and deadlines are displayed as shown in the following example:

Owner	Network	Run	Job	Node	JobId	Latest Start	Deadline	Time	
NATQA5	A-OGCDEMO2	325	J-OGCDEMO	55522		07-04 12:20	07-04 13:00	07-04 12:20	Latest
NATQA5	N1459C	9717	JOB1	55522		07-05 05:29	07-05 05:30	07-04 11:33	Dumr

➤ To display the Version column

- Select the **Show version** check box.

The column with the job version is displayed as shown in the following example:

Owner	Network	Version	Run	Job	Node	JobId	Time	Mess
NATQA5	A-OGCDEMO2		325	J-OGCDEMO	55522		07-04 12:20	Latest Start 07-04
NATQA5	N1459C		9717	JOB1	55522		07-04 11:33	Dummy Job tern
NATQA5	N1459C		9716	JOB1	55522		07-04 10:33	Dummy Job tern
INCIDENT	I5033788BA		22548	JOB1	0		07-04 10:20	Node N0146 - E

➤ To change the sort order

- 1 Click on the column header area of any column in the table to sort by that column.

The list is then sorted by that column. The current sorting state is shown above the table.

The first time a user requests sorting, all data is retrieved from the server if it has not been done before. Therefore, some delay is possible, depending on the count of active jobs. Changing the preselection causes to re-read data again and initial sorting settings are applied.

- 2 Click on the column header again if you want to change the sort sequence from ascending (default) to descending.

The initial sort order can be changed on the tabbed page **Other Settings** of the **Create new User** or **Maintenance User** window (see *User Maintenance* in the *Administration* documentation).

➤ To filter the job list

- 1 Click on **Filter** ▶ **Filter** to expand (or collapse) the **Filter** section.

The **Filter** section is expanded as shown in the following example:

The screenshot shows the 'OGC - List Active Jobs' dialog box. The 'Filter' section is expanded, revealing several input fields and a 'Status filter' section. The 'Status filter' section contains several checkboxes, with 'All active jobs' and 'Waiting for activation' checked. Below the filter section, there are checkboxes for 'Show schedule' and 'Show version', and a sorting option 'Sorted ascending using attribute 'Owner/Network/Run/Job''. The main area of the dialog is a table with the following data:

Owner	Network	Version	Run	Job	Node	JobId	Latest Start	Deadline	
NATQA5	A-OGCDEMO2		325	J-OGCDEMO	55522		07-04 12:20	07-04 13:00	07
NATQA5	N1459C		9717	JOB1	55522		07-05 05:29	07-05 05:30	07
NATQA5	N1459C		9716	JOB1	55522		07-05 04:29	07-05 04:30	07
INCIDENT	I5033788BA		22548	JOB1	0		07-02 10:20	07-04 11:00	07
NATQA5	NET1787A		192	MA04	55522	55741	07-04 15:50	07-04 18:00	07
NATQA5	NET1787A		192	MA03	55522	55740	07-04 15:50	07-04 18:00	07
NATQA5	NET1787A		192	MA02	55522	55739	07-04 15:50	07-04 18:00	07
NATQA5	NET1787A		192	MA01	55522	55738	07-04 15:50	07-04 18:00	07
NATQA5	N1459C		9715	JOB1	55522		07-05 03:29	07-05 03:30	07
NATQA5	N1459C		9714	JOB1	55522		07-05 03:29	07-05 03:30	07

- 2 Specify or select the required filter criteria and choose **OK** when finished.

The columns and input fields are explained in [Columns and Filter Criteria: List Active Jobs](#).

➤ To refresh the list contents

- Proceed as described [Refreshing Object Lists](#).

This section covers the following topics:

- [Columns and Filter Criteria: List Active Jobs](#)

- [Listing Active Jobs by Process Status](#)

Columns and Filter Criteria: List Active Jobs

The columns and input fields available in the [List Active Jobs](#) window are explained in the following section.

Depending on the node or node instance for which the **List Active Jobs** function is used, the fields below [► Filter](#) are either read-only or can be used to specify filter criteria to reduce the number of active jobs listed in the window.

Column/Field	Description
Owner	The name of the owner whose active jobs are shown. If available as input field, you can select an owner out of the drop-down-box to show only its active jobs.
Network	The name of the network for which the active jobs are shown. If available as input field, you can select a network out of the drop-down-box to show only its active jobs.
Version	The network version for which the active jobs are shown. If available as input field, you can select a version out of the drop-down list box to show only its active jobs (see also <i>Object Versioning</i> in the <i>Concepts and Facilities</i> documentation).
Run	The run number for which the active jobs are shown. The run number is automatically assigned at activation time. The job run number uniquely identifies an active copy of a job together with the job name. If available as input field, you can select the run number of the network to show only its active jobs.
Job	The job name as defined to Entire Operations. If available as input field, you can select a job out of the drop-down-box to show only its active jobs.
Node	Execution node of the machine designated for the job. If available as input field, you can select a node out of the drop-down-box to show only its active jobs.
JobId	The job identifier as assigned by the operating system or by the job entry subsystem.
Time	Last action or check date/time for the job.
Message	Last message issued for the job by Entire Operations. 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.
Submit user ID	Select a user ID out of the drop-down-box to show only its active jobs.

Column/Field	Description
Show last <i>n</i> runs	Enter a number to restrict the list of active jobs to the last <i>n</i> runs where <i>n</i> is a number in the range 1 - 999999999. Valid valuesThe maximum value for <i>n</i> is 999999999.
Show schedule	See To display the Latest Start and Deadline columns.
Show version	See To display the Version column.
Status filter	Select job status filters to additionally reduce the list of active jobs: see Listing Active Jobs by Process Status.

Listing Active Jobs by Process Status

You can list active jobs according to their processing status by opening the **Filter** section of the **List Active Jobs Network Active** window and selecting one or more **Status filter** options:

Selection Option	Description
All active jobs	All active jobs (default).
Waiting for an event	All active jobs waiting for at least one event.
Jobs in hold	All jobs in hold.
In spool input queue	All jobs in the scheduling system input queue.
Ended not ok	All jobs with job not ok terminating status.
Ended ok	All jobs with job ok terminating status.
Submitted	All submitted jobs.
Terminated	All terminated jobs.
Waiting for activation	Planned network runs waiting for activation.
Executing	All jobs in execution.
Latest start time exceeded	All jobs that exceed the Latest Start Time : see Modifying the Latest Start Time for an Active Run.

Listing Jobs of an Active Subnetwork

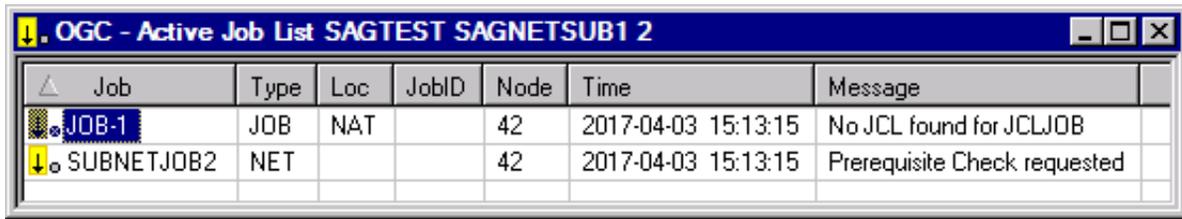
➤ To list jobs of an active subnetwork

- From a **Job Active** node, select a job of the type **Subnetwork** (NET), and choose **Zoom Subnetwork** from the context menu.

Or:

From the table in an **Active Job List** window, select a job of the **Type** NET, and choose **Zoom Subnetwork** from the context menu.

The jobs of the subnetwork defined for the selected NET-type job are listed in a (separate) **Active Job List** window, as shown in the following example:



Job	Type	Loc	JobID	Node	Time	Message
JOB-1	JOB	NAT		42	2017-04-03 15:13:15	No JCL found for JCLJOB
SUBNETJOB2	NET			42	2017-04-03 15:13:15	Prerequisite Check requested

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

- Select a **Job Active** node and choose **New** from the context menu.

A **Create new Job Active** window (similar to the **Maintenance Job Active**) opens where you can enter the required definitions as described for a master job in the section *Adding a Job Definition*.

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.

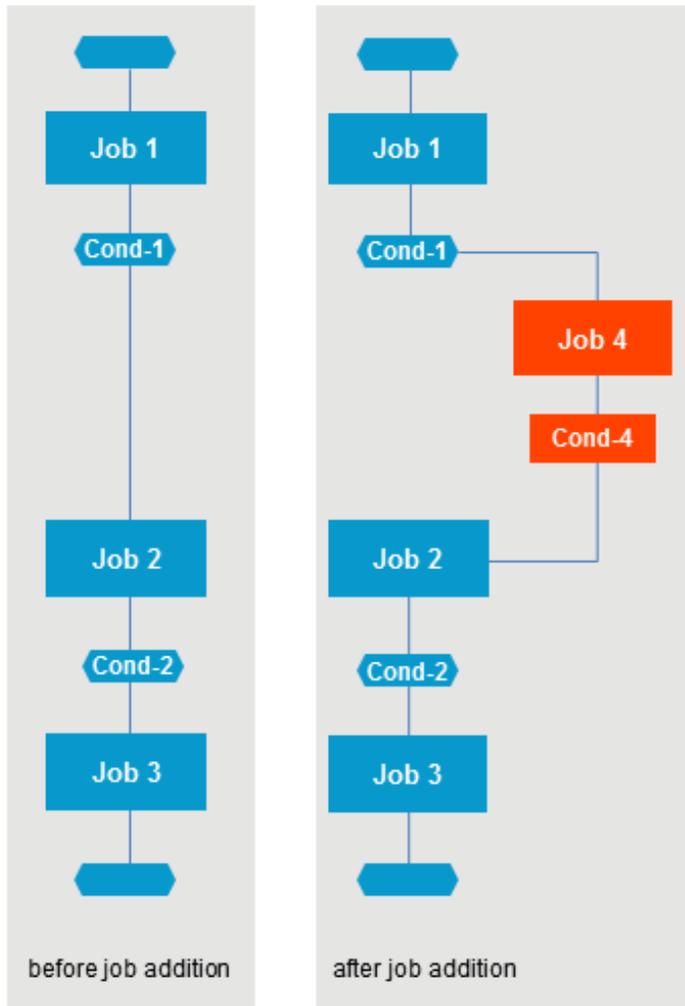
This section covers the following topics:

- [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 **Create new Job 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.

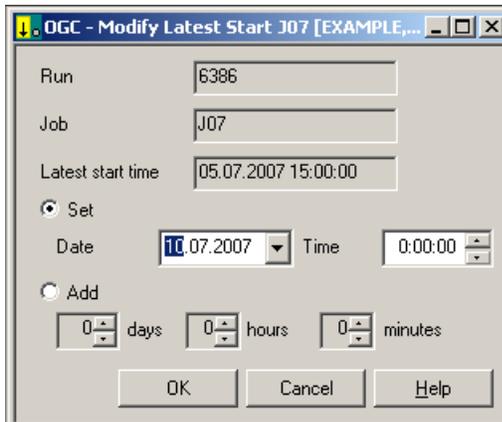
Modifying the Latest Start Time for an Active Job

You can change the latest start time or an active job instance where the active job status permits it. The latest start time may be changed for an active job that is not started or is already terminated.

➤ To modify the latest start time for an active job

- 1 In the object workspace, select an instance of **Job Active**.
- 2 Invoke the context menu and choose **Modify Latest Start**.

A window similar to the following opens:



The options available correspond to the methods described for an **active job run**.

Choosing the Job ID of a Job for Logging

If you have several instances of one job running, for example, after resubmitting a job, you receive a selection window for selecting the job ID of the job whose log you want to view.

➤ To choose the log

- 1 Select a job from the **Job Active** node and choose **Extended Log / Active JCL changes** or **Extended Log / JCL** from the context menu.

The following window opens:



JobID	Time
31869	13.04.2005 10:58:59
31522	13.04.2005 10:41:56

- 2 Select a job by double-clicking on a job ID in the **JobID** column.

The selected log is displayed:

```

OGC - Extended Log Job Active JOB-LONG [XSETAB0,TBOA3,240]
New TBOA3 Jb JOB-LONG Run 240 (11.04.2005 17:46:17) Act.JCL chng. -----
00001 /.AA LOGON
00002 /REMARK
00003 /REMARK THIS IS VERY LOOOOOONG JCL EXAMPLE
Del>---- /REMARK
00004 /REMARK THIS IS VERY LOOOOOONG JCL EXAMPLE
Del>---- /REMARK THIS IS VERY LOOOOOONG JCL EXAMPLE
00005 /REMARK
00006 /REMARK THIS IS VERY LOOOOOONG JCL EXAMPLE
Del>---- /REMARK THIS IS VERY LOOOOOONG JCL EXAMPLE
Del>---- /REMARK
Del>---- /REMARK THIS IS VERY LOOOOOONG JCL EXAMPLE
Del>---- /REMARK THIS IS VERY LOOOOOONG JCL EXAMPLE
00007 /REMARK
00008 /REMARK THIS IS VERY LOOOOOONG JCL EXAMPLE
00009 /REDDRK
Old>0009 /REMARK
00010 /REMARK DDDS IS VERY LOOOOOONG JCL EXAMPLE
Old>0010 /REMARK THIS IS VERY LOOOOOONG JCL EXAMPLE
00011 /REMARK
00012 /REMARK THIS IS VERY LOOOOOONG JCL EXAMPLE
    
```

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 a **Job Active** instance.

When Entire Operations submits jobs to the operating system, they are assigned a job number which appears, for example, in the **Maintenance Job Active** window. 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

- In the object workspace, select a **Job Active** instance and choose **Cancel** from the context menu.

A **confirmation dialog** prompts you to choose **Yes** to cancel the job.

The job is cancelled from the operating system.

Notes:

1. A cancelled job cannot be released for further processing.
2. For BS2000 jobs, you can only use the **Cancel** function 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

- In the object workspace, select a **Job Active** instance and choose **Hold** from the context menu.

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

- In the object workspace, select a **Job Active** instance and choose **Release** from the context menu.

The message `Job released from Hold` appears in the **Message** column for the job.



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.

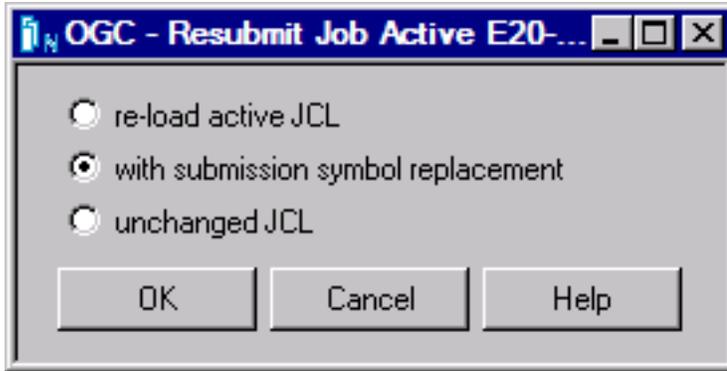
For example, if a JCL error has occurred, you can modify the active JCL (see [Editing Active JCL](#)) and resubmit the job. The resubmitted job uses the old input conditions for submission, but can set different output conditions, depending on the End-of-Job analysis.

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 In the object workspace, select a **Job Active** instance.
- 2 Invoke the context menu and choose the **Resubmit** function.

The following window opens:



Select a resubmission option and choose **OK**.

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** in the **Defaults for Other Settings** window described in the *Administration* documentation.

Resubmitting an Active Job of NET type (Subnetwork)

If the **Repeat** function 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`.

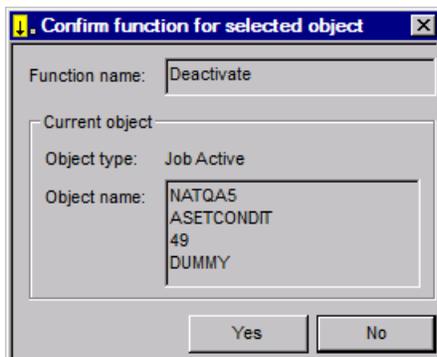
For each unsuccessful attempt, the message text will be set to `SYSOUT File Renaming - File in use`. In addition, the event will be logged with the SYSOUT file name.

Deactivating a Job in an Active Network

➤ To prevent the execution of a certain job in an active network for the current run

- 1 From a **Job Active** node in the object workspace, select an active job and choose **Deactivate** from the context menu.

A confirmation dialog similar to the example below opens for the selected job:



The window indicates the name of the owner (here: NATQA5), the name of the active network (here: ASETCONDIT), the run number (here: 49) and name of the active job (here: DUMMY).

- 2 Choose **Yes** to deactivate the job or **No** to cancel the action.

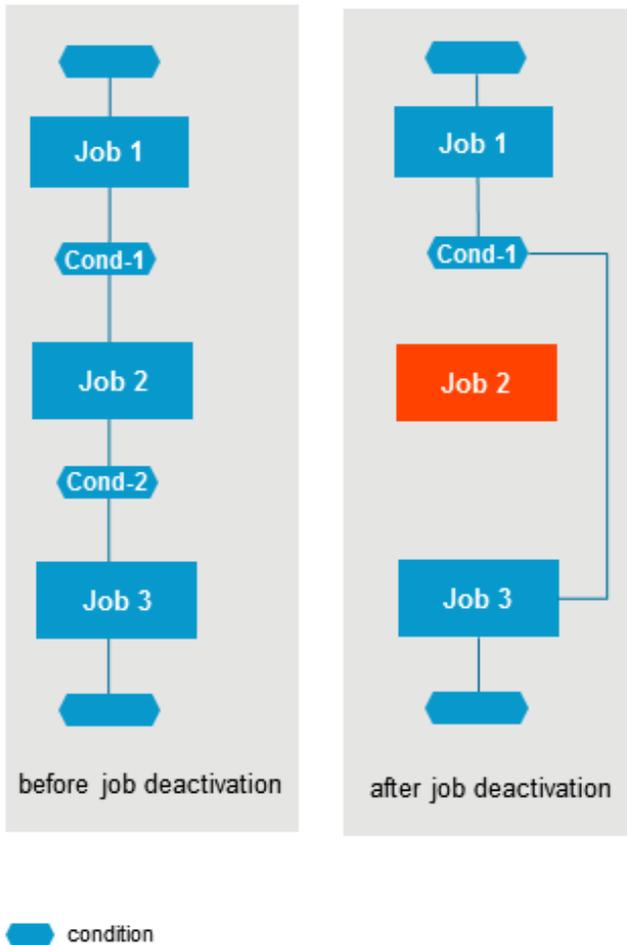
Deactivation is performed by the Entire Operations Monitor in the background. Deactivating a job from the active network involves deleting it from the active job network and restructuring the active network in order to ensure uninterrupted job flow. The following **example** illustrates job deactivation 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 with the **Deactivate** function, 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

- In the object workspace, select a **Job Active** instance and choose **Reactivate** from the context menu.

A **confirmation dialog** prompts you to choose **Yes** to reactivate the job.

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.

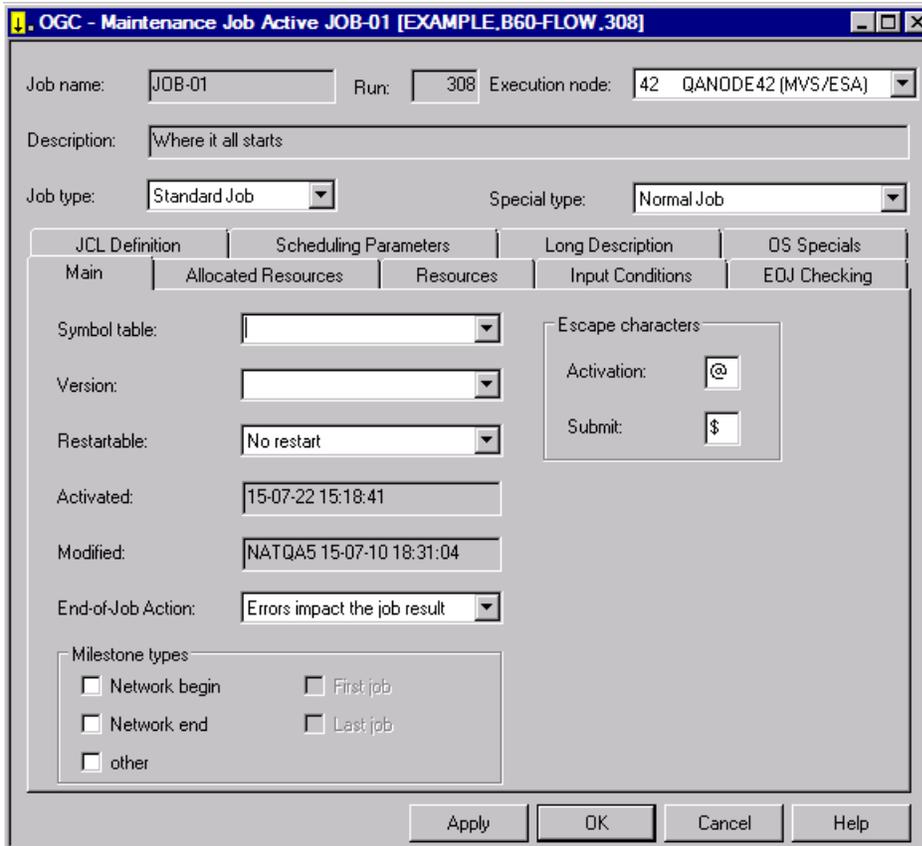
- All input and output definitions are recreated.
- The active JCL is generated newly.

Displaying and Modifying an Active Job Definition

➤ To view and modify an active job

- 1 In the object workspace, select a **Job Active** instance.
- 2 Invoke the context menu and choose **Open** or **Display**.

Open opens a **Maintenance Job Active** window similar to the example below:



Display opens a **Display Job Active** window which corresponds to the **Maintenance Job Active** window above but is write-protected.

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 **Resubmit** function. This causes a new check of all prerequisites.

This section covers the following topics:

- [Fields: Maintenance Job Active](#)

- [Special Job Type D - Execution as a Dummy](#)

Fields: Maintenance Job Active

The input fields in the [Maintenance Job 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 [Maintenance Job 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 [Maintenance Job Active](#) window, it is possible to modify the field **Special Type** to **Execute as dummy**.

- If **Execute as dummy** is not set, an automatic reloading of the JCL is performed. In case of active jobs of the type [NET](#), the subnetwork is activated.
- If **Execute as dummy** 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](#).

Modifying EOJ Checking and Actions

➤ To modify the End-of-Job checking and actions for the current run of a specific job

- In the [Maintenance Job Active](#) window, open the **EOJ Checking** page.

The current definitions are displayed on the page similar to the [example](#) shown in the section [End-of-Job Checking and Actions](#). 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**

- In the **Maintenance Job Active** window, open the **Long Description** page.

An **editor screen** similar to the example shown in the section *Job Maintenance* is shown.

The page contains the long description of the active job, provided a long description exist for the job master. The description is displayed in Editor format. 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*.

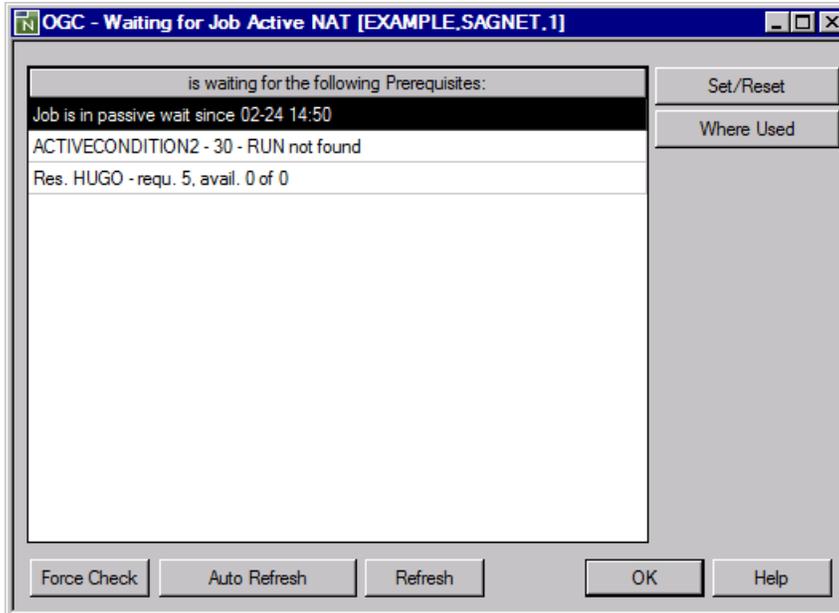
Displaying Prerequisites for Active Jobs: Waiting for

The **Waiting for** function is used to display all events pending for an active job. You can also use this function to add additional conditions to a job.

➤ **To display all waiting prerequisites for an active Job**

- 1 In the object workspace, select a **Job Active**.
- 2 Invoke the context menu and choose the **Waiting for** function.

A **Waiting for Job Active** window similar to the example below opens:



The window lists the events an active job is waiting to complete.

The listed messages indicate that the current run of the job (run number 1) is waiting for the input condition ACTIVECONDITION2 from the run number 30, and the availability of the resource HUGO. A resource is indicated by a Res. prefix.

The text Job is in passive wait since 02-24 14:50 shows since when the job has been in passive wait for the input condition and the resources.

In this case, the job is automatically returned to the active wait by setting the correct input condition and making sure that the resource can be allocated; that is, the Monitor will execute a prerequisite check.

For a complete list of possible messages and their meaning, see *Messages in Active Jobs Display* in the *Messages and Codes* documentation.

All functions available in the **Waiting for** window are explained in [Functions: Prerequisites for an Active Job](#).

Functions: Prerequisites for an Active Job

The following functions are available in the **Waiting for Job Active** window:

Function	Description
Force Check	Force an active prerequisite check. Only before the message <code>Job is in passive wait since ...</code>
Auto Refresh	See Refreshing Object Lists .
Refresh	
Set/Reset	<p>Set:</p> <ul style="list-style-type: none"> ■ Used for a condition wait: set the condition to true. <p>Reset:</p> <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.
Where Used	<p>Where used:</p> <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. ■ If used for a resource wait: display the usage of the resource. For further information, see Viewing and Modifying Resources Used by Active Jobs.

Viewing and Modifying Resources Used by Active Jobs

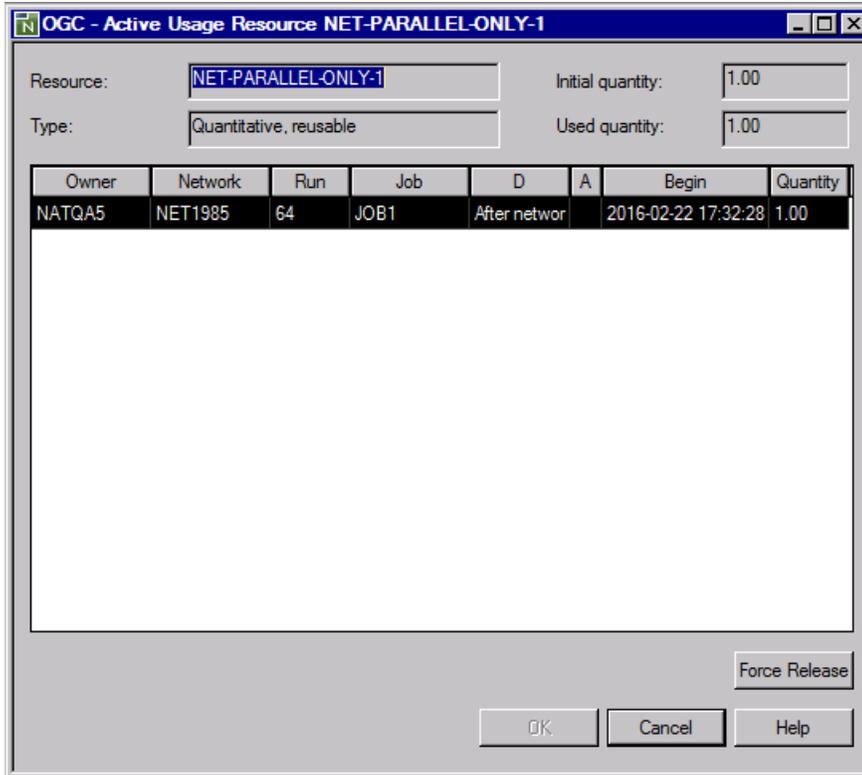
You can display the resources used by active jobs.

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 view resources used by an active job

- In the **Waiting for Job Active** window, select the required resource and choose **Where Used**.

An **Active Usage Resource** window similar to the example below opens:



The window lists other active jobs in your active network environment which are currently using partial quantities of the resource for which your active job is waiting.

The fields and columns in the window are described in *Fields and Columns: Active Resource Usage* in the *Administration* documentation.

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](#)
- [Available Functions: Browse SYSOUT](#)

Viewing Job SYSOUT

> To view job SYSOUT

- 1 In the object workspace, select a **Job Active** instance and choose **Browse SYSOUT** from the context menu.

Depending on your access rights, a **Logon Node window** opens.

- 2 Enter your user ID and password and choose **OK**.

A **Browse** window similar to the example below opens:

On UNIX:

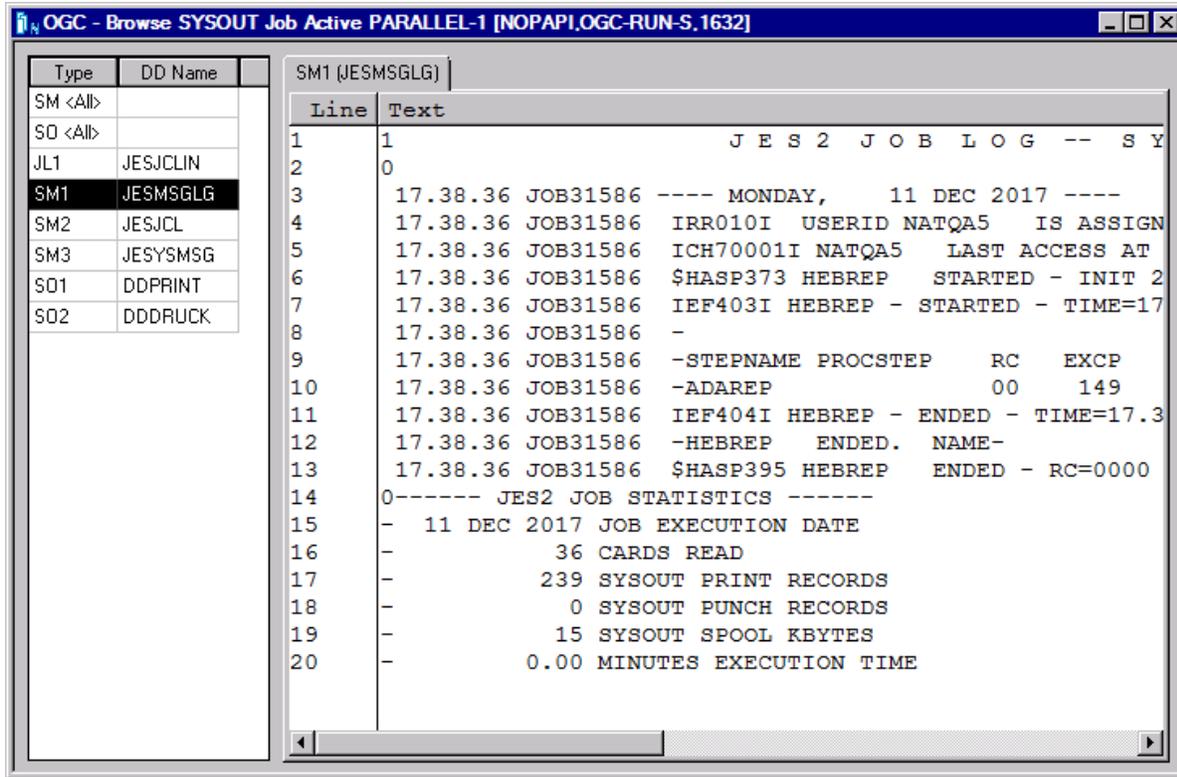
Line	Text
1	%% EOR0301 - PID 32532 Start 20171214-114346 +0100
2	Linux susnat2 2.6.32.12-0.7-default #1 SMP 2010-05-20 11:
3	uid=4051(mut) gid=1023(natural) groups=1023(natural)
4	+ /NAT/nathome/natqa5/NOP531/nop_jcl/first.bsh
5	/FS/fs0404/natprod/nop/543/work/natqa5/0010300204/NOPAPI/
6	+ set +x
7	%% EOR0306 - User Script Exitcode 0
8	%% EOR0302 - PID 32532 End 20171214-114346 +0100 Exitcode
9	%% EOR0303 - times - Begin
10	0m0.004s 0m0.000s
11	0m0.000s 0m0.000s
12	%% EOR0304 - times - End
13	PID TTY TIME CMD
14	32532 ? 00:00:00 PARALLEL-1.curr
15	+ exit 0

On BS2000:

Line	Text
1	/REMARK EOR4127 - SYSOUT File Assignment by NOP
2	/GETJV \$NATRRI.NOPAPI.OGC-RUN-S.1652.PARALLEL-1S
3	\$NATRRI.NOPAPI.OGC-RUN-S.1652.PARALLEL-1
4	
5	/ERAJV \$NATRRI.NOPAPI.OGC-RUN-S.1652.PARALLEL-1S
6	/REMARK =====
7	/REMARK S O F T W A R E A G
8	/REMARK Entire Operations Version 5.4.3.10
9	/REMARK
10	/REMARK Owner NOPAPI Run 1652
11	/REMARK Network OGC-RUN-S Symbol Table OGC-RUN-S
12	/REMARK Version Version
13	/REMARK Job PARALLEL-1 Escape Act. @ Sub.
14	/REMARK Repetition 0
15	/REMARK Submit User ID NATRRI
16	/REMARK JCL Node 125 Exec.Node 125
17	/REMARK NPR Version 3.6.2
18	/REMARK MonJV \$NATRRI.NOPAPI.OGC-RUN-S.1652.PAR
19	/REMARK Monitor Codepage EDF03DRV
20	/REMARK
21	/REMARK 17-09-25 14:21 created/modified .. NATQA
22	/REMARK 17-12-12 02:16 activated NATQA - Orig
23	/REMARK 17-12-12 02:16 last action NATQA
24	/REMARK 17-12-12 02:16 submitted

The window displays active SYSOUT produced for the DD statements processed by the selected job (here: PARALLEL-1, owner NOPAPI, network OGC-RUN-S, run number 1628 (UNIX) or 1652 (BS2000)). SYSOUT output data is contained in numbered text lines sorted in ascending order.

On z/OS and z/VSE:



If more than one SYSOUT file exists, a table in the left window section lists all DD statements produced by the job and the **type of the SYSOUT file**. For each file you select in this table, a tabbed page opens in the right window section showing the SYSOUT of the corresponding DD statement. Several files of the same file type are numbered sequentially from one (SM1, SM2, and so on). Files of the same type are combined in an <All> collection file, for example, SM <All>.

The browse functions available are explained in [Available Functions: Browse SYSOUT](#).

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 *Defaults for Network Options* 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.
<A11>	Identifies a collection file that combines all files that belong to a certain file type, for example, SM <A11>.	

Available Functions: Browse SYSOUT

The following functions are available from the context menu of the **Browse SYSOUT** window:

Function	Shortcut	Description
Copy	CTRL+C	Copy the text lines selected in the SYSOUT file to the clipboard to be pasted into another file.
Reverse Order	CTRL+R	Display SYSOUT text in descending order of line numbers (last line first). The text is sorted in ascending order by default.
Refresh	F5	See Refreshing Object Lists .

Function	Shortcut	Description
Find	CTRL+F	Open the Find dialog to enter a search string and specify search options (case sensitive and/or whole word only) and the search direction backward (scrolls up in the source) or forward (scrolls down in the source).
Find Next	F3	Find the next occurrence of the search string specified in the Find dialog.
Go To	CTRL+G	Scan the SYSOUT file for a specified line number and highlight this line.
Print	CTRL+P	Output the SYSOUT file on a printer.

49

Maintaining Active Job Conditions

- Available Functions: Condition Active 478
- Listing Active Conditions 479
- Viewing an Active Condition and Changing Its Status 481
- Adding an Active Condition 481
- Using Global Active Conditions 482
- Viewing the Usage of Conditions Concatenated with an Active Job 485
- Viewing the Last Job Used by a Condition Active 486

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*.

Available Functions: Condition Active

➤ **To list all available functions for a Condition Active**

- In the object workspace, select a **Condition Active** node and open the context menu.

The following functions are available:

Function	Shortcut	Description
List	F8	See <i>Listing Active Conditions</i> .
New	CTRL+N	See <i>Adding an Active Condition</i> .
Refresh	F5	See <i>Refreshing Object Lists</i> .
Filter	F3	Use filter criteria to list conditions active: see <i>Filtering Objects</i>
Export	---	Opens the Export Objects window to export all items of the metanode Condition Active : see <i>Exporting Objects</i> in the <i>Import/Export Functions</i> documentation.

➤ **To list all available functions for a Condition Active instance**

- Select an active condition from a **Condition Active** node and open the context menu.

The following functions are available:

Function	Shortcut	Description
Open	CTRL+O	See Viewing an Active Condition and Changing Its Status .
Display	CTRL+D	Display condition active information: see Viewing an Active Condition and Changing Its Status .
Active Usage	---	See Viewing the Last Job Used by a Condition Active .
Delete	DELETE	Delete a condition active: see Deleting Objects - Delete Function .
Add to Workplan	---	See Add to Workplan .

Listing Active Conditions

➤ To list active job conditions

- In the object workspace, select a **Condition Active** node from a **Network Active** node or an **Active Run** instance and choose **List** from the context menu, or press F8.

A **Condition List** window similar to the example below opens:

Condition	Time	Run	Status	Special
DUMMY-OK	2016-02-17 12:18:08	52	free	
NET-BEGIN	2016-02-17 12:17:57	52	free	
NET-END	2016-02-17 12:18:13	52	free	
SETCON-OK	2016-02-17 12:18:13	52	free	

Depending on the **Condition Active** node selected, the window lists all active conditions of the selected network active or network active run, respectively.

The example above lists all active conditions of the active job run 52 of the network ASETCONDIT which belongs to the owner NATQA5.

The columns contained in the window are explained in [Columns and Fields: Active Conditions](#).

➤ To list global active conditions

- See [Listing Global Active Conditions](#).

This section covers the following topics:

■ Columns and Fields: Active Conditions

Columns and Fields: Active Conditions

The following table explains the columns of the **Condition List** window. The names of the columns correspond to the names of the fields available when adding or viewing an active condition in the **Condition Active or Global Active Condition** window, unless indicated otherwise in the **Field** column of the following table.

Column	Field	Description		
Condition		<p>Active condition name as defined on the master database.</p> <p>Note: In the object workspace, an active condition that is being used for JCL checking only has the suffix <i>C</i> or <i>run-number C</i> (for example, TESTCONDITION 2 C).</p> <p>When adding or modifying a condition:</p> <ul style="list-style-type: none"> ■ Pay attention to the restrictions for condition names described in the section <i>Job Maintenance</i>. ■ For global active conditions, use the functions provided for the Global Active Condition metanode. 		
Time	Date Time	<p>Date and time when this condition was set. The date and time serve as search criteria for Entire Operations if no run number is specified.</p> <p>See also <i>Date and Time Formats</i>.</p>		
Run	Run number	Run number of the job which sets this condition, or the value <i>abs.</i> or <i>void</i> .		
Status		Status of the condition. Possible selection options:		
	free	Can be used by any job.		
	in use	Can be used by jobs which do not require exclusive usage.		
	exclusive	Currently in use by a job; not usable by other jobs.		
	destructive	Currently in use by a job; not usable by other jobs. The condition will be deleted after job termination.		
Special	Special type	Usage of the condition.		
		Possible selection options for input fields:		
		<table border="1"> <tr> <td>Normal usage</td> <td>Condition is not used for checking JCL.</td> </tr> <tr> <td>Use only for JCL check</td> <td> Condition is only used for checking JCL. If this option is selected, the letter C is shown in this column. </td> </tr> </table>	Normal usage	Condition is not used for checking JCL.
Normal usage	Condition is not used for checking JCL.			
Use only for JCL check	Condition is only used for checking JCL . If this option is selected, the letter C is shown in this column.			

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

- In the object workspace, select a **Condition Active** instance and choose **Display** (CTRL+D) or **Open** (CTRL+O) from the context menu.

If you choose **Open**, you can change the **status** of the condition, if required.

A **Condition Active** window opens.

The fields in the window have the same meaning as the corresponding table columns in the **Condition List** window. They are explained in *Columns and Fields: Active Conditions*.

Adding an Active Condition

You can add an active condition to change the job flow by defining if for a job before submission.

➤ To create a new condition active

- 1 In the object workspace, select a **Condition Active** node.
- 2 Open the context menu and choose **New**, or press CTRL+N.

A **Condition Active** window similar to the example below opens:

The input fields have the same meaning as the corresponding columns in the **Active Conditions** window. They are explained in *Columns and Fields: Active Conditions*.

- 3 Make your definitions and choose **OK** when finished.

The condition is created.

Using Global Active Conditions

This section covers the following topics

- [Available Functions: Global Active Conditions](#)
- [Listing Global Active Conditions](#)
- [Viewing a Global Active Condition and Modifying Its Status](#)
- [Adding a Global Active Condition](#)

Available Functions: Global Active Conditions

➤ **To list all available functions for the Global Active Condition metanode**

- In the object workspace, select the **Global Active Condition** metanode and open the context menu.

The following functions are available:

Function	Shortcut	Description
List	F8	See <i>Listing Global Active Conditions</i> .
New	CTRL+N	See <i>Adding a Global Active Condition</i> .
Refresh	F5	See <i>Refreshing Object Lists</i> .
Filter	F3	Specifies selection criteria for listing global active conditions. See <i>Filtering Objects</i> .
Export	---	Exports global active conditions. See <i>Exporting Objects</i> in the <i>Import/Export Functions</i> documentation.
Set Drag And Drop Function	---	See <i>Drag & Drop</i> .

➤ To list all available functions for a global active condition

- In the object workspace, select a **Global Active Condition** instance and open the context menu.

The following functions are available:

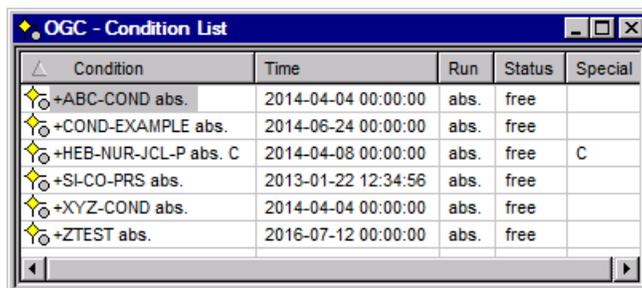
Function	Shortcut	Description
Open	CTRL+O	View a global active condition and change its status. See <i>Viewing a Global Active Condition and Modifying Its Status</i> .
Display	CTRL+D	View a global active condition (no status change possible). See <i>Viewing a Global Active Condition and Modifying Its Status</i> .
Active Usage	---	See <i>Viewing the Last Job Used by a Condition Active</i> .
Delete	DELETE	Delete a global active condition. See also <i>Deleting Objects</i> .
Add to Workplan	---	See <i>Add to Workplan</i> .
Set Drag And Drop Function	---	See <i>Drag & Drop</i> .

Listing Global Active Conditions

➤ To list global active job conditions

- In the object workspace, select the **Global Active Condition** metanode and choose **List** from the context menu, or press F8.

A **Condition List** window similar to the example below opens:



This window lists all global active conditions defined in your environment.

The fields contained in the window correspond to the fields in the **Condition List** window for a single owner. They are explained in *Columns and Fields: Active Conditions*.

Viewing a Global Active Condition and Modifying Its Status

➤ To view a global active condition and change its status

- 1 In the object workspace, select a **Global Active Condition** instance
- 2 From the context menu, choose **Display** (CTRL+D) or **Open** (CTRL+O) .

If you choose **Open**, you can change the **status** of the condition, if required.

A **Global Active Condition** window opens.

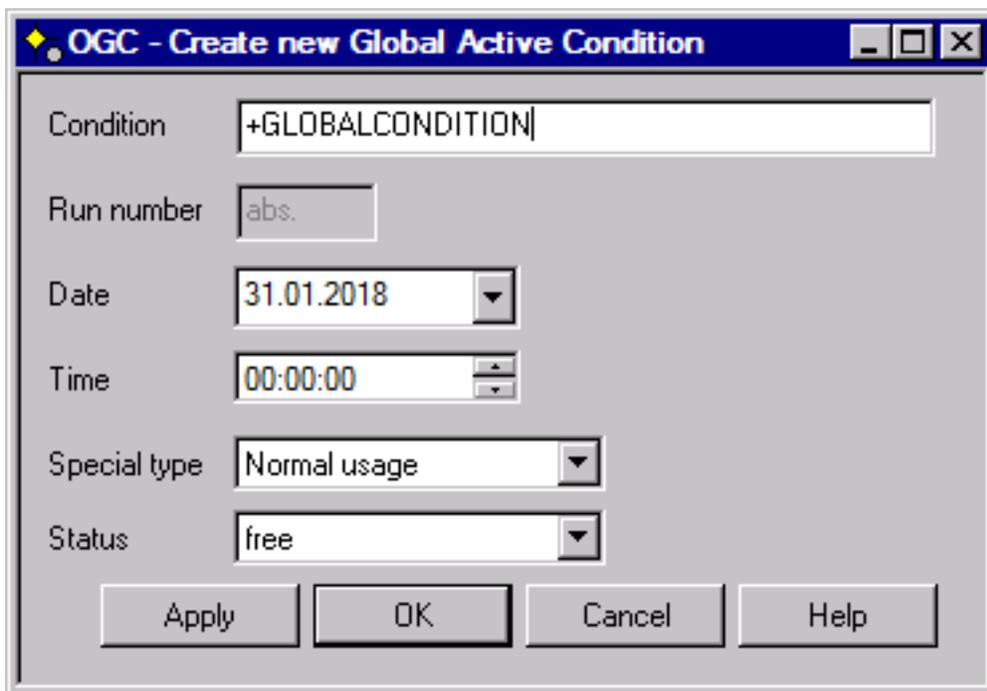
The fields in the window have the same meaning as the corresponding table columns in the **Condition List** window. They are explained in *Columns and Fields: Active Conditions*.

Adding a Global Active Condition

➤ To add a global condition

- 1 In the object workspace, select the **Global Active Condition** metanode and choose **New** from the context menu, or press CTRL+N.

A **Global Active Condition** window similar to the example below opens:



The fields contained in the window correspond to the fields in a **Condition Active** window. They are explained in *Columns and Fields: Active Conditions*.

For the rules and restrictions that apply to global active conditions, see *Global Conditions* in the section *Job Maintenance*.

- 2 Change or add the required values and choose **OK** when you are finished.

The global condition is created and can be used in further networks.

Viewing the Usage of Conditions Concatenated with an Active Job

➤ To view condition usage of an active job

- From the list of items on the **Input Conditions** page of the **Maintenance Job Active** window, select the required condition and choose **Where Used**.

An **Active Usage Condition** window similar to following example opens:

Owner	Network	Run	Job	Owner	Network	Run	Job
EXAMPLE	E60-FLOW	0000000061	JOB-01	EXAMPLE	E60-FLOW	0000000061	JOB-012
EXAMPLE	E60-FLOW	0000000061	JOB-02	EXAMPLE	E60-FLOW	0000000061	JOB-019

The columns contained in the window correspond to the columns in the **Where used Condition** window described in *Input Condition Maintenance*.

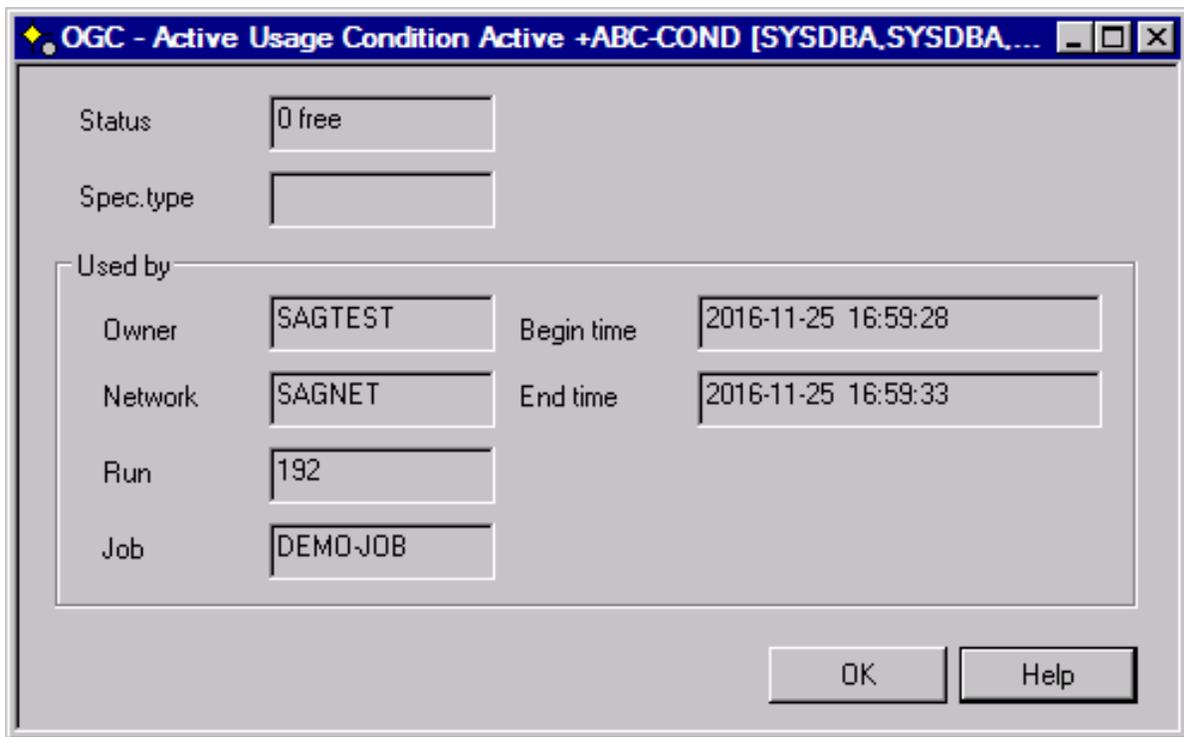
Viewing the Last Job Used by a Condition Active

This function is used to display the job that last used an active condition.

➤ **To view the job that last used an active condition**

- 1 In the object workspace, select a **Condition Active** instance.
- 2 Open the context menu and choose **Active Usage**.

An **Active Usage** window similar to the example below opens:



The window displays the active job that last used the active condition. The fields in the window are explained in [Columns and Fields: Active Conditions](#).

50

Maintaining Active JCL (Job Control Language)

- Browsing Active JCL 488
- Editing Active JCL 489
- Viewing the Extended Log with Active JCL Changes 490
- Exchanging Active JCL 491
- Regenerating Active JCL 491

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](#)

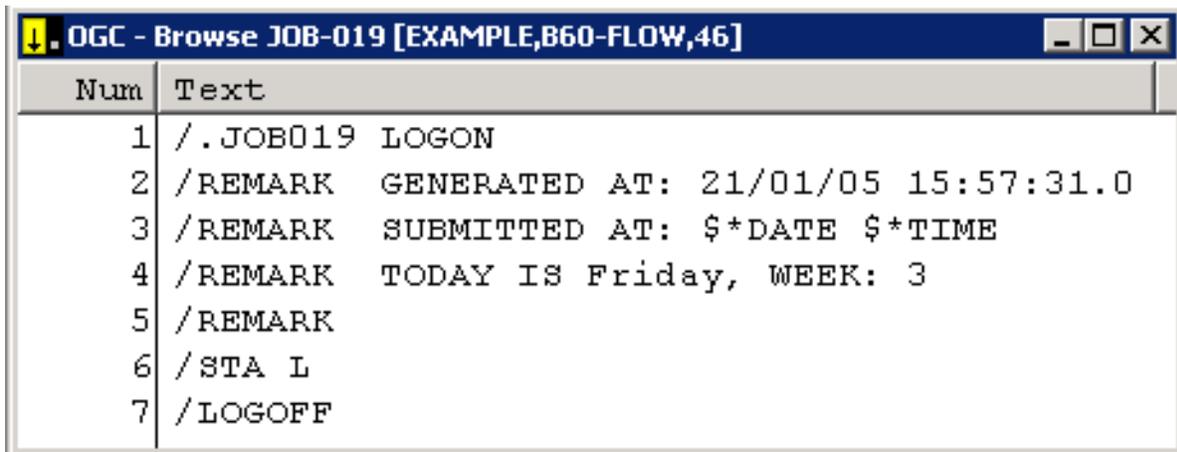
Browsing Active JCL

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 In the object workspace, select a **Job Active** instance.
- 2 Invoke the context menu and choose the **Browse Active JCL** function.

The following window appears:



- 3 Here, you can see 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.

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.

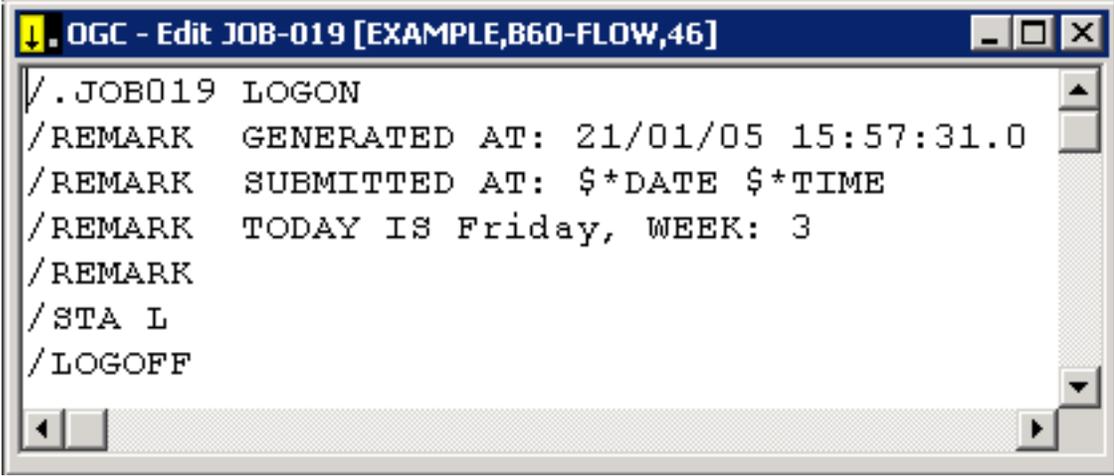
The following applies when editing active JCL:

- 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 **Job Active** node 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, use the **Resubmit** function from the context menu (see [Resubmitting an Active Job](#)).

➤ To edit active JCL

- 1 From a **Job Active** node in the object workspace, select an active job and choose **Edit Active JCL** from the context menu.

An editor window similar to the example below opens:



```

OGC - Edit JOB-019 [EXAMPLE,B60-FLOW,46]
/ .JOB019 LOGON
/REMARK   GENERATED AT: 21/01/05 15:57:31.0
/REMARK   SUBMITTED AT: $*DATE $*TIME
/REMARK   TODAY IS Friday, WEEK: 3
/REMARK
/STA L
/LOGOFF
  
```

The window contains the source of an active JCL.

- 2 Modify the source as required and confirm the changes when closing the window. See also *Editing Master JCL and Natural Sources* in the section *Job Maintenance*.

Viewing the Extended Log with Active JCL Changes

➤ To view the Extended Log / Active JCL Changes

- 1 In the object workspace, select a **Job Active** instance.
- 2 Invoke the context menu and choose the **Extended Log / Active JCL Changes** function.

The following window appears:

```

OGC - Extended Log Job Active JOB-LONG [XSETABO,TBOA3,240]
New TBOA3 Jb JOB-LONG Run 240 (11.04.2005 17:46:17) Act.JCL chng. -----
00001 /.AA LOGON
00002 /REMARK
00003 /REMARK THIS IS VERY LOOOOOONG JCL EXAMPLE
Del>---- /REMARK
00004 /REMARK THIS IS VERY LOOOOOONG JCL EXAMPLE
Del>---- /REMARK THIS IS VERY LOOOOOONG JCL EXAMPLE
00005 /REMARK
00006 /REMARK THIS IS VERY LOOOOOONG JCL EXAMPLE
Del>---- /REMARK THIS IS VERY LOOOOOONG JCL EXAMPLE
Del>---- /REMARK
Del>---- /REMARK THIS IS VERY LOOOOOONG JCL EXAMPLE
Del>---- /REMARK THIS IS VERY LOOOOOONG JCL EXAMPLE
00007 /REMARK
00008 /REMARK THIS IS VERY LOOOOOONG JCL EXAMPLE
00009 /REDDRK
Old>0009 /REMARK
00010 /REMARK DDDS IS VERY LOOOOOONG JCL EXAMPLE
Old>0010 /REMARK THIS IS VERY LOOOOOONG JCL EXAMPLE
00011 /REMARK
00012 /REMARK THIS IS VERY LOOOOOONG JCL EXAMPLE
    
```

All **Extended Log / Active JCL Changes** which were made since the last editing process are listed.

For further information, see *Editing Master JCL and Natural Sources* in the section *Job Maintenance*.

See also *Choosing the Job ID of a Job for Logging*.

 **Note:**

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 **Resubmit** function. 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 *Defaults for Other Settings* 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 In the object workspace, select a Job Active instance.
- 2 Invoke the context menu and choose the **Regenerate active JCL** function.

The job is regenerated and ready to execute.

Resubmitting the Active Job after JCL Regeneration

After regeneration of the JCL with the **Re-generate Active JCL** function, 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

- Select the job with the regenerated JCL and choose **Resubmit** from the context menu.

See also [Resubmitting an Active Job of NET type \(Subnetwork\)](#).

X 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*

51

General Scheduling Considerations

▪ Optional Schedule Definition	496
▪ Schedule Extraction Times	496
▪ Manual and Automatic Activations on the same Day	496
▪ Multiple Network Activations	496
▪ Influence of Deactivations on Schedules	496
▪ Imported Schedules	497
▪ Schedule Dependencies across the Turn of the Year	497
▪ Using Calendars	497

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 functions, 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.

52

Maintaining Schedules

- Listing Schedules 500
- Displaying, Adding or Modifying a Schedule 501
- Deleting a Schedule Definition 512

Listing Schedules

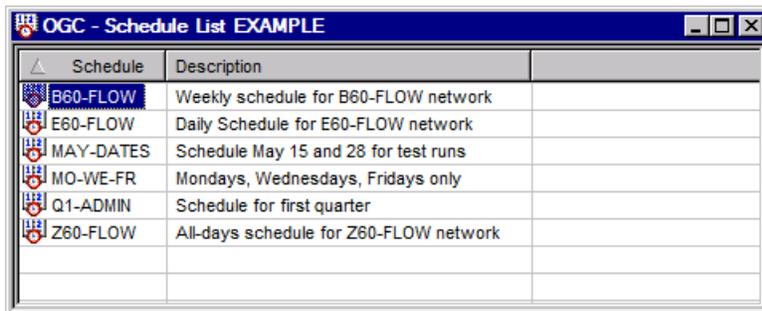
➤ To list all schedule definitions of an owner

- 1 In the object workspace, select the **Schedule** node of an owner.
- 2 From the context menu, choose **List**.

Or:

Press F8.

All schedules defined for the selected owner (here: EXAMPLE) are listed in the content pane of the **Schedule List** window as shown in the following example:



If one or more job schedules are defined for the owner, they appear in this window.

This section covers the following topics:

- [Fields: Schedule List Window](#)

Fields: Schedule List Window

The following fields are available in the [Schedule List window](#):

Fields	Description
Schedule	User-defined schedule name.
Description	Short description of the schedule.

Displaying, Adding or Modifying a Schedule

This section covers the following topics:

- Available Functions: Schedule Maintenance
- Viewing a Schedule Definition
- Viewing a Schedule Definition as a Calendar
- Adding a Schedule Definition
- Modifying a Schedule Definition
- Fields: Schedule Definition
- Available Functions: Schedule Definition
- Using Day Lists

Available Functions: Schedule Maintenance

➤ To list all functions available for a schedule node

- In the object workspace, select the **Schedule** node from an **Owner** instance and open the context menu.

The following functions are available for a selected **Schedule** node:

Function	Description
List	List all schedules for an owner in the Schedule List window .
New	Add a schedule definition: see Adding a Schedule Definition .
Refresh	Refresh the list of schedules: see Refreshing Object Lists .
Filter	Use selection criteria to list schedules: see Filtering Objects .
Paste data	Paste a schedule: see Pasting Objects .
Export	Open the Export Objects window to export all items of the metanode Schedule : see Exporting Objects in the <i>Import/Export Functions</i> documentation.
Set Drag and Drop Function	Drag nodes and drop them in the empty working area: see Drag & Drop .

➤ To list all functions available for a schedule

- From the **Schedule** node, select an instance and open the context menu.

The following functions are available for a selected schedule:

Function	Description
Open	Modify the schedule: see Modifying a Schedule Definition .
Display	Display the schedule definition: see Viewing a Schedule Definition .
Display as a Calendar	Display the calendar view of the schedule definition. See Viewing a Schedule Definition as a Calendar .
Where used	List using networks.
Delete	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, choose the Where used function.
Copy data	Copy a schedule definition: see Copying Objects .
Export	Open the Export Objects window to export the schedule: see Exporting Objects in the <i>Import/Export Functions</i> documentation.
Add to Workplan	Store a reference to the schedule in a list of activities to be done: see Add to Workplan .
Set Drag And Drop Function	See Drag & Drop .

Viewing a Schedule Definition

➤ To view a schedule definition

- In the object workspace, select a **Schedule** instance and choose **Display** from the context menu, or press Ctrl+D.

A **Display Schedule** window opens showing the current date settings of the selected schedule.

The protected fields and tabbed pages in the window correspond to the input fields and tabbed pages of the **Maintenance Schedule window**. They are explained in [Fields: Schedule Definition](#).

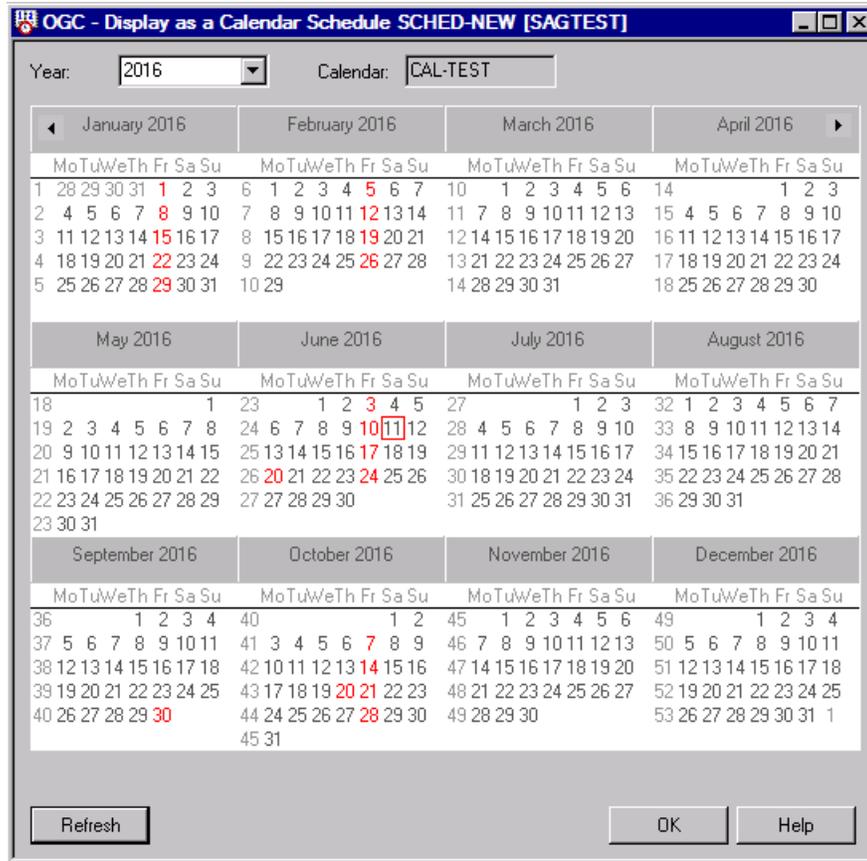
Viewing a Schedule Definition as a Calendar

If a calendar is assigned with a schedule, you can view the schedule definition as a calendar.

➤ To display the calendar view of a schedule

- 1 In the object workspace, select a **Schedule** instance and choose **Display as a Calendar** from the context menu.

A **Display as a Calendar** window similar to the example below opens:



The current six months defined for the schedule are shown by default. You can view additional months or all month of the year by drawing the window borders to resize the window as required.

The current date is indicated by a red square, defined dates are highlighted in red.

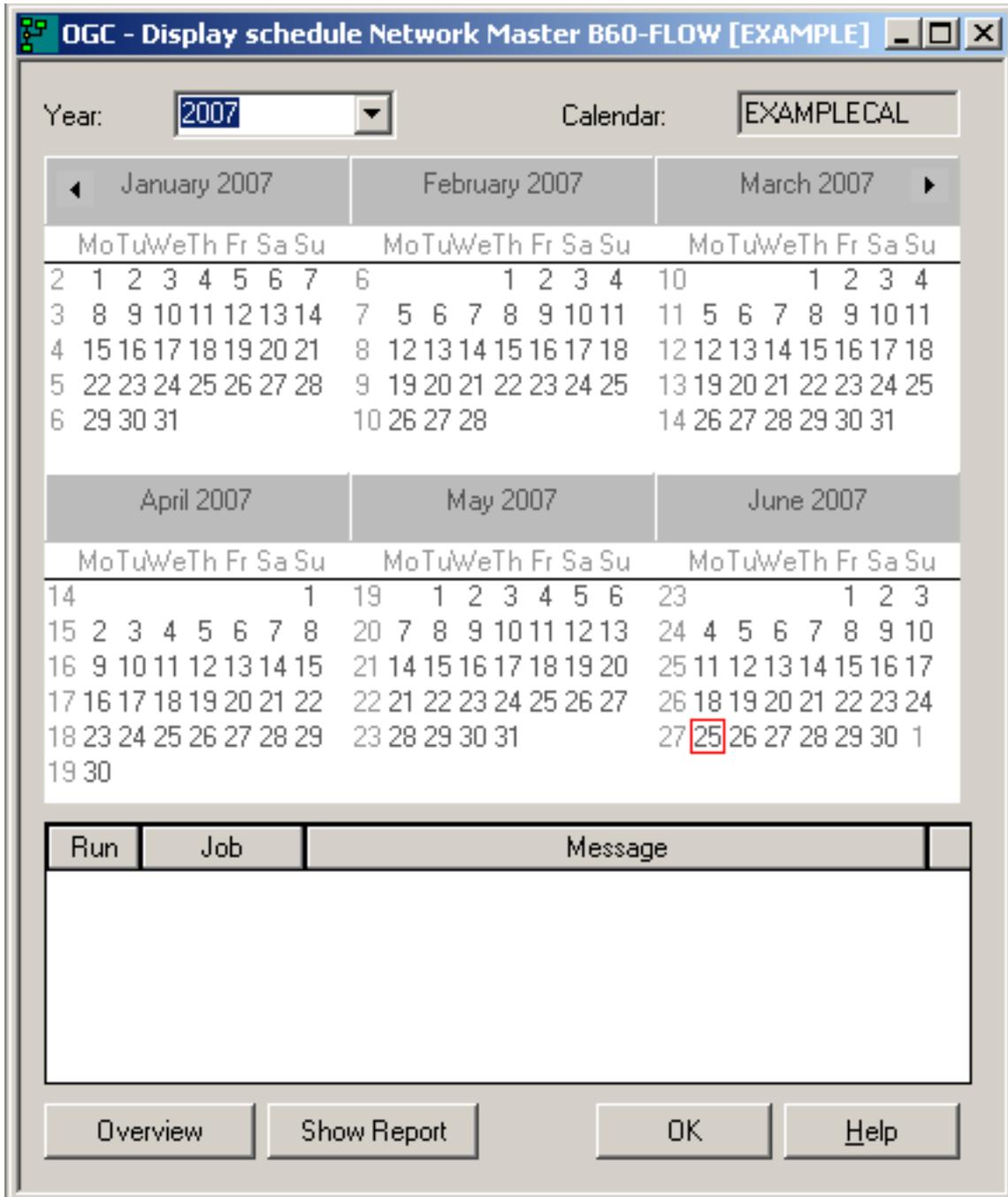
You can choose **Refresh** to update the calendar view.

- 2 If you select a year in the **Year** box that is not defined in the referenced calendar, an EOR3014 error can occur with a corresponding message.

» **To display the calendar view of a network schedule**

- 1 In the object workspace, select a **Network Master** instance.
- 2 Invoke the context menu and choose the **Display Schedule** function.

A **Display schedule** window similar to the example below opens:



- 3 Select the year you want to display.

Dates that are marked in red indicate that a schedule is provided.

You can view an overview of network starts by proceeding as described in [Producing a Network Start Summary](#).

Adding a Schedule Definition

» To add a schedule definition

- 1 In the object workspace, select a **Schedule** node.
- 2 From the context menu, choose **New**, or press Ctrl+N.

A **Create new Schedule** window opens.

The fields and tabbed pages in the window correspond to the fields and tabbed pages of the **Maintenance Schedule window**.

- 3 Fill in the fields you want to define. The fields and commands available in the window are explained in *Fields: Schedule Definition* and *Available Functions: Create/Maintenance Schedule*, respectively.
- 4 Choose **OK** when you are finished.

The schedule is saved and you can use it for networks and jobs.

Modifying a Schedule Definition

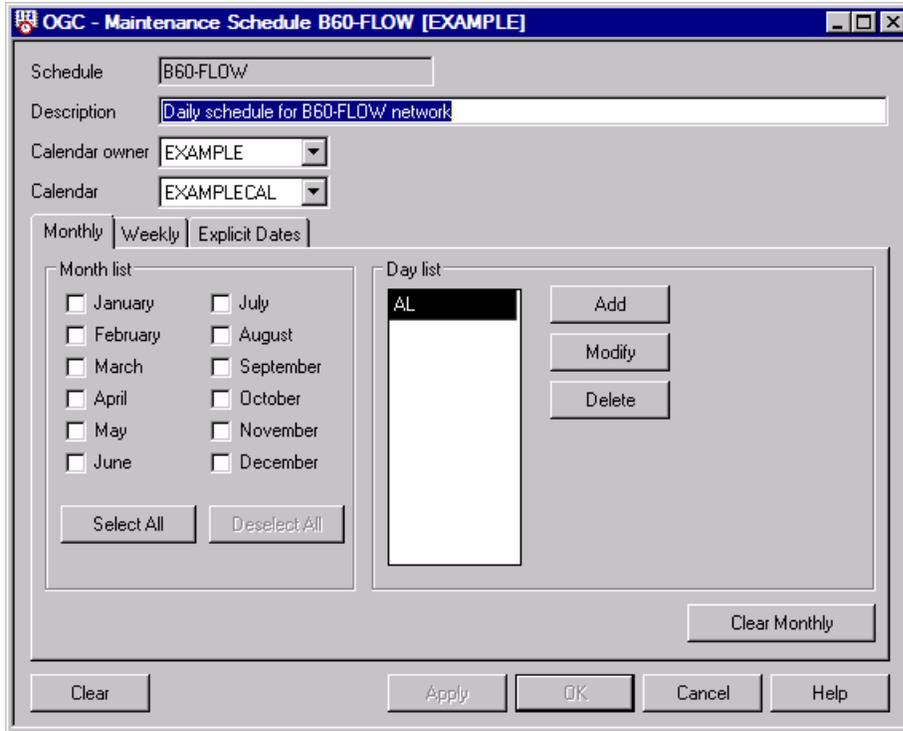
» To modify a schedule

- 1 In the object workspace, select a schedule from the **Schedule** metanode.
- 2 From the context menu, choose **Open**.

Or:

Press Ctrl+O.

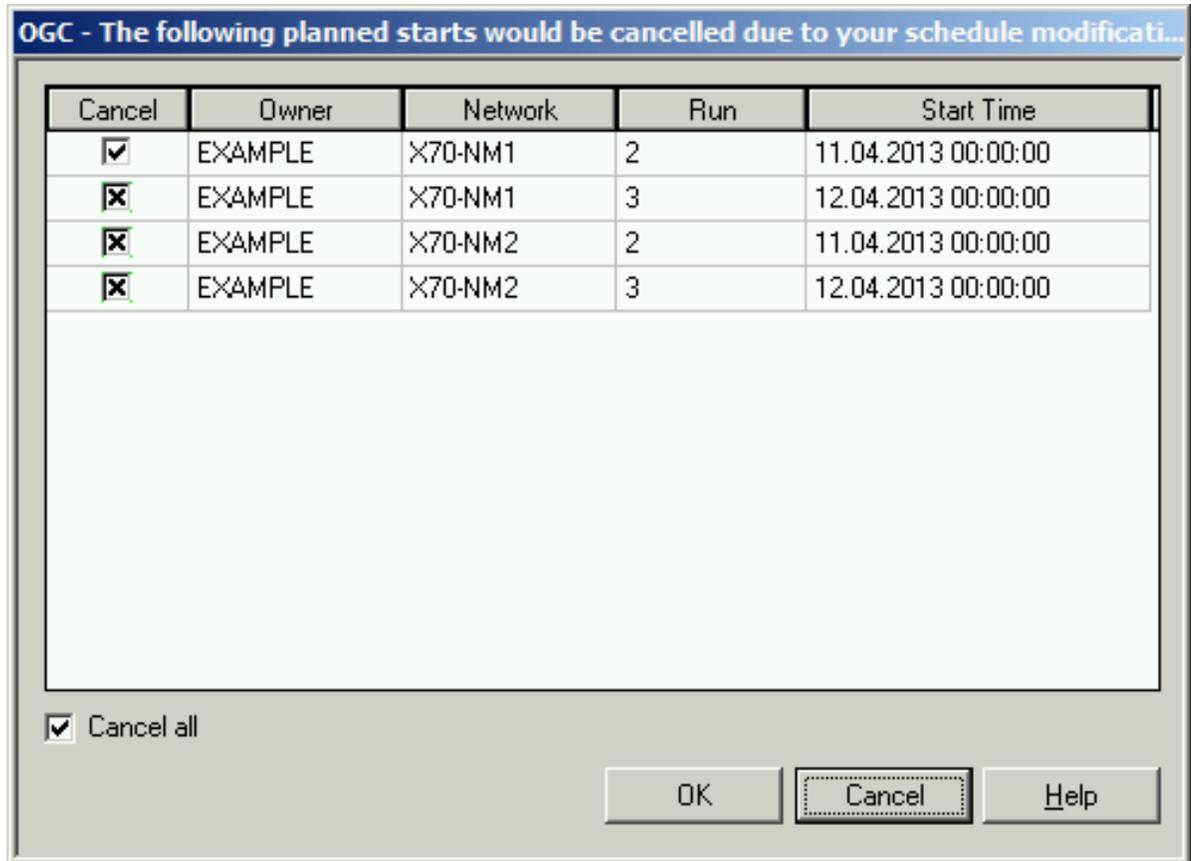
A **Maintenance Schedule** window similar to the example below opens:



- 3 Make your definitions. The fields and commands available in the window are explained in [Fields: Schedule Definition](#) and [Available Functions: Create/Maintenance Schedule](#), respectively.
- 4 Choose **OK** to save the definitions.

Or:

Choose **Cancel** to keep the previous definitions.
- 5 If your modification of a schedule affects any planned starts, a window prompts you to cancel the starts or to keep them active.



- 6 Select the **Cancel all** check box if you want to cancel all starts.

Or:

Select the **Cancel** check box next to the listed start(s) you want to cancel.

Or:

Choose **Cancel** to keep all listed starts active, regardless of your schedule modifications.

Fields: Schedule Definition

The fields and tabbed pages provided in the **Create new Schedule** or **Display/Maintenance Schedule window** are explained in the following table. Each tab within this window opens a separate schedule definition time frame (monthly, weekly and explicit dates) to make your schedule the most precise.

Field/Tabbed Page	Description
Owner	Owner of the schedule in the window title
Schedule	Name of the schedule
Description	Description of the schedule
Calendar owner	Owner of the assigned calendar Default: the schedule owner You can select a name from the drop-down list box.
Calendar	Assigned calendar You can select a name from the drop-down list box.
Monthly / Month list	List of months for the schedule They always correlate with Day list .
Weekly / Month list	A list of months for the schedule. They always correlate with Day list .
Day list	Define the days to be used for the monthly and/or weekly schedule. The days specified in this list correlate with the list of months selected on the tabbed pages Weekly and/or Monthly . See <i>Using Day Lists</i> for the options provided to define days.
Explicit Dates	Note: 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. See also <i>Defining Dates for Explicit Network Execution</i> .

Available Functions: Schedule Definition

The functions provided in the **Create new Schedule** or **Maintenance Schedule** window are explained in the following table.

Function	Description
Select All	Select all months in the current month list.
Deselect All	Deselect all months marked in the current month list.
Add/Modify (Day list)	Open the Maintenance Monthly or Maintenance Weekly window where you can select a day definition for the Day list .
Add/Modify (Explicit Dates)	Add a date or modifies a selected date. See also <i>Defining Dates for Explicit Network Execution</i> .

Function	Description
Delete (Day list)	Delete all entries in the Day list .
Delete (Explicit Dates)	Delete the selected date. See also <i>Defining Dates for Explicit Network Execution</i> .
Clear Weekly	Clear the Month list on the tabbed page Weekly .
Clear Monthly	Clear the Month list on the tabbed page Monthly .
Clear	Clear all existing definitions.
Apply	Apply changes and save the schedule.
OK	Save the schedule.
Cancel	Cancel all pending changes.

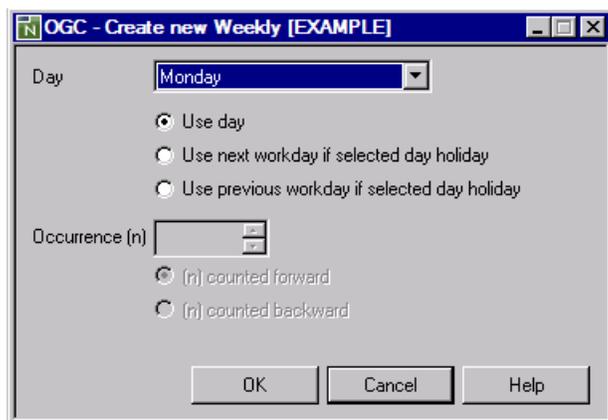
Using Day Lists

The **Day list** is used to define days for a monthly and/or a weekly schedule.

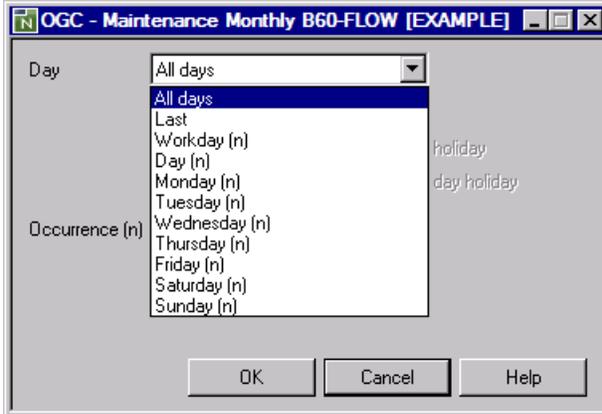
➤ To add or modify a day definition

- 1 In the **Day list** section of the **Create new Schedule** or **Maintenance Schedule** window, choose **Add** or **Modify**.

A **Monthly** or **Weekly** window similar to the example below opens:



- 2 Open the drop-down list box to select the required definition:



The day definitions available for selection depend on whether you use the **Day list** on the tabbed page **Weekly** or **Monthly**. For details, see [Possible Day Definitions](#).

- 3 Select a radio button if you want to define special conditions for a day:

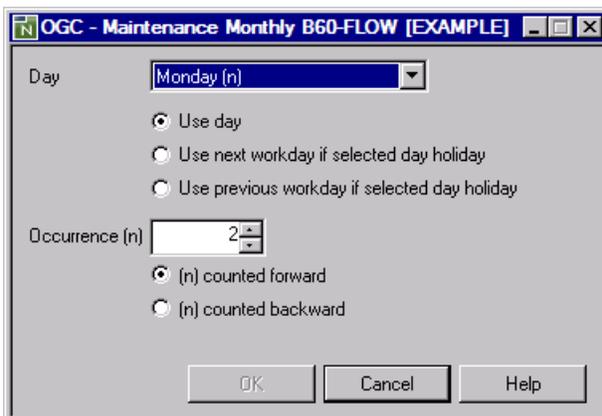
For a selected day (in the **Weekly example: Monday**) you can specify whether to use the next or previous workday if this day is a holiday.

Or:

For a monthly schedule, you can also select a day **(n)** to determine which occurrence of the weekday to use; enter a valid number in the **Occurrence (n)** box:

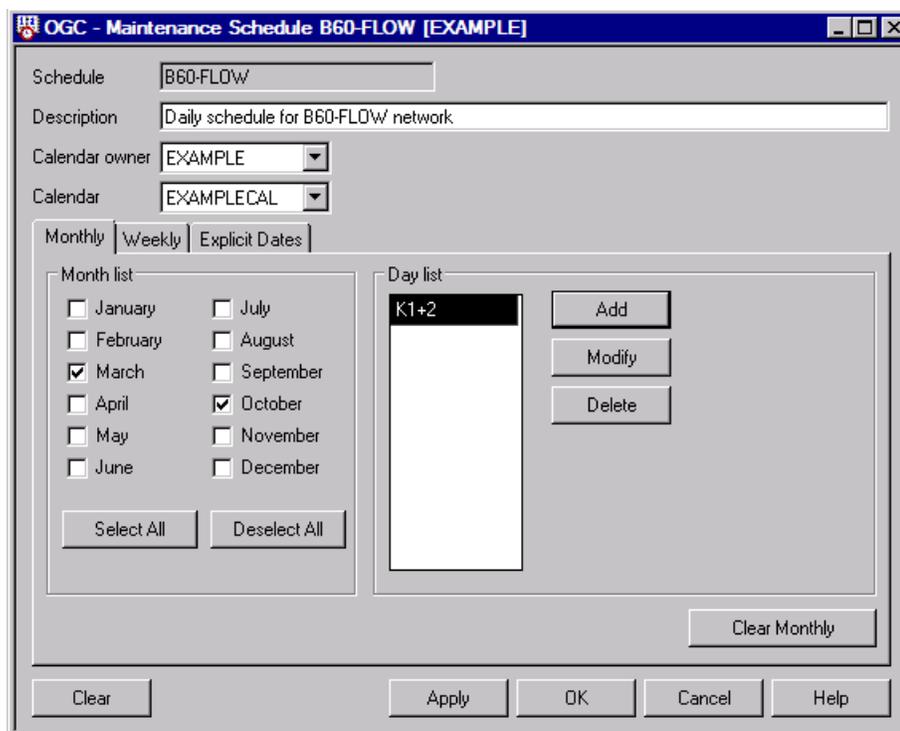
- **(n) counted forward:** counts *n* times forward in the given time range.
- **(n) counted backward:** counts *n* times backward in the given time range.

In the example below, 2 determines that the second Monday of the month is used:



- 4 Select the required definition and choose **OK**.

The **Day list** now contains the code Entire Operations automatically generates for the day definition you made:



In the example above, the **K1+2** code indicates that you defined the second Monday for the months of March and October.

Possible Day Definitions

The day definitions available for selection depend on whether you use the **Day list** on the tabbed page **Weekly** or **Monthly**. Remember that the month(s) selected on these pages always correlate with **Day list**. Each combination of month and day is a schedule date.

The following table lists possible selection options for day definitions and provides examples of codes Entire Operations generates for the defined day into the **Day list** of a **Weekly** or **Monthly** schedule:

Day	Example Weekly	Example Monthly
Weekday (Monday through Sunday)	1 is Monday and 2 is Tuesday.	K1 is each Monday, K2 each Tuesday of the month.
Workday (day defined in the calendar):		
Next workday if the specified day is a holiday	1A is the next workday after Monday.	K1A is the next workday after a Monday in a month.
Previous workday if the specified day is a holiday	1B is the workday before Monday.	K1B is the workday before a Monday in the month.

Day	Example Weekly	Example Monthly
Workday (<i>n</i>), counted forward (default setting) in the given time range	1W is the first workday of the week.	01W is the first workday of the month.
Workday (<i>n</i>), counted backward in the given time range	1V is the last workday of the week.	01V is the last workday of the month.
All days of the month	n/a	AL is all days of the month.
Day (<i>n</i>)	n/a	01 is the first day of the month.
Last day of the month	n/a	LD is the last day of the month.

For further information on the codes generated for a day definition and weekday dependency within a month, refer to the corresponding *Schedule Maintenance* section for mainframes and UNIX.

Deleting a Schedule Definition

➤ **To delete a schedule definition**

- 1 Choose **Clear** on the **tabbed page Scheduling** of the required network.

The name in the **Schedule** field and all entries on the tabbed pages **Time**, **Explicit dates** and **Repeating activation** are removed from the schedule. The **Schedule effective from** option is unchecked (deactivated).

- 2 Choose **Apply** or **OK** to confirm the deletion.

The schedule definition is deleted for the selected network and the window closes.



Note: You cannot delete a schedule, which is defined (used) in at least one network. To find out which networks are using the schedule, choose the **Where used** function for the schedule. See also *Available Functions: Schedule Maintenance*.

53 Scheduling a Network

- Defining a Network Schedule 514
- Defining and Deleting Time Frames for a Network 518
- Defining Dates for Explicit Network Execution 523
- Defining Multiple Network Activations 526
- Viewing a Network Schedule Definition as a Calendar 529
- Producing a Network Start Summary 531
- Displaying Next Network Starts - Next Activations 531
- Displaying the Network Execution History 536

Defining a Network Schedule

➤ **To define a schedule for a network**

- 1 On the **Create new Network Master** or **Maintenance Network Master** window, open the tabbed page **Scheduling** as shown in the example below:

OGC - Maintenance Network Master SAGNET [SAGTEST]

Owner: SAGTEST
 Network: SAGNET
 Version:
 Description: Test Network

Default Values for the Jobs | OS Specials | Symbol Prompting
 Granting Definition | **Scheduling** | Long Description | Message and Message Recipients

Schedule owner: SAGTEST | Schedule: SCHED-NEW

Schedule effective from: 30.09.2017 | 00:00:00

Time | Explicit dates | Repeating activation

Schedule dep.	Earliest start	Latest start	Days later	Deadline	Days later
+WY+001	08:00:00	09:00:00	1	10:00:00	2
	09:25:00	15:00:00	1	16:00:00	2

Buttons: Add, Modify, Delete, Move up, Move down

Clear

Apply | OK | Cancel | Help

Define a new schedule or modify an existing one. The fields and tabbed pages available in the window are explained in [Fields: Network Scheduling](#).

- 2 Choose **OK** when you are finished.

This section covers the following topics:

- Fields: Network Scheduling

Fields: Network Scheduling

The input fields of the **tabbed page Scheduling** of the **Create new Network Master** or **Maintenance Network Master** window are described in the following table:

Field/Tabbed Page	Description
Schedule Owner	Owner of the schedule to be used. 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.
Schedule effective from	Schedule effective from (date and time) Select this field (not selected by default) to activate the schedule and/or change the 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.
Time page:	
Schedule dep.	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 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 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 Behavior for Schedule Times .

Field/Tabbed Page	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>
Explicit dates page:	See Defining Explicit Dates for a Single Network only .
Repeating activation page:	See Defining Multiple Network Activations .
Number of Activations	<p>This field is used to activate a network more than once a day. In this case, select the button next to this field and 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>
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 H:MM</p> <p>Examples: 20:11, 1:31</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>

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 On the tabbed page **Time** of the tabbed [page Scheduling](#).
- 2 Choose **Add**.

Or:

Select a time frame definition from the table and choose **Modify**.

Depending on the function chosen, a **Create new Time Frame** or **Maintenance Time Frame** window similar to the example below opens:

OGC - Maintenance Time Frame SAGNET [SAGTEST]

Schedule dependency: +WY +1 [Modify]

Earliest start: 08:00:00

Latest start: 09:00:00 [Days later: 1]

Deadline: 10:00:00 [Days later: 2]

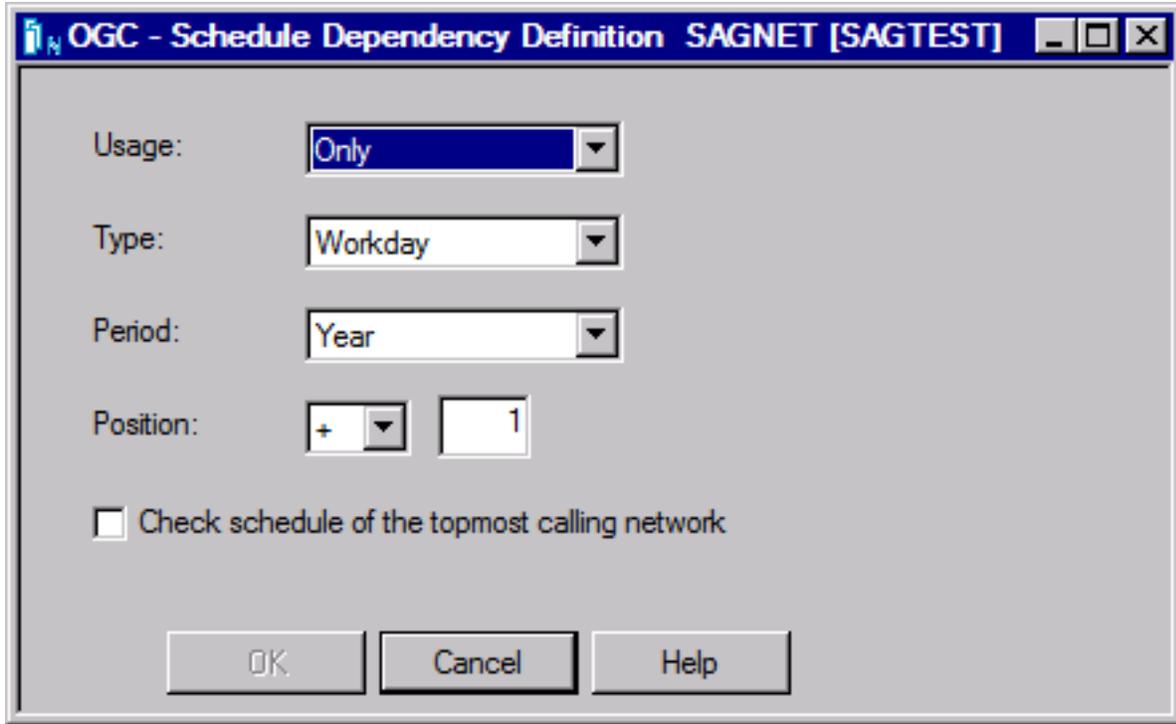
[OK] [Cancel] [Help]

3. Enter the required values. The input fields contained in the window are described in *Fields: Network Scheduling*.

See also *Time Frame Selection Rules for a Network* and *Behavior for Schedule Times*.

4. If you want to add or modify the schedule dependency for a time frame, choose **Modify** next to **Schedule dependency**.

A **Schedule Dependency Definition** window similar to the example below opens:



The input fields contained in the window are described in [Fields: Schedule Dependency Definition](#).

- 5 Choose **OK** when you are finished.

The time frame definition is shown in the table of the **Time** page as shown in the example below:

OGC - Maintenance Network Master SAGNET [SAGTEST]

Owner: SAGTEST

Network: SAGNET

Version:

Description: Test Network

Default Values for the Jobs | OS Specials | Symbol Prompting

Granting Definition | Scheduling | Long Description | Message and Message Recipients

Schedule owner: SAGTEST | Schedule: SCHED-NEW

Schedule effective from: 30.09.2017 | 00:00:00

Time | Explicit dates | Repeating activation

Schedule dep.	Earliest start	Latest start	Days later	Deadline	Days later
+WY+001	08:00:00	09:00:00	1	10:00:00	2
	09:25:00	15:00:00	1	16:00:00	2

Buttons: Add, Modify, Delete, Move up, Move down

Clear

Apply | OK | Cancel | Help

If a schedule dependency is defined for the time frame definition, a code is shown in the **Schedule dep.** column (here: +WY+001).

- You can move up or down a selected column row by choosing **Move up** or **Move down**, respectively.

You can choose **Delete** for a selected column row to remove a time frame definition.

Choose **Clear** to remove all entries from the table.

- 7 Choose **OK** when you are finished.

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 *Defaults for Time Ranges* 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 **Time Ranges** page 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 time frame dependency from a time frame definition

- 1 Open the tabbed page **Time** of the tabbed page **Scheduling**.
- 2 Select the table row that contains the definition you want to delete and choose **Delete**.



Note: **Clear** removes all **Scheduling** entries as described in [Deleting a Schedule Definition](#).

- 3 Choose **Apply** or **OK** to confirm the deletion and close the window.

The time frame definition is removed from the table.

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:

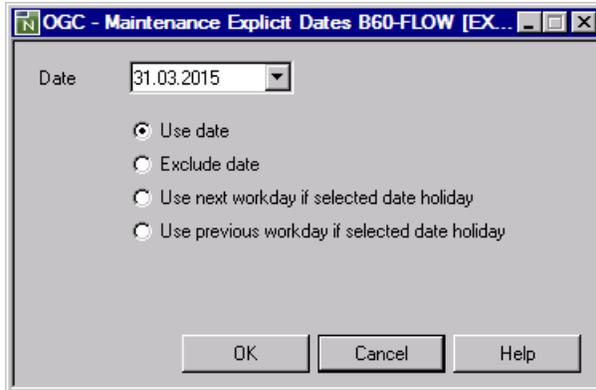
- [Defining Explicit Dates for a Schedule](#)
- [Defining Explicit Dates for a Single Network only](#)

Defining Explicit Dates for a Schedule

› To add or modify explicit dates for a schedule

- 1 In the **Create new Schedule** or **Maintenance Schedule window**, open the tabbed page **Explicit Dates**.
- 2 Choose **Add** or **Modify**.

An **Explicit Dates** window similar to the example below opens:



- 3 Open the drop-down list box and select the required date or replace the date entered in the format DD.MM.YYYY.

If you want to associate the date with a special condition, select one of the following options:

- Exclude date: do not use the specified day.
- Use the next workday if the specified date is a holiday.
- Use the previous workday if the specified date is a holiday.

- 4 Choose **OK**.

The **Explicit Dates** window closes and the specified date is shown on the **Explicit Dates** page.

- 5 Choose **OK** to save the date specification.

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" (**Exclude day**) in this table (see also [Fields: Network Schedule – Explicit Dates](#)).

➤ To define explicit dates for a single network

- 1 In a **Create new Network Master** or **Maintenance Network Master** window, open the **tabbed page Scheduling**.
- 2 Open the tabbed page **Explicit dates**.

A table similar to the example below is provided in the window:

Time	Explicit dates	Repeating activation
16-06-22	Execution on the workday before	
16-07-28	Execution on the workday after	
16-08-02	Exclude date	

- 3 In the input field of the left column, choose a date from the calendar to add or change the date you want to explicitly use for the selected network.

From the drop-down list box in the right column, select a condition for the specified date, if required.

See also *Fields: Network Schedule – Explicit Dates*.

You can choose **Clear** or **Delete** to remove a selected table entry, or choose **Clear All** to remove all table entries.

- 4 Choose **OK** when you are finished.

Fields: Network Schedule – Explicit Dates

The input fields in the columns on the **Explicit dates page** are described in the following table:

Field	Description	
<i>date</i> (left column)	In the left table column, you can enter a date in the date format specified within Entire Operations defaults.	
<i>condition</i> (right column)	From the drop-down list box in the right table column, you can select one of the following conditions for the date specified in the left column:	
	Execution on the workday before	Execute network on the workday that precedes the specified date.
	Execution on the workday after	Execute network on the workday that follows the specified date.
	Exclude day	Do not execute the network on the specified date.

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

- 1 On the tabbed page **Scheduling**, open the tabbed page **Repeating activation**:

2 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:

Enter (or replace) the following:

- In the input fields next to **Activate at**, enter single or multiple times to be used for network activation.

You can choose **Clear** to remove all entries from the **Repeating activation** page.

- 3 If required, select or enter a time limit for activation in the **Day Deadline**.
- 4 Choose **OK** to save your entries and close the window.

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.



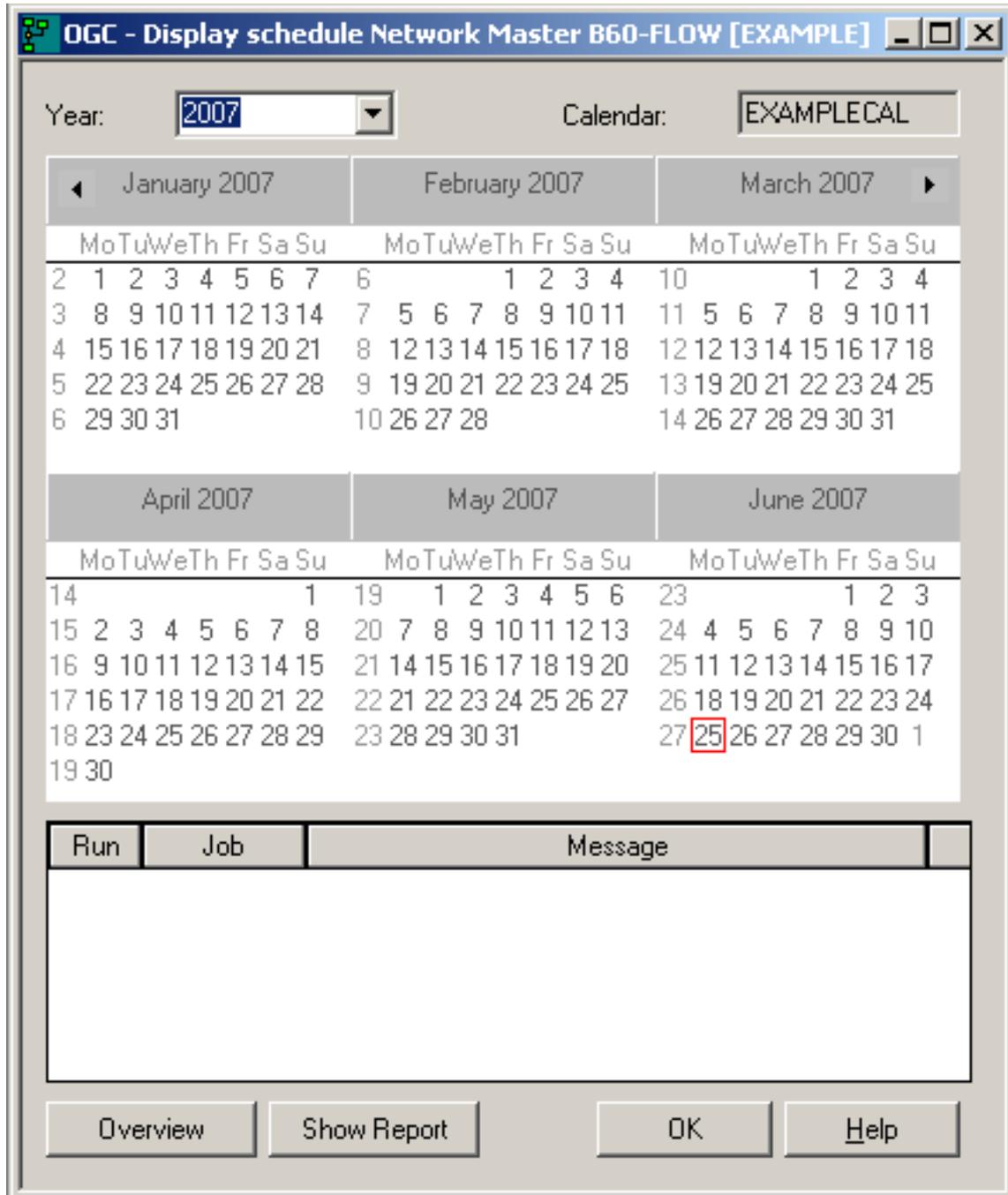
Note: A network is no longer treated as “already activated on the current day” if there was a multiple activation on the day before, which resulted in some runs that were executed after midnight. In this case, automatic activations (due to schedule) will not be blocked by the “activation on current day” check.

Viewing a Network Schedule Definition as a Calendar

➤ To display the calendar view of a network schedule

- 1 In the object workspace, select a **Network Master** instance.
- 2 Invoke the context menu and choose the **Display schedule** function.

A window similar to the example below opens:



- 3 Select the year you want to display.

Dates outlined in red indicate that a schedule is provided.

- 4 Select a date and choose the **Overview** button.

All activities scheduled for this date are displayed for online viewing in the lower area of the window.

- 5 Select a date and choose the **Show Report** button.

A list of available reports is displayed in a separate window.

Select the required report for printing purposes.



Note: For further information on reports, see the section [Reporting](#).

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](#).

➤ To produce a network start summary for the current network

- 1 In the [Display schedule window](#), select a valid date and choose **Overview**.

All activities scheduled for this date are displayed for online viewing in the lower area of the window.

- 2 Select a date and choose the **Show Report** button.

A list of available reports is displayed in a separate [Reporting window](#).

- 3 Select the required report and choose **Save as File** from the context menu and determine the file to be created from the report: see [Report Output Options](#).

For detailed information on generating reports, see the section [Reporting](#).

Displaying Next Network Starts - Next Activations

Scheduled starts are extracted from the network schedules and put into a pre-activation status. The extraction is usually performed at midnight, one day in advance.

Manual starts are explicitly invoked as manual activations by any user. If a special start time was not entered, the daily time frame is used. See [Activating a Job Network Manually](#) in the section [Network Maintenance](#).

This section covers the following topics:

- Viewing and Maintaining Next Network Activations
- Columns: Next Activations

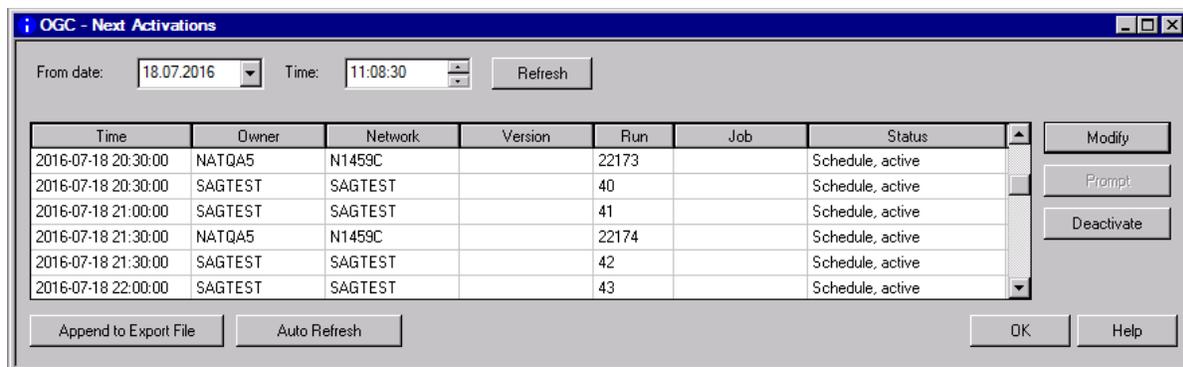
Viewing and Maintaining Next Network Activations

The context menu function **Next Activations** invokes the display of the future network activations. These activations are, in general, maintained by a schedule or a calendar but can also be invoked manually. You can view, modify or delete a planned activation.

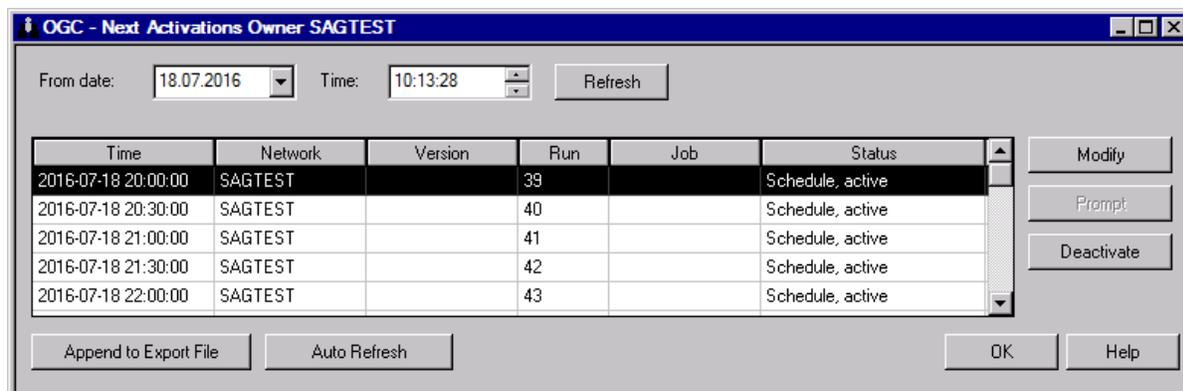
➤ To view and modify the next network activations

- 1 In the **object workspace**, select a **General** node, an **Owner** instance or a **Network Master** instance.
- 2 Invoke the context menu and choose the **Next Activations** function.

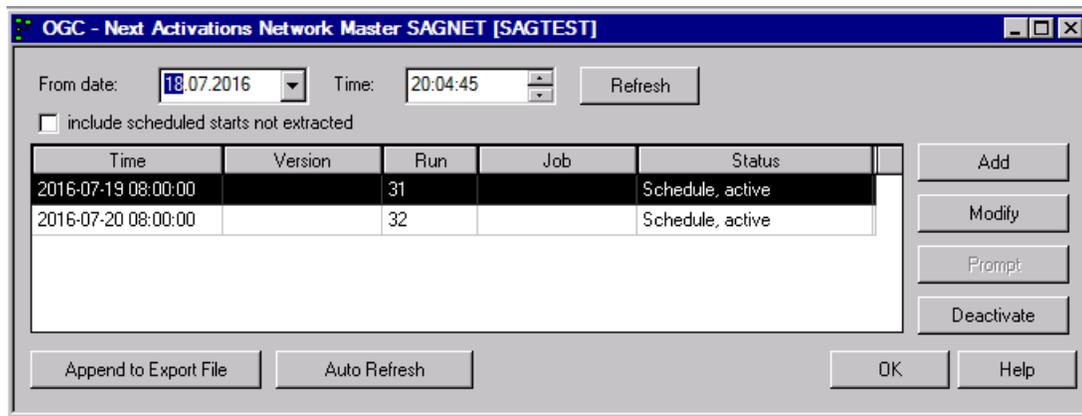
For a **General** node, the **Next Activations** window lists all network activations planned for all networks in your current Entire Operations environment as shown in the following example:



For an **Owner** instance, the **Next Activations** window lists all network activations planned for all networks that belong to the selected owner. Example:



For a **Network Master** instance, the **Next Activations** lists all activations planned for the selected network master. Example:



The **Next Activations** window displays a chronological list of all planned network or job starts, both scheduled and manual.



Note: The table view is filled dynamically with data if scrolled to the end of the table.

The columns contained in the window are described in [Columns: Next Activations](#).

- 3 You can refresh the list by choosing **Refresh** and **Auto Refresh** (see also [Refreshing Object Lists](#)).
- 4 You can cancel a future activation by selecting the respective row and choosing **Deactivate**.

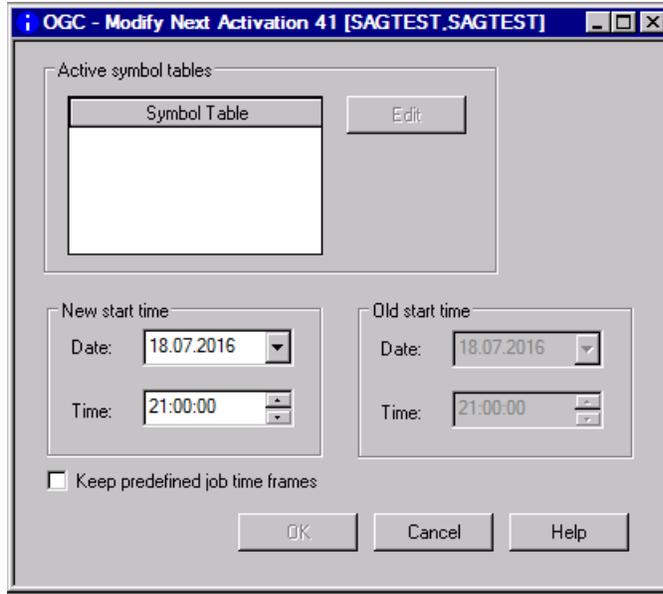
You can only cancel a job network to which a run number is assigned.

- 5 You can prompt symbols (if defined) by selecting the respective row and choosing **Prompt**.

This button is only active for activation with type `Aw.Symbol Prompting` and if a run number is assigned to the network.

- 6 You can edit a future activation by selecting the respective row and choosing **Modify**.

A window similar to the example below opens:



This window allows you to define a new start date and time for execution, edit a symbol table as used for this activation (see also [Displaying and Modifying a Master or Active Symbol Table](#)), and select the option to **keep predefined job time frames**.

➤ To add a next activation

Next activations can only be added for a network master because the objects have to be specified in advance of any activation.

- 1 In the **object workspace**, select a **Network Master** instance.
- 2 Invoke the context menu and choose the **Next Activations** function.

The **Next Activations** window opens.

- 3 Select the **Add** button.

The **Activation Network Master** window opens.

- 4 Define the time schedule for activation.
- 5 Select **OK**.

The network is now marked and scheduled for its next activation and appears in the activations list.

➤ To keep predefined job time frames

- If you select the check box **Keep predefined Job Time Frames**, Jobs with master time frame definitions will not be adapted.

Or:

If you do not select the check box **Keep predefined Job Time Frames**, all job time frames will be adapted (default setting).

Columns: Next Activations

The columns contained in the **Next Activations** window depend on whether you execute the **Next Activations** function from the **General** node or an **Owner** instance (see also [Viewing and Maintaining Next Network Activations](#)). All columns that can be available are described in the following table:

Column	Description
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.
Network	Network to be activated 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. Your selection is stored in the user profile and is available when you invoke this function the next time.
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	Date of activation 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.
Start	Starting time of 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, ■ Aw. Symbol Prompting, ■ Symbol Entry in Progress.

Displaying the Network Execution History

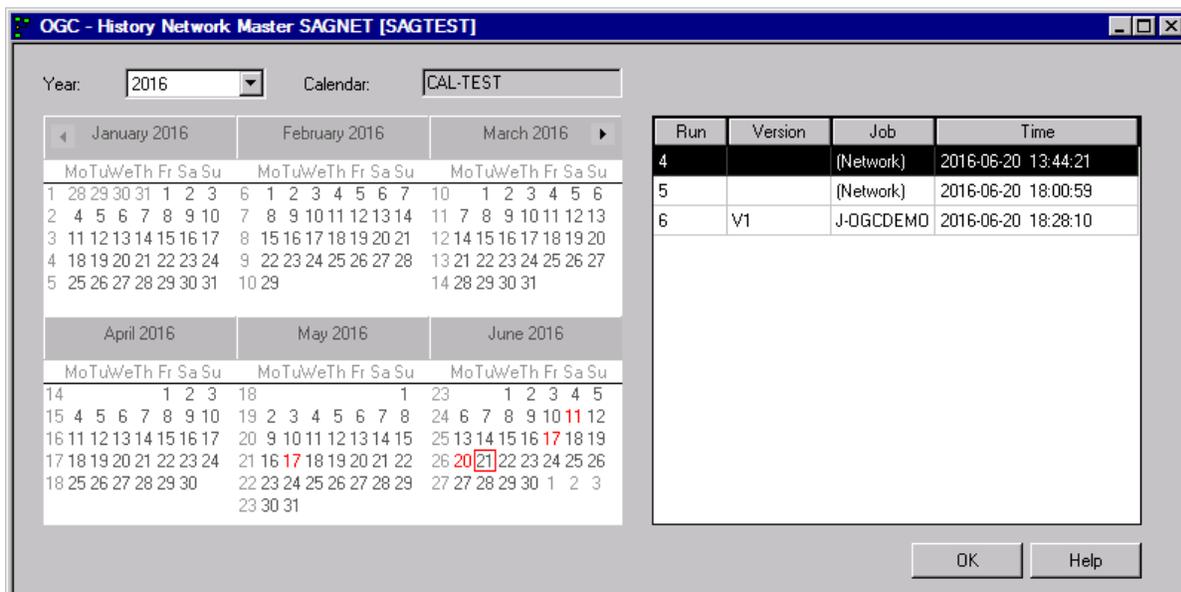
The execution history is a record of past schedules. It shows the days on which the network was scheduled for execution. The data in a **History** window cannot be modified.

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 view the history of a network master

- 1 In the **object workspace**, select a **Network Master** instance.
- 2 Invoke the context menu and choose the **History** function.

A **History Network Master** window similar to the example below opens:



A calendar is displayed where the dates on which the network ran are highlighted in red.

- 3 If available, select a year from the drop-down-list box next to **Year** to switch the display of the calendar to the selected year.
- 4 Click on the highlighted date for which you want to view the history data.

If history data exists for the selected network and date, execution information on all network starts is contained in the right-hand table columns of the window as shown in the example above. The columns are explained in [Columns: Network History](#).

If no history data exists, an appropriate message occurs. Log data is only retained for the number of days specified in the **long term log** field of the **Entire Operations Defaults** described in *Defaults for Time Ranges* in the *Administration* documentation.

This section covers the following topic:

- [Columns: Network History](#)

Columns: Network History

The columns of the [History Network Master](#) or **History Network Active** window 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.
Time	Date and time when the network executed. See also Date and Time Formats .

54 Scheduling a Job

- Defining Scheduling Parameters for a Job 540
- Maintaining Schedule Dependencies for a Job 542
- Defining Schedule Dependencies for an Input Condition 546
- Sending a Late Message to One or More Users 546
- Viewing Job Accounting Data 547

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 for a job

- 1 In the object workspace, open a **Job Master** instance.
- 2 In the **Job Maintenance** window, select the tabbed page **Scheduling Parameters**:

The screenshot shows the 'OGC - Maintenance Job Master JOB-01 [SAGTEST.B60-FLOW]' window. The 'Scheduling Parameters' tab is active. The window contains the following fields and controls:

- Job name: JOB-01
- Use symbol as execution node:
- Execution node: 42 QANODE42 (MVS/ESA)
- Description: Where it all starts
- Job type: Standard Job
- Special type: Cyclic execution
- Estimated elapsed time: [] min. Average: []
- History elapsed time (min.): []
- Earliest start time: 13:14:00 [] days later [] Late Message
- Latest start time: 23:00:00 [] days later [] Account Info
- Deadline time: 23:30:00 [] days later []
- Schedule dependency: B H1 -1 [] Modify
- Cyclic interval: [] 0

Buttons at the bottom: Apply, OK, Cancel, Help.

- 3 Make your definitions. The fields contained in the window are explained in [Fields: Scheduling Parameters](#).
- 4 Choose **OK**.

This section covers the following topics:

- [Fields: Scheduling Parameters](#)

Fields: Scheduling Parameters

The fields of the [tabbed page Scheduling Parameters](#) 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 Dummy Job), 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 Subnetwork, 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 (min.)).
History Elapsed Time (min.)	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> <p>See also Behavior for Schedule Times.</p>
Latest Start Time	<p>The job will not be started after this time.</p> <p>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.</p> <p>See also Behavior for Schedule Times.</p>
Deadline Time	<p>Time by which the job must be finished</p> <p>If the job cannot be finished before this time, it is not submitted.</p>

Field	Description
	Dummy jobs with Estimated Elapsed Time not equal to zero (0) are finished at this time at the latest. See also 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 Behavior for Schedule Times .
Schedule Dependency	A code shown in this read-only field indicates that this job has a schedule dependency. The field is empty if no schedule dependency is defined for the job. You can choose Modify to open a window where you can view, add or modify a schedule dependency. 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 Cyclic execution). Default: Monitor wait time
Late Message	See Sending a Late Message to One or More Users .
Account Info	See Viewing Job Account Data .

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 on the [Scheduling Parameters page](#) 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 [Check schedule of the topmost calling network](#) 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

- On the tabbed page [Scheduling Parameters](#) of the **Maintenance Job Master** window, choose **Modify** next to **Schedule dependency**.

A **Schedule Dependency Definition** window similar to the example below opens:

The screenshot shows a dialog box titled "OGC - Schedule Dependency Definition SAGNET [SAGTEST]". It contains the following fields and controls:

- Usage:** A dropdown menu with "Only" selected.
- Type:** A dropdown menu with "Workday" selected.
- Period:** A dropdown menu with "Year" selected.
- Position:** A dropdown menu with a "+" icon and a text box containing "1".
- Check schedule of the topmost calling network:** An unchecked checkbox.
- Buttons:** "OK", "Cancel", and "Help" buttons at the bottom.

The input fields and options 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			
Check schedule of the topmost calling network	Possible check box settings:			
	<table border="0"> <tr> <td style="vertical-align: top;"><i>unchecked</i></td> <td>Checks the schedule of the current network (default). This setting is recommended for jobs in main networks.</td> </tr> <tr> <td style="vertical-align: top;"><i>checked</i></td> <td>Checks the schedule of the topmost network that calls the job. This setting is recommended for jobs in subnetworks.</td> </tr> </table>	<i>unchecked</i>	Checks the schedule of the current network (default). This setting is recommended for jobs in main networks.	<i>checked</i>
<i>unchecked</i>	Checks the schedule of the current network (default). This setting is recommended for jobs in main networks.			
<i>checked</i>	Checks the schedule of the topmost network that calls the job. This setting is recommended for jobs in subnetworks.			

Field	Description	
Usage	Possible selection options:	
	Only	For a job or condition: The job or condition is activated only if the execution date satisfies the criteria entered in the Type, Period and Position fields.
		For a network: The object is valid only for schedule days matching the following definitions.
	After holiday too	Like Only , but the first workday after an otherwise matching holiday is valid, too.
	Before holiday too	Like Only , but the last workday before an otherwise matching holiday is valid, too.
	Except	For a job or condition: The job or condition is activated except when the execution date satisfies the criteria entered in the Type, Period and Position fields.
For a network: The object is valid for all schedule days except if they match the following definitions.		
Type	Possible selection options:	
	Is holiday	Any holiday (no period and/or position required) The relative position to the holiday can be specified.
	Is workday	Any workday (no period and/or position required) The relative position to the workday can be specified.
	Calendar day	Calendar day within the defined period A relative position refers to the real calendar period.
	Workday	Workday within the defined period A relative position refers to the workdays in the period.
	Schedule day	Schedule day within the defined period A relative position refers to the schedule days in the period.
Period	Possible selection options:	
	Week	Current week
	Month	Current month
	Year	Current year

Field	Description
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 Is holiday and Is workday, 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
Calendar day	Week	+2	Tuesday
		-1	Sunday
	Month	+1	01
		-1	28 ... 31
	Year	+1	January 01
		-3	December 29
Workday	Month	-1	Last workday of the month
Is holiday	n/a	+1	Day after a holiday

Deleting Schedule Dependency Entries

» To delete a schedule dependency entry for a job

- 1 On the tabbed page [Scheduling Parameters](#), choose **Modify** next to the entry in the **Schedule dependency** field.

The [Schedule Dependency Definition](#) window opens.

- 2 Clear the **Usage** field by selecting a blank option from the drop-down list box.

All entries are removed from the window.

- 3 Choose **OK**.

The **Schedule Dependency Definition** window closes and the **Schedule dependency** entry is removed from the **Scheduling Parameters** page.

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 From the table on the **tabbed page Input Conditions**, select an input condition and choose **Schedule Dep..**

A **Schedule Dependency Definition 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*.

- 2 Make your definitions.
- 3 Choose **OK**.

The schedule dependency is allocated to the input condition and the window closes.

The schedule dependency is now indicated in the **Sched.Dep.** column.

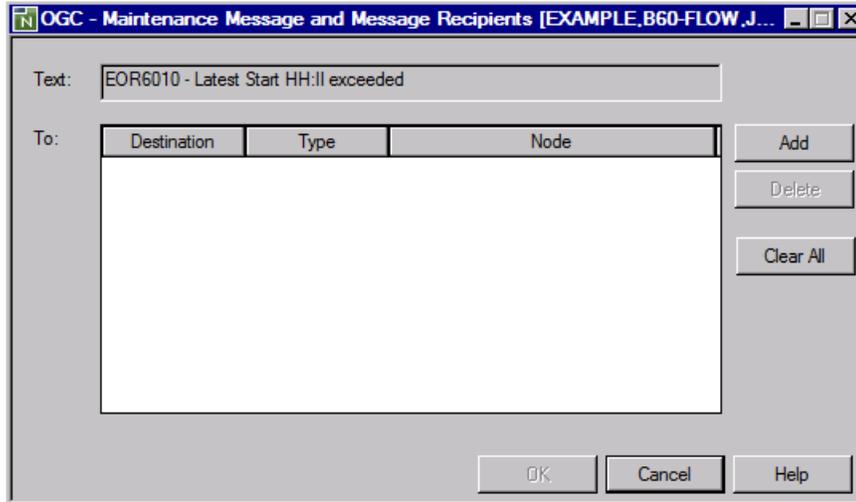
Sending a Late Message to One or More Users

You can define recipients (users) to be informed in the case that the job could not be executed in the defined time range.

➤ To add a late message definition

- 1 On the tabbed page **Scheduling Parameters**, choose **Late Message**.

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



- 2 Choose **Add** to enter new destinations for the message in the table columns.

The columns in the window are explained in *Field and Columns: Message and Message Recipients* in the section *End-of-Job Checking and Actions*.

- 3 Choose **OK**.

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.

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

» To view accounting data

- 1 On the tabbed page **Scheduling Parameters**, choose **Account Info**.

An **Accounting Data List Info** window similar to the following example opens.

- 2 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*.
- 3 Select **Refresh**.

The **Accounting Data List Info** window is refreshed and the accounting information is listed as shown below:

Job	Run	JobId	Date	Start	Stop	Elapsed min	CPU Tm sec
JOB-01	2117	12771	25.06.07	13:14:24	13:15:24	1.00	0.00

The window 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 window.

For further information, see [Fields and Columns: Reporting](#) in the section *Reporting*.

- 4 Choose **OK**.

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 **Accounting Data List Info** window are described in the following table:

Field	Description
From Date/Time To Date/Time	Enter or select start (From) and end (To) dates and times to specify the time frame for which to display accounting data. Default is the current date, from midnight until the current time.
From Run/To Run	Enter or select start (From) and end (To) run numbers to specify the number range for which to display accounting data. Default is from 1 to 99999.
Averages from ... to ... are	Read-only field at the bottom of the window listing average consumption values which correspond to the values in the field History Elapsed Time (min.) .

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.

XI

Calendar Maintenance

55

Calendar Maintenance

- General Rules and Restrictions 554
- Available Functions: Calendar 555
- Listing Calendars 556
- Displaying, Modifying or Adding a Calendar Definition 557
- Defining Workdays and Holidays 559
- Where Used - Listing Schedules Using a Calendar 560
- Deleting a Calendar Definition 561

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*).

Available Functions: Calendar

➤ **To list all functions available for a Calendar node**

- In the object workspace, select a **Calendar** node and open the context menu.

The following functions are available:

Function	Shortcut	Description
List	F8	See <i>Listing Calendars</i> .
New	CTRL+N	See <i>Displaying, Modifying or Adding a Calendar Definition</i> .
Refresh	F5	See <i>Refreshing Object Lists</i> .
Filter	F3	See <i>Filtering Objects</i> .
Paste data	CTRL+V	See <i>Pasting Objects</i> .
Export	---	See <i>Exporting Objects</i> in the <i>Import/Export Functions</i> documentation.
Set Drag And Drop Function	---	See <i>Drag & Drop</i> .

➤ To list all available functions for a Calendar instance

- In the object workspace, select a **Calendar** instance and open the context menu.

The following functions are available:

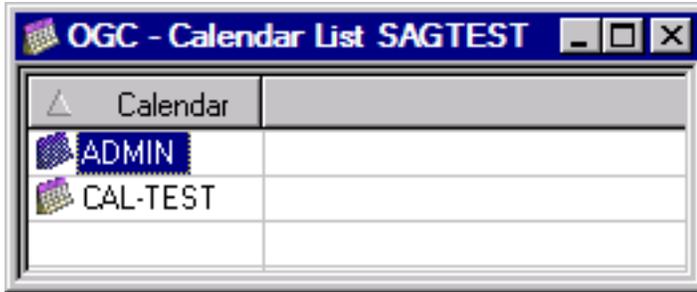
Function	Shortcut	Description
Open	CTRL+O	See <i>Displaying, Modifying or Adding</i> .
Display	CTRL+D	See <i>Displaying, Modifying or Adding</i> .
Where Used	---	Displaying schedules using a calendar. See <i>Where Used - Listing Schedules Using a Calendar</i> .
Delete	DELETE	See <i>Deleting a Calendar Definition</i> .
Copy data	CTRL+C	See <i>Copying Objects</i> .
Export	---	See <i>Exporting Objects</i> in the <i>Import/Export Functions</i> documentation.
Add to Workplan	---	See <i>Add to Workplan</i> .
Set Drag And Drop Function	---	See <i>Drag & Drop</i> .

Listing Calendars

➤ To list calendars

- 1 In the object workspace, select the **Calendar** node of the required owner
- 2 From the context menu, choose **List**, or press F8.

All calendars defined for the selected owner are listed in the **Calendar List** window as shown in the following example:



The list of calendars contains system-wide calendars and calendars belonging to the selected owner (here: SAGTEST).

A calendar is identified uniquely by its owner name and year.

This section covers the following topics:

Displaying, Modifying or Adding a Calendar Definition

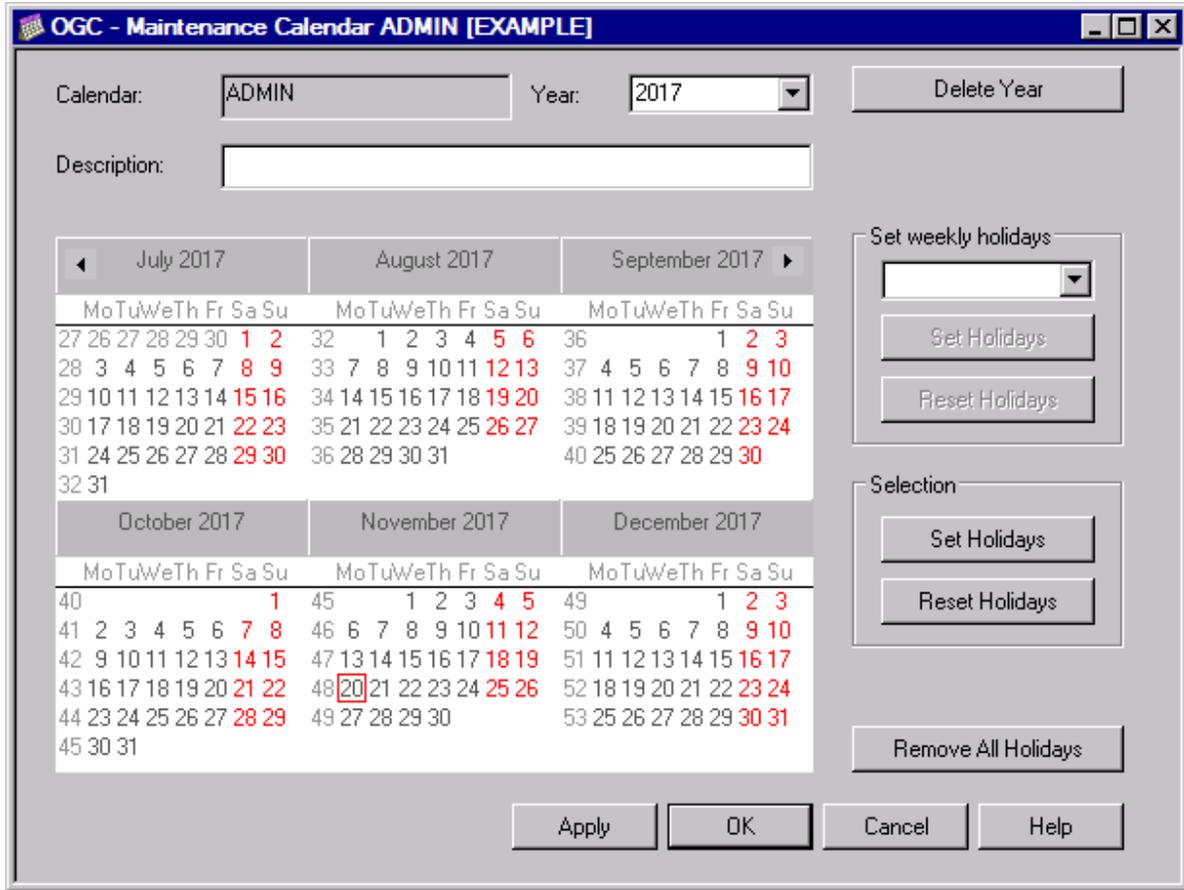
➤ To display, modify or add a calendar definition

- 1 From the **Calendar** node in the object workspace, select the calendar you want to view or modify and choose **Display** or **Open**, respectively, from the context menu. Alternatively, you can press **CRTL+D** or **CRTL+O**, respectively.

Or:

Select a **Calendar** node and choose **New** from the context menu, or press **CRTL+N**.

Depending on the function performed, a **Display Calendar**, **Maintenance Calendar** or **Create new Calendar** window opens:



If the calendar is for the current year, the window displays six months from the current month of the specified calendar year by default.

For other years, the first six months are displayed initially.

Click on the right arrow button  to scroll right in the calendar and view the next months, or click on the left arrow button  to scroll left and view the previous months.

The current date is indicated by a red square, holidays are highlighted in red.

The fields in the upper section of the window are explained in [Fields: Calendar Definition](#).

- 2 If defined for the calendar, you can select another year from the **Year** drop-down list box.
When modifying or adding a calendar, you add another year by replacing the current entry in the **Year** field.
- 3 When modifying or adding a calendar, define the required holidays as described in [Defining Workdays and Holidays](#).
- 4 Choose **OK** to save your calendar definition.

The calendar is saved and you can use it for scheduling network and jobs.

This section covers the following topics:

- [Fields: Calendar Definition](#)

Fields: Calendar Definition

The fields contained in the upper section of the **Display Calendar**, **Maintenance Calendar** and **Create new Calendar** window are described in the following table:

Field	Description
Calendar	Calendar name.
Year	Calendar year. The current year is entered by default.
Description	A short description of the 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 or remove weekly holidays

- 1 In the **Set weekly holidays** section of the [Maintenance Calendar](#) or **Create new Calendar** window, open the drop-down list box, select the weekday you want to define as a holiday and choose **Set Holidays**.

Or:

In the **Set weekly holidays** section of the [Maintenance Calendar](#) or **Create new Calendar** window, open the drop-down list window and select the defined holiday you want to change to a weekday and choose **Reset Holidays**.

Or:

Choose **Remove All Holidays**.



Caution: This function removes both all weekly and all special holidays for the year currently selected and shown in the **Maintenance Calendar** window.

- 2 Choose **OK** to save your calendar definition.

Weekdays defined as holidays are now highlighted in red.

» To define or remove special holidays

- 1 In the **Selection** section of the **Maintenance Calendar** or **Create new Calendar** window, click on single or multiple (SHIFT-click) dates and choose **Set Holidays**.

The selected dates are highlighted in red.

Or:

In the **Selection** section of the **Maintenance Calendar** or **Create new Calendar** window, click on single or multiple (SHIFT-click) highlighted (defined) dates and choose **Reset Holidays**.

Or:

Choose **Remove All Holidays**.



Caution: This function removes both all weekly and all special holidays for the year currently selected and shown in the **Maintenance Calendar** window.

- 2 Choose **OK** to save your calendar definition.

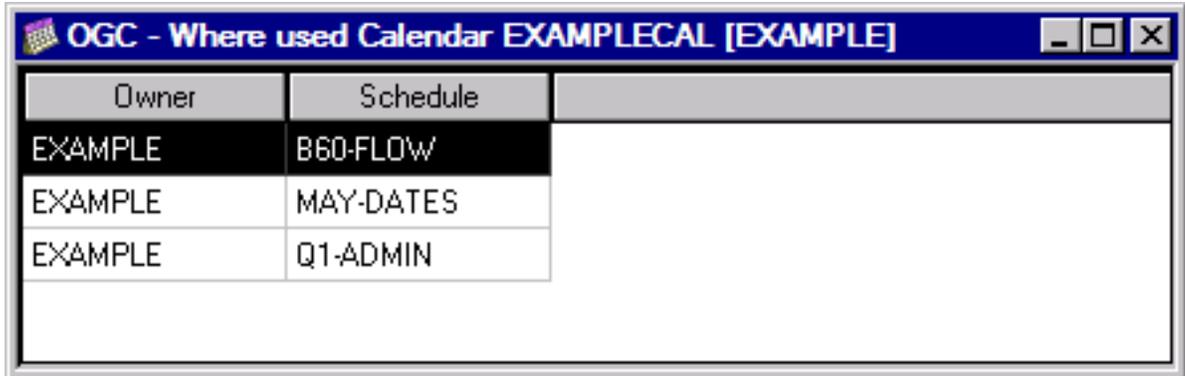
Weekdays defined as holidays are now highlighted in red.

Where Used - Listing Schedules Using a Calendar

» To list schedules using a calendar

- 1 In the object workspace, select the required calendar from the **Calendar** node and choose **Where used** from the context menu.

A **Where used** window similar to the example below opens:



Owner	Schedule
EXAMPLE	B60-FLOW
EXAMPLE	MAY-DATES
EXAMPLE	Q1-ADMIN

The window indicates the name of the selected calendar (here: EXAMPLECAL) and its owner (here: EXAMPLE) and lists all schedules (and their owners) associated with the calendar.

- 2 You can modify a listed schedule by selecting the required table row and choosing **Open** from the context menu.

A **Maintenance Schedule window** opens where you can change or remove the name of the calendar entered for the schedule.

Deleting a Calendar Definition

➤ To delete a calendar definition

- 1 In the object workspace, select the calendar you want to delete from the **Calendar** node.
- 2 From the context menu, choose **Delete**.

Or:

Press DELETE.

A dialog opens where you have to confirm 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 **Where used** function 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.

XII **Working with Mailboxes**

56 Working with Mailboxes

- Available Functions: Mailbox Definition 566
- Viewing Mailbox Messages 567
- Fields, Columns and Functions: Messages and Requests 568
- Message Types 570
- Origin of Mailbox Messages 571
- Group Mailbox 572
- SYSDBA Mailbox 572
- User Mailbox 572
- Clearing Mailboxes 572

Mailboxes are used for message switching to Entire Operations users. 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. 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.

Available Functions: Mailbox Definition

➤ To list all available functions for the Mailbox Definition metanode

- In the object workspace, select the **Mailbox Definition** metanode and open the context menu.

The following functions are available:

Function	Shortcut	Description
List	F8	Lists mailboxes: see <i>Listing Mailboxes defined to Entire Operations</i> in the <i>Administration</i> documentation.
New	Ctrl+N	Defines a new mailbox: see <i>Adding or Modifying a Mailbox Definition</i> in the <i>Administration</i> documentation.
Refresh	F5	See <i>Refreshing Object Lists</i> .
Filter	F3	See <i>Filtering Objects</i> .
Export	---	See <i>Exporting Objects</i> in the <i>Import/Export Functions</i> documentation.
Set Drag And Drop Function	--	See <i>Drag & Drop</i> .

➤ To list all available functions for a Mailbox Definition instance

- Select a **Mailbox Definition** instance and open the context menu.

The following functions are available:

Function	Shortcut	Description
Open	Ctrl+O	Modifies a mailbox definition: see <i>Adding or Modifying a Mailbox Definition</i> in the <i>Administration</i> documentation.
Display	Ctrl+D	Displays a mailbox definition see <i>Adding or Modifying a Mailbox Definition</i> in the <i>Administration</i> documentation.
Delete	Del	See <i>Deleting Objects</i> .
Export	---	See <i>Exporting Objects</i> in the <i>Import/Export Functions</i> documentation.
Show Messages	---	See <i>Viewing Mailbox Messages</i> .
Add to Workplan	---	See <i>Add to Workplan</i> .
Set Drag And Drop Function	--	See <i>Drag & Drop</i> .

Viewing Mailbox Messages

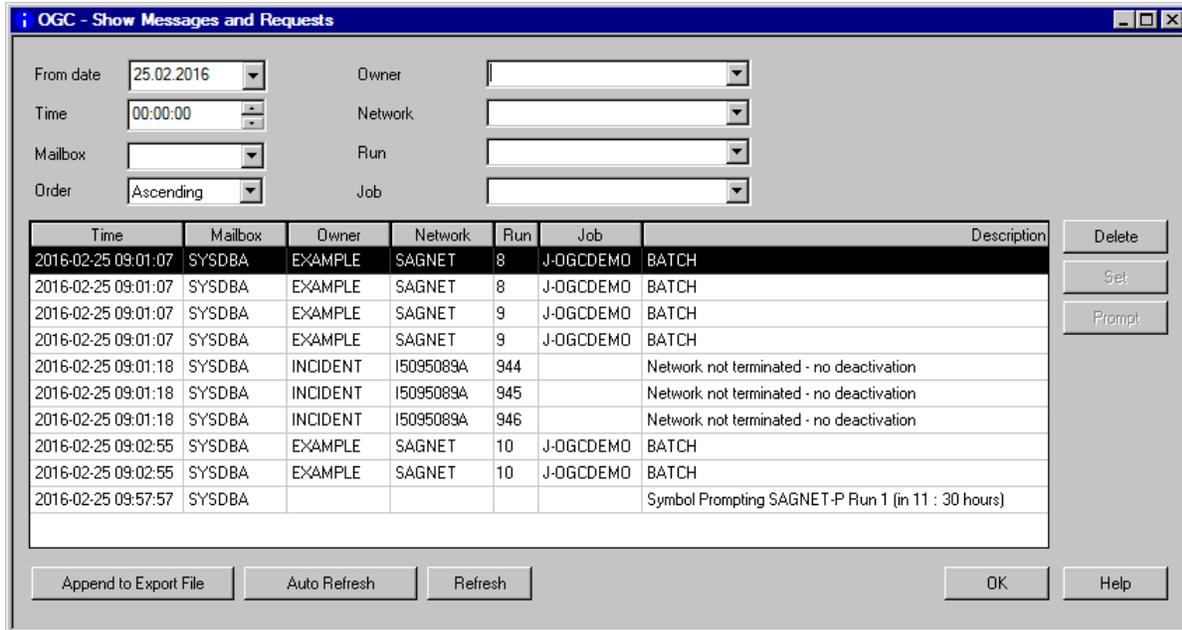
➤ To view the messages for a mailbox

- In the object workspace, select a **Mailbox Definition** instance and choose **Show Messages** from the context menu.

Or:

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

A **Show Messages and Requests** window similar to the example below opens:



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 Functions: Messages and Requests](#).

For the functions provided in the window, see [Available Functions: Show Messages and Requests](#).

Fields, Columns and Functions: Messages and Requests

This section describes the fields, columns and functions provided in the **Show Messages and Requests** window.

- [Fields and Columns: Messages and Requests](#)

- Available Functions: Show Messages and Requests

Fields and Columns: Messages and Requests

The following table explains the columns and fields of the [Show Messages and Requests window](#).



Note: Preselection settings of the **Show Messages and Requests** window can be saved within a session. For this purpose, check the **Save mailbox message list settings within a session** option in the user profile: see the **GUI General** page of the **Maintenance User** window described in *Defining GUI-Specific Attributes* in the *Administration* documentation.

Column/Field	Description			
From Date/Time	Input fields: Specify the start date and time for displaying the messages.			
	Date and time when the message was sent.			
Mailbox	Mailbox to which the message was sent.			
	Drop-down list box: Select the mailbox for which to list the messages. If there is no specification, all messages of all mailboxes from the user are displayed.			
Order	Sort order of the message list			
	Select either option from the drop-down list box:			
	<table border="1"> <tr> <td>Ascending</td> <td>Ascending order. Earliest messages come first.</td> </tr> <tr> <td>Descending</td> <td>Descending order. Latest messages come first.</td> </tr> </table>	Ascending	Ascending order. Earliest messages come first.	Descending
Ascending	Ascending order. Earliest messages come first.			
Descending	Descending order. Latest messages come first.			
Owner	Name of the owner.			
	Drop-down list box: Select the owner whose messages you want to list.			
Network	Name of the network.			
	Drop-down list box: Select the network for which to list the messages.			
Run	Number of the run.			
	Drop-down list box: Select the run number for which to show the messages.			
Job	Name of the job.			
	Drop-down list box: Select the run number for which to show the messages.			
Description	Message text: see Message Types for explanations.			

Available Functions: Show Messages and Requests

The following functions are provided in the **Show Messages and Requests** window:

Function	Description
Delete	Removes the selected message from the list. A dialog prompts you to confirm the deletion.
Set	Only available for jobs requesting a condition. Sets the selected condition to true. A dialog prompts you to confirm with YES (true) or reject with NO (false).
Prompt	Only available for networks or jobs requesting symbol prompting. Opens a Symbol Prompting window where you can view, add, modify or deactivate the symbol(s) defined for the network indicated in the message. If required, change the value assigned to the symbol in the Value column. ADD and MODIFY open a Symbol Active window where you can add a new symbol to the specified symbol table or modify an existing one. See also the section <i>Maintaining a Symbol Active</i> . Continue saves any symbol changes and closes the Symbol Prompting window. Back closes the Symbol Prompting window without any deactivate or activate action. Deactivate deactivates the active run of the specified network. A dialog prompts you to confirm the deactivation.
Append to Export File	Opens the Export Objects window to export the message list. Proceed as described in <i>Exporting Objects</i> in the <i>Import/Export Functions</i> documentation.
Auto Refresh	See <i>Refreshing Object Lists</i> .
Refresh	

Message Types

The types of messages that can be shown in the **Show Messages and Requests** window 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.

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.

Information Messages

```
NETWORK DEMO-NET ACTIVATED ==> SN / DEMO-NET / 195 /  
MESSAGE
```

This is an information message.

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 *Defaults for Time Ranges* in the *Administration* documentation.

XIII

Log Information

57

Log Information

- Displaying Logged Information - Browse Log Function 576
- Monitoring Entire Operations Activities 580

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 - Browse Log Function

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 log information**

- 1 For a single node item:

From the object workspace, select the object whose log you want to see and choose **Browse Log** (if applicable) from the context menu.

Or:

For a selected owner and network:

From the object workspace, select the **General** node and choose **Browse Log**.

A **Log Display Selection** window similar to the example below opens:

OGC - Log Display Selection

Date/Time from: 14.09.2015 00:00:00

Date/Time to: 14.09.2015 10:58:31

Max. lines: 0

Object selection:

Owner: [dropdown]

Network: [dropdown]

Run: [dropdown]

Job: [dropdown]

User selection:

User: [dropdown]

Automatically go to the end

OK Cancel Help

- 2 Specify the required selection criteria. The input fields are explained in [Fields: Log Display Selection](#).
- 3 Choose **OK**.

The logging information is shown in a **Browse Log** window similar to the example below:

Num	User ID	Owner	Network	Run	Job	Date	Time	Message
1	TASK 2	EXAMPLE	B60-FLOW	297		2015-08-03	11:30:29.1	Network to be
2	TASK 2	EXAMPLE	B60-FLOW	298		2015-08-05	10:00:05.0	Network to be
3	TASK 2	EXAMPLE	B60-FLOW	299		2015-08-10	10:00:00.0	Network to be
4	TASK 2	EXAMPLE	B60-FLOW	300		2015-08-12	10:00:00.0	Network to be
5	TASK 2	EXAMPLE	B60-FLOW	305		2015-08-31	10:47:29.0	Network to be
6	TASK 2	EXAMPLE	B60-FLOW	305		2015-09-04	07:55:19.9	Network not t
7	TASK 2	EXAMPLE	B60-FLOW	305		2015-09-04	21:09:37.7	Activation re
8	TASK 2	EXAMPLE	B60-FLOW	305	JOB-01	2015-09-04	21:09:37.7	Job Deactivat
9	TASK 2	EXAMPLE	B60-FLOW	305	JOB-012	2015-09-04	21:09:37.7	Job Deactivat
10	TASK 2	EXAMPLE	B60-FLOW	305	JOB-03	2015-09-04	21:09:37.7	Job Deactivat
11	TASK 2	EXAMPLE	B60-FLOW	305	JOB-013	2015-09-04	21:09:37.7	Job Deactivat
12	TASK 2	EXAMPLE	B60-FLOW	305	JOB-014	2015-09-04	21:09:37.7	Job Deactivat
13	TASK 2	EXAMPLE	B60-FLOW	305	JOB-015	2015-09-04	21:09:37.7	Job Deactivat
14	TASK 2	EXAMPLE	B60-FLOW	305	JOB-019	2015-09-04	21:09:37.7	Job Deactivat
15	TASK 2	EXAMPLE	B60-FLOW	305	JOB-02	2015-09-04	21:09:37.7	Job Deactivat
16	TASK 2	EXAMPLE	B60-FLOW	305	JOB-04	2015-09-04	21:09:37.7	Job Deactivat
17	TASK 2	EXAMPLE	B60-FLOW	305	JOB-05	2015-09-04	21:09:37.7	Job Deactivat
18	TASK 2	EXAMPLE	B60-FLOW	305	JOB-06	2015-09-04	21:09:37.7	Job Deactivat
19	TASK 2	EXAMPLE	B60-FLOW	305	JOB-1-TEST	2015-09-04	21:09:37.8	Job Deactivat
20	TASK 3	EXAMPLE	B60-FLOW	305	JOB-01	2015-09-04	21:09:40.4	Job deactivat

For explanations of the columns, see [Column Headings: Browse Log](#). The functions of the context menu are described in [Available Functions: Browse Log](#).

- [Fields: Log Display Selection](#)

- [Column Headings: Browse Log](#)
- [Available Functions: Browse Log](#)
- [Extended Log](#)

Fields: Log Display Selection

The **Log Display Selection** window provides the following input fields:

Field	Description
Date/Time from	The date and time (or a start date or time) for which the log entries are selected. Default is the current date and 00:00:00 for time in the format HH:MM:SS.
Date/Time to	The end date/time for the date/time specified in the Date/Time from fields. Default is the current date and time in the format HH:MM:SS.
Max. Lines	The maximum number of lines in the log display. 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. Leave blank to select log of all owners.
Network	Enter name of the network whose log is to be displayed. Leave blank to select log of all networks.
Run	Enter a run number or leave blank to select all numbers. 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. Leave blank to select log of all jobs.
User	Enter the user ID for which the log is to be displayed or leave blank to select all user IDs. To display the log for the Monitor, enter the name of the Monitor task.
Automatically go to end	Scrolls down to the end of the log report to see the most recent log entries. 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.

Column Headings: Browse Log

The **Browse Log** window contains the following columns:

Column	Description
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.

Column	Description
Date	Date of log entry.
Time	Time of log entry.
Message	<p>Message text.</p> <p>Depending on the default display option set (see <i>List Display Options</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.</p>

Available Functions: Browse Log

The context menu of the **Browse Log** window provides the following functions:

Function	Shortcut	Description
Copy	Ctrl+C	Copies the currently selected text. In case some text from the selection is not received from a server, a warning is shown and the Copy function is not performed.
Select All	Ctrl+A	Selects all text.
Reverse Order	Ctrl+R	Shows text in the reverse order.
Find	Ctrl+F	Finds the text fragment.
Find next	F3	Finds the next text fragment.
Go To	Ctrl+G	Moves the cursor to the specified line number.
Print	Ctrl+P	Prints log data.
Refresh	F5	Refreshes the log with "time to"=actual time
Refresh with new start time	F4	Refreshes the log with "time from"=time from the selected row "time to"=actual time
Extended Log	Ctrl+E	Log items with extended log information (active JCL changes, JCL, SYSOUT) are marked with an asterisk (*) in the User ID column. See also Extended Log .

Extended Log

An asterisk (*) in the **User ID** 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

- In the **User ID** column of the **Browse Log** window, place the cursor in an item that begins with an asterisk (*) and choose **Extended Log** from the context menu, or press **Ctrl+E**.

The extended log appears.

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 *Defaults: System/Log Files (Administration documentation)*, the changes are visible in detail in the extended logging screen.

Monitoring Entire Operations Activities

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.
 - [Viewing Entire Operations Activities](#)
 - [Columns: Activity Log](#)

Viewing Entire Operations Activities

➤ To start monitoring and display activities

- 1 In the object workspace, select the **General** metanode.
- 2 Open the context menu and choose **Activity Log**.

An **Activity Log** window similar to the example below opens:

Owner	Network	Job	Run	Job ID	Time	Message
EXAMPLE	B60-FLOW	JOB-013	380		18:28:29	Activation in Progress - Job type JOB
EXAMPLE	B60-FLOW	JOB-014	380		18:28:29	Activation in Progress - Job type JOB
EXAMPLE	B60-FLOW	JOB-015	380		18:28:29	Activation in Progress - Job type DUM
EXAMPLE	B60-FLOW	JOB-019	380		18:28:29	Activation in Progress - Job type JOB
EXAMPLE	B60-FLOW	JOB-02	380		18:28:29	Activation in Progress - Job type JOB
EXAMPLE	B60-FLOW	JOB-03	380		18:28:29	Activation in Progress - Job type NAT
EXAMPLE	B60-FLOW	JOB-04	380		18:28:29	Activation in Progress - Job type JOB
EXAMPLE	B60-FLOW	JOB-05	380		18:28:29	Activation in Progress - Job type DUM
EXAMPLE	B60-FLOW	JOB-06	380		18:28:29	Activation in Progress - Job type JOB
EXAMPLE	B60-FLOW	JOB-1-TEST	380		18:28:29	Activation in Progress - Job type JOB
EXAMPLE	B60-FLOW	JOB-03	380		18:28:32	NAT Module SYSEORU/B60-P01 not found
EXAMPLE	B60-FLOW	JOB-03	380		18:28:32	JCL Load Error occurred

All current activities within Entire Operations are logged and displayed in this window. Error messages are highlighted in red.

The columns contained on the screen are described in [Columns: Activity Log](#).

- 3 You can use the **Auto Refresh** and **Refresh** options to update the display and modify the update interval. See also [Refreshing Object Lists](#).

You can use the **Auto scrolling** option to automatically scroll to the latest activity.

Columns: Activity Log

The columns in the **Activity Log window** are described in the following table:

Column	Meaning
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.
Job ID	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. For a list of possible messages, see <i>Messages in Active Job Display</i> in the <i>Messages and Codes</i> documentation.

XIV

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

58

Purpose and Use of Symbol Tables and Symbols

▪ Symbol Tables	586
▪ Symbols	587
▪ Subnetworks and Recovery Jobs	589
▪ User Exits for User-specific Symbol Maintenance Tasks	589

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 [Message and Message Recipients - 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*

59

Adding and Maintaining Symbol Tables

▪ Available Functions: Symbol Table Master	592
▪ Listing Symbol Table Masters	593
▪ Displaying a Symbol Table Master	594
▪ Adding a Symbol Table Master	595
▪ Modifying a Symbol Table Master	598
▪ Maintaining the Usage of Symbol Table Versions	601
▪ Saving Symbol Tables as Files	604
▪ Using Active Symbol Tables	605
▪ Deleting a Symbol Table Master	608

Available Functions: Symbol Table Master

➤ To list all functions available for a Symbol Table Master node

- In the object workspace, select the **Symbol Table Master** metanode from an **Owner** instance and open the context menu.

The following functions are available:

Function	Shortcut	Description
List	F8	See <i>Listing Objects</i> .
New	CTRL+N	See <i>Adding a Symbol Table Master</i> .
Refresh	F5	See <i>Refreshing Object Lists</i> .
Filter	F3	See <i>Filtering Objects</i> .
Paste data	CTRL+V	See <i>Pasting Objects</i> .
Export	---	See <i>Exporting Objects</i> in the <i>Import/Export Functions</i> documentation.
Set Drag And Drop Function	---	See <i>Drag & Drop</i> .

➤ To list all functions available for a symbol table master

- From the **Symbol Table Master** node, select an instance and open the context menu.

The following functions are available:

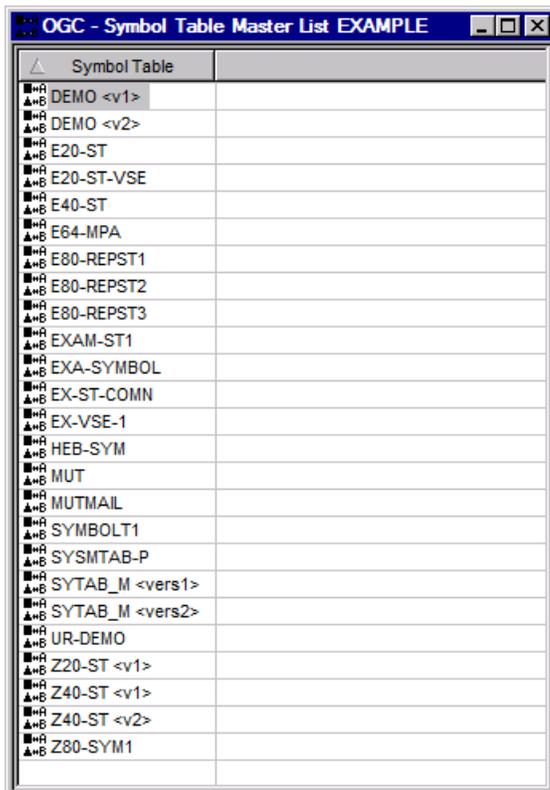
Function	Shortcut	Description
Open	CTRL+O	See <i>Modifying a Symbol Table Master</i> .
Display	CTRL+D	See <i>Displaying a Symbol Table Master</i> .
Where used	---	See <i>Listing Jobs and Networks that use a Symbol Table Master</i> .
Delete	DELETE	See <i>Deleting a Symbol Table Master</i> .
Copy data	CTRL+C	See <i>Copying Objects</i> .
Version Usage	---	See <i>Maintaining the Usage of Symbol Table Versions</i> .
Export	---	See <i>Exporting Objects</i> in the <i>Import/Export Functions</i> documentation.
Add to Workplan	---	See <i>Add to Workplan</i> .
Save as File	---	See <i>Saving Symbol Tables as Files</i> .
Set Drag And Drop Function	---	See <i>Drag & Drop</i> .

Listing Symbol Table Masters

➤ To list all master symbol tables of an owner

- In the object workspace, select the **Symbol Table Master** node of an owner and choose **List** from the context menu, or press F8.

A **Symbol Table Master List** window similar to the example below opens:



Symbol Table	
DEMO <v1>	
DEMO <v2>	
E20-ST	
E20-ST-VSE	
E40-ST	
E64-MPA	
E80-REPST1	
E80-REPST2	
E80-REPST3	
EXAM-ST1	
EXA-SYMBOL	
EX-ST-COMN	
EX-VSE-1	
HEB-SYM	
MUT	
MUTMAIL	
SYMBOLT1	
SYSMTAB-P	
SYTAB_M <vers1>	
SYTAB_M <vers2>	
UR-DEMO	
Z20-ST <v1>	
Z40-ST <v1>	
Z40-ST <v2>	
Z80-SYM1	

This window lists master symbol tables already defined to Entire Operations. You see only the symbol tables that belong to your owner (here: EXAMPLE). If no master symbol tables have been defined for your owner, the list is empty.

The functions available from the context menu are explained in [Available Functions: Symbol Table Master List](#).

This section covers the following:

- Available Functions: Symbol Table Master List

Available Functions: Symbol Table Master List

When you select a symbol table from the **Symbol Table Master List** window, you can open a context menu and perform the following functions:

Function	Shortcut	Description
Open	CTRL+O	See Modifying a Symbol Table Master .
Display	CTRL+D	See Displaying a Symbol Table Master .
New	CTRL+N	See Adding a Symbol Table Master .
Refresh	F5	See Refreshing Object Lists .
Where used	---	See Listing Jobs and Networks that use a Symbol Table Master .
Delete	DELETE	See Deleting a Symbol Table Master .
Copy data	CTRL+C	See Copying Objects - Copy Data Function .
Paste data	CTRL+V	See Pasting Objects .
Version Usage	---	See Maintaining the Usage of Symbol Table Versions .
Export	---	See Exporting Objects in the <i>Import/Export Functions</i> documentation.
Add to Workplan	---	See Add to Workplan .
Save as File	---	See Saving Symbol Tables as Files .

Displaying a Symbol Table Master

» To display a symbol table master

- 1 In the object workspace, select a **Symbol Table Master** instance.
- 2 Choose **Display** from the context menu, or press CTRL+D.

A **Display Symbol Table Master** window similar to the **Maintenance Symbol Table Master** window opens.

The window contains a table with a list of all symbols defined for the symbol table master.

The fields and table columns contained in the window are explained in [Fields and Columns: Symbol Table Master](#).

- 3 If you want to view additional information on a listed symbol, select the required symbol from the table and choose **Display**.

A **Display Symbol Master** window similar to the **Maintenance Symbol Master** window opens.

- 4 You can view all symbol definitions on the tabbed pages of the window. The fields on the pages are described in *Fields: Master and Active Symbols*.

Adding a Symbol Table Master

You can add a symbol table master by either copying an existing table (see the **Copy data/Paste data** functions of a symbol table instance/master node) or creating a new symbol table with the **New** function. This requires that you define at least one symbol master along with the table.

This section provides instructions for using the **New** function.

» To add a symbol table master

- 1 In the object workspace, select the **Symbol Table Master** metanode.
- 2 From the context menu choose **New**, or press CTRL+N.

A **Create new Symbol Table Master** window opens.

The fields and columns contained in the window correspond to the field and columns in the **Maintenance Symbol Table Master** window. They are explained in *Fields and Columns: Symbol Table Master*.

- 3 Enter a name for the symbol table and symbol table version and choose **Add**.

A **Create new Symbol Master** window opens.

The fields and tabbed pages contained in the window correspond to the fields and tabbed pages in the **Maintenance Symbol Master** window. They are explained in *Fields: Master and Active Symbols*.

- 4 If required, enter a different symbol table name and version and make your definitions on the tabbed pages **Main**, **Multiple Symbol Values** and **Value Check**.
- 5 Choose **OK** when you are finished.

The symbol table master is created and now ready to use in the active network.

This section covers the following topics:

■ [Fields: Master and Active Symbols](#)

Fields: Master and Active Symbols

The fields on the tabbed pages **Main**, **Multiple Symbol Values** and **Value Check** (symbol masters only) of a **Symbol Master** or **Symbol Active** window are explained 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.

Main Page:		
Symbol	Name of the symbol to be used as a variable.	
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 *** hidden ***.
	L	Alphanumeric; conversion to lower case
	N	Numeric See Value for the numeric field format.
	U	Alphanumeric; conversion to upper case
Value	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. Example: 'Fixed Length ' Note: <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. See also Defining Multiple Symbol Values and Specifying a Range Check for Numeric Symbol Values .	

<p>Prompting (list box)</p>	<p>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.</p> <p>Possible selection options:</p> <table border="1" data-bbox="527 367 1482 537"> <tr> <td data-bbox="527 367 971 415">A</td> <td data-bbox="971 367 1482 415">Prompt for symbol at each activation.</td> </tr> <tr> <td data-bbox="527 415 971 493">E</td> <td data-bbox="971 415 1482 493">Prompt only if no value is specified in the table.</td> </tr> <tr> <td data-bbox="527 493 971 537">N</td> <td data-bbox="971 493 1482 537">Never prompt for symbol.</td> </tr> </table>	A	Prompt for symbol at each activation.	E	Prompt only if no value is specified in the table.	N	Never prompt for symbol.
A	Prompt for symbol at each activation.						
E	Prompt only if no value is specified in the table.						
N	Never prompt for symbol.						
<p>Prompting (check box)</p>	<p>Write back modified active symbol to the master table as well.</p> <p>Note: The symbol is also written back if a respective global setting is available. See also <i>Entire Operations Defaults</i> in the section <i>System Administrator Services</i> of the <i>Entire Operations Administration Documentation</i>.</p>						
<p>Prompt Text</p>	<p>Optional text to be displayed when symbol prompting is performed, to help the user specify the correct value.</p> <p>See also Symbol Prompting during Network or Job Activation.</p>						
<p>Multiple Symbol Values Page:</p>							
<p>Values 1 to 100</p>	<p>Assign multiple values to a symbol.</p> <p>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.</p> <p>You can enter up to 100 values here. The maximum (internal) value length is 80. These values are used for subsequent substitutions.</p> <p>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.</p>						
<p>Value Check Page:</p>							
<p>Not applicable to active symbols</p> <p>Used to define a range of numbers to be checked for valid input of either numeric symbol values or a user exit. You cannot use both options.</p> <p>See also Specifying a Range Check for Numeric Symbol Values and User Exit for Validating Symbol Values.</p>							
<p>None</p>	<p>Select this button (default) if you neither use Range check nor User exit.</p>						
<p>Range check</p>	<p>Select this button if you want to define and/or activate a range of numbers to be checked during input of numeric symbol values.</p>						
<p>Value from</p>	<p>Start (from) and end (to) number of the range of numbers to be checked.</p>						
<p>Value to</p>	<p>Valid input values: 1 to 999.</p>						
<p>User exit</p>	<p>Select this button if you want to specify and/or activate a user exit.</p>						

Library	Name of the library in which the user exit resides 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 select a user exit from the drop-down list box.

Modifying a Symbol Table Master

Modifying a master symbol table or an active symbol table consists of adding, modifying or deleting symbols and their current values.

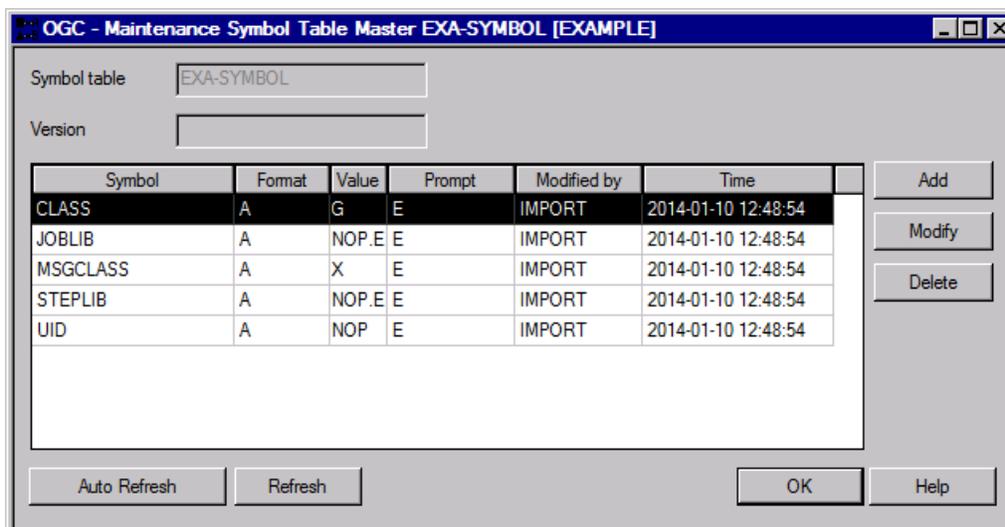
» To modify a symbol table master

- 1 In the object workspace, select a **Symbol Table Master** instance.
- 2 Invoke the context menu and choose the **Open** function.

Or:

Press CTRL+O.

A **Maintenance Symbol Table Master** window similar to the example below opens:



You can modify a selected symbol (here: CLASS) by choosing **Modify**, add a new symbol by choosing **Add**, or delete a selected symbol by choosing **Delete**.

The fields and columns contained in the window are explained in *Fields and Columns: Symbol Table Master*.

- 3 If you choose **Delete**, a window prompts you to delete the selected symbol.
- 4 If you choose **Modify** or **Add**, a **Maintenance Symbol Master** window similar to the example below opens:

- 5 Make the required definitions in the fields of the tabbed pages **Main**, **Multiple Symbol Values** and **Value Check**. They are explained in *Fields: Master and Active Symbols*.
- 6 Choose **OK** when you are finished.

The symbol table master is changed and now ready to use in the active network.

This section covers the following topics:

▪ Fields and Columns: Symbol Table Master

Fields and Columns: Symbol Table Master

The fields and columns in the **Create new Symbol Table Master**, **Display Symbol Table Master** or **Maintenance Symbol Table Master** window are explained in the following table:

Field/Column	Description	
Symbol	Name of the symbol used as a variable.	
Format	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
Prompt	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 column	No value is defined for the variable.
	*** <i>n</i> multiple values ***	One or more values (<i>n</i>) are defined for the variable. Select the Multiple Symbol Values tab to view or change the values.
Modified by	User who last modified the symbol.	
Time	Date and time when a user last modified the symbol.	

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](#)

Related Topic:

- *Object Versioning* in the *Concepts and Facilities* documentation

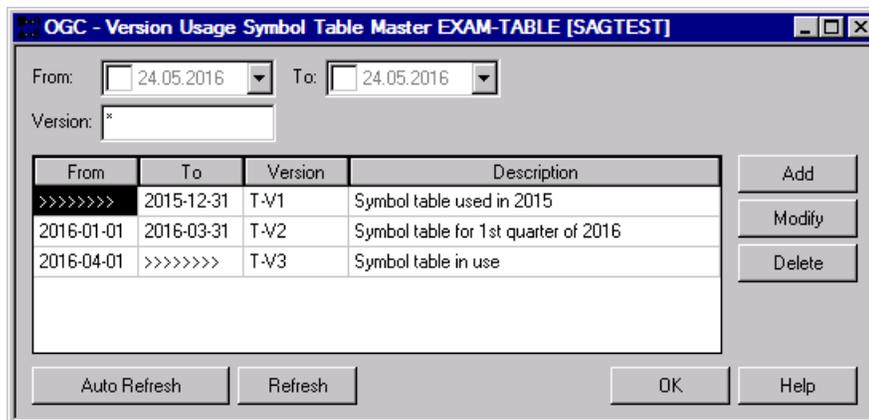
Listing, Defining and Deleting Version Usage

This section provides instructions for listing, defining and deleting version usage definitions for symbol table versions.

» To list version usage definitions

- 1 In the object workspace, select a symbol table master.
- 2 Open the context menu and choose **Version Usage**.

A **Version Usage Symbol Table Master** window similar to the example below opens:



The window lists the version usage defined for the selected symbol table master version (the table is empty if no definition exists).

If you enter an asterisk (*) in the **Version** field, all version usages defined for all versions of the symbol table master are listed in the window as shown in the example above. The fields and columns are explained in *Fields and Columns: Version Usage Definition*.

The following functions are available:

Function	Description
Add	See <i>To add or modify a version usage definition</i> .
Modify	Modifies the usage definition selected in the table row. See <i>To add or modify a version usage definition</i> .
Delete	Deletes the usage definition selected in the table row. A window prompts you to confirm the deletion with Yes or cancel the action with No .
Auto Refresh/Refresh	See <i>Refreshing Object Lists</i> .

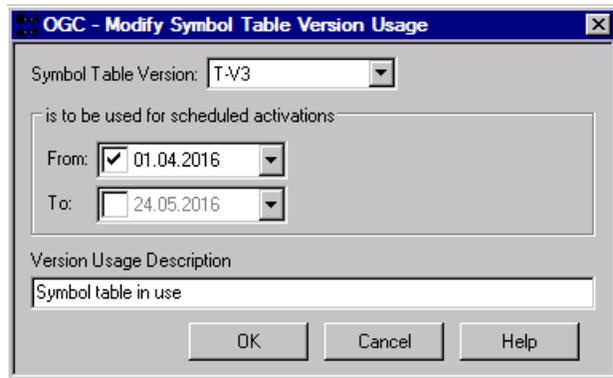
➤ **To add or modify a version usage definition**

- 1 In the **Version Usage Symbol Table Master window**, choose **Add** to create a new definition.

Or:

In the **Version Usage Symbol Table Master window**, select the table row that contains the definition you want to modify, and choose **Modify**.

The input fields that appear in the window are shown in the following example:



- 2 Enter the required values or select a value from the drop-down list boxes. Activate or deactivate a **From** or **To** field by selecting the check box next to it. Deactivated fields are ignored and not processed. For additional information, see *Fields and Columns: Version Usage Definition*.

3 Choose **OK** when you are finished.

Fields and Columns: Version Usage Definition

Field/Column	Description
<p>From (column or input field)</p>	<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 or no value (check box of field not selected) for no start date.</p> <p>You can open the drop-down list box to select a date from a calendar.</p> <p>Default: start date is not defined.</p> <p>Note: Date ranges must not overlap.</p>
<p>To (column or input field)</p>	<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> <p>Possible value for an input field:</p> <p>An end date or no date (check box of field not selected) for an unlimited (infinite) time.</p> <p>You can open the drop-down list box to select a date from a calendar.</p> <p>Note: Date ranges must not overlap.</p> <p>Default: end date is not defined.</p>
<p>Version (column)</p>	<p>Symbol table version for which version usage is defined.</p>
<p>Description (column)</p>	<p>Description of the defined version usage.</p>
<p>Symbol Table Version</p>	<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>Enter the required version or select a version from the the drop-down list box.</p>
<p>Version Usage Description</p>	<p>Input field for a description of the version usage.</p> <p>Maximum input: 70 characters.</p>

Saving Symbol Tables as Files

➤ To save a master symbol table as a file

- 1 In the object workspace, select a **Symbol Table Master** instance and choose **Save as File** from the context menu.
- 2 Specify the file format and type as described in *Report Output Options* and choose **OK**.

An output file similar to the example below is created:

Master Symbol Table

Owner: SAGTEST Symbol Table: EXAM-TABLE

Symbol	MV	Symbol Value	Modified by	Time
CLASS	1	22.2	NATQA5	2016-05-12 14:18:10
CLASS	2	10.2	NATQA5	2016-05-12 14:18:10
CLASS	3	7	NATQA5	2016-05-12 14:18:10
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		*** empty ***	NATQA5	2016-05-04 18:51:16

This output displays the selected symbol table master and all related information.

➤ To save an active symbol table as a file

- 1 In a **Symbol Prompting** window, select **Save active symbol table as file** and choose **Continue**.
- 2 Specify the file format and type as described in *Report Output Options* and choose **OK**.

An output file similar to the example below is created:

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

This output displays the active symbol table and all related information.

Using Active Symbol Tables

This section covers the following topics:

- [Available Functions: Symbol Table Active](#)
- [Modifying a Symbol Table Active](#)
- [Listing Active Jobs Using a Symbol Table Active](#)

Available Functions: Symbol Table Active

➤ To list all functions available for a symbol table active node

- In the object workspace, select a **Symbol Table Active** node and open the context menu.

The following functions are available:

Function	Shortcut	Description
List	F8	See Listing Objects .
Refresh	F5	See Refreshing Object Lists .
Filter	F3	See Filtering Objects .
Set Drag And Drop Function	---	See Drag & Drop .

➤ To list all functions available for a symbol table active

- In the object workspace, select an instance of a **Symbol Table Active** node and open the context menu.

The following functions are available:

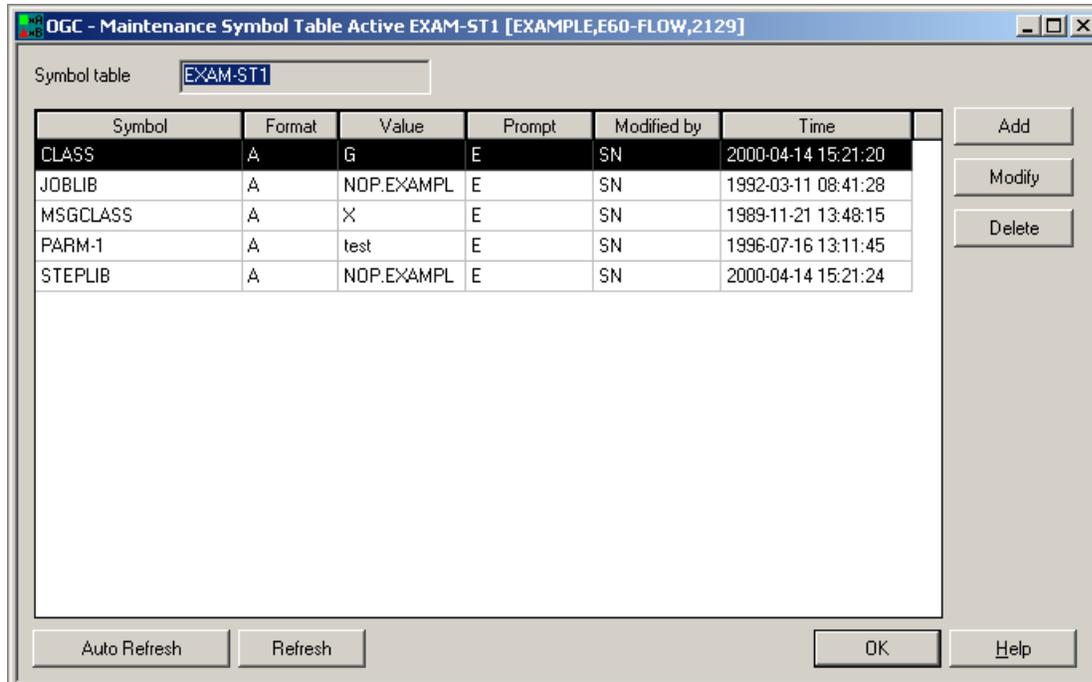
Function	Shortcut	Description
Open	CTRL+O	See Modifying a Symbol Table Active .
Display	CTRL+D	See Displaying Objects .
Active Usage	---	See Listing Active Jobs Using a Symbol Table Active .
Add to Workplan	---	See Add to Workplan .
Save as File	---	See Saving Symbol Tables as Files .

Modifying a Symbol Table Active

➤ To modify an active symbol table

- 1 In the object workspace, select a **Symbol Table Active** instance.
- 2 Open the context menu and choose **Open**, or press CTRL+O.

A **Maintenance Symbol Table Active** window similar to the example below opens:



Here, you can add, modify or delete a symbol of the symbol table without affecting the master symbol table definition.

The fields, columns and functions available in the window correspond to the fields, columns and functions of the **Maintenance Symbol Table Master** window. They are explained in the following sections:

- *Fields and Columns: Symbol Table Master,*
- *Adding a Symbol Table Master,*
- *Modifying a Symbol Table Master,* and
- *Deleting a Symbol Table Master.*

- 3 Make your definitions and choose **OK** when you are finished.

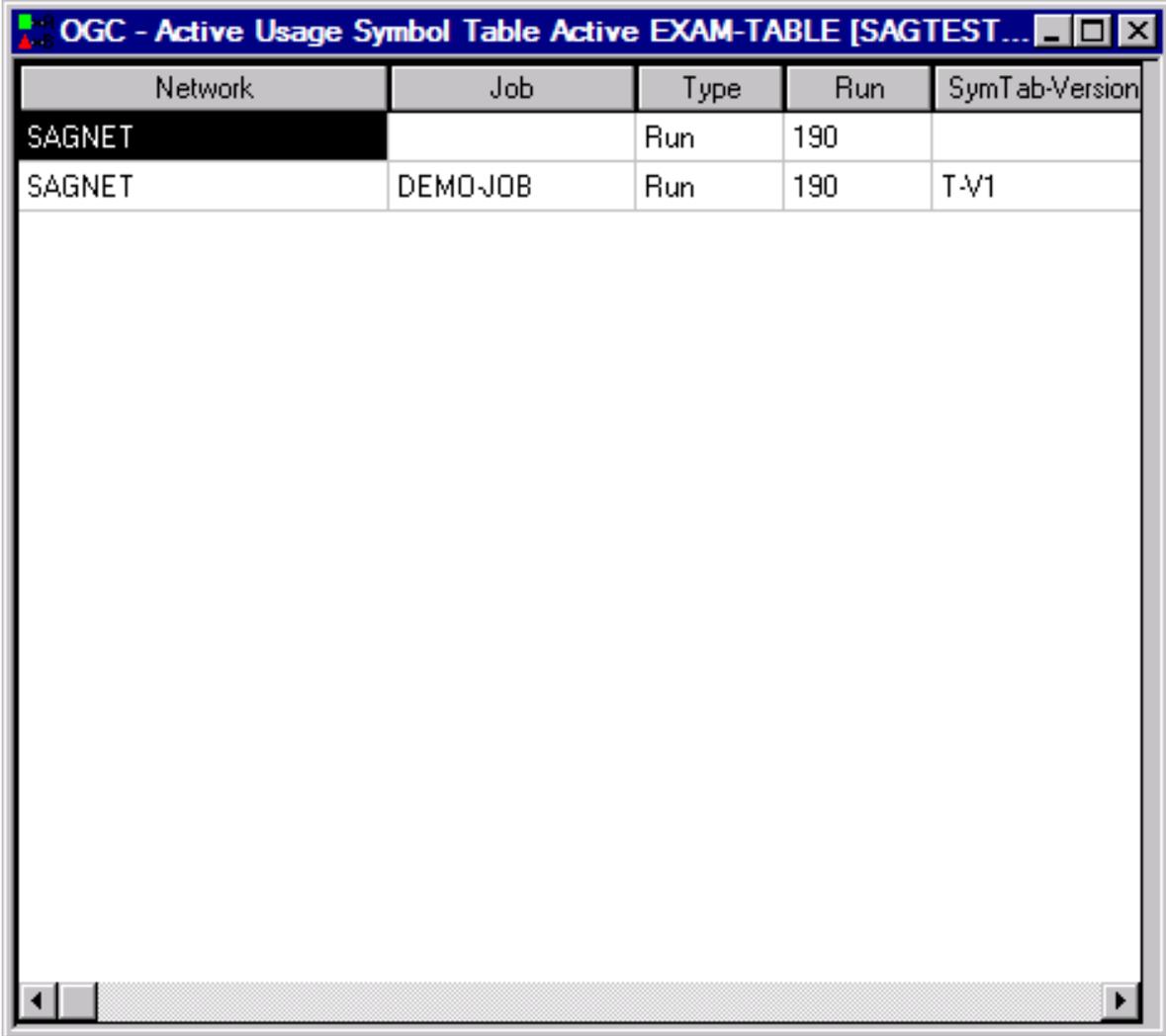
Listing Active Jobs Using a Symbol Table Active

This function is used to display all active jobs using an active symbol table.

➤ **To list all active jobs using an active symbol table**

- 1 In the object workspace, select a **Symbol Table Active** instance.
- 2 Open the context menu and choose **Active Usage**.

An **Active Usage Symbol Table Active** window similar to the example below opens:



Network	Job	Type	Run	SymTab-Version
SAGNET		Run	190	
SAGNET	DEMO-JOB	Run	190	T-V1

The window lists all active jobs used by the active symbol table. The columns contained in the window correspond to the columns in the [Where used Symbol Table Master](#). They are explained in [Columns: Symbol Table Usage](#)

Deleting a Symbol Table Master

Restrictions:

- You cannot delete a master symbol table that is still used by one or more jobs. In this case, the error message `EOR1859 - Symbol Table ... used in Definitions` occurs. You can only delete the symbol table if you either specify another symbol table for these jobs or delete them. Proceed as described in [Listing Jobs and Networks that use a Symbol Table Master](#) to find the jobs that use the symbol table.
- You cannot delete a master symbol table with a symbol table version that is still defined in the version usage of the symbol table. In this case, the error message `EOR2691 - Version ... - current usage found` occurs. You then need to delete the usage defined for the version as described in [Maintaining the Usage of Symbol Table Versions](#).

» To delete a symbol table master

- 1 In the object workspace, select a **Symbol Table Master** instance.
- 2 Choose **Delete** from the context menu, or press `DELETE`.

A confirmation window opens.

- 3 Choose **Yes** to confirm the deletion or **No** to cancel the action.

60 Finding Symbol Tables Associated with Jobs and Networks

- Listing Usable Symbol Tables 610
- Listing Jobs and Networks that use a Symbol Table Master 611

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](#)

Columns: Usable Symbol Tables

The columns contained in the **Usable Symbol Tables window** are described in the following table:

Column	Description
Type	Type of the symbol table: see <i>Symbol Table Types and Symbol Search Order</i> .
Owner	Owner of the symbol table.
Symbol 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.

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 of jobs and networks that use a symbol table master

- In the object workspace, select a **Symbol Table Master** instance and select **Where used** from the context menu.

A **Where used Symbol Table Master** window similar to the example below opens:

Network	Job	Type	Run	SymTab-Version
E40-REC-01	E40-J01	Def.		
E40-REC-01	E40-J02	Def.		
E40-REC-01	E40-J03	Def.		
E40-REC-02	E40-J01-RC	Def.		
V40-REC-01	E40-J01	Def.		
V40-REC-01	E40-J02	Def.		
V40-REC-01	E40-J03	Def.		
V40-REC-02	E40-J01-RC	Def.		
E40-REC-01		Def.		
E40-REC-02		Def.		
V40-REC-01		Def.		
V40-REC-02		Def.		

The window lists all networks and jobs that use the selected symbol table master (here: E40-ST of the owner EXAMPLE).

The columns in the window are explained in *Fields and Columns: Symbol Table Usage*.

➤ **To open a list of jobs and networks that use an active symbol table**

- In the object workspace, select a **Symbol Table Active** instance and choose **Active Usage** from the context menu.

An **Active Usage Symbol Table Active window** which corresponds to the **Where used Symbol Table Master** window opens. The fields and columns are explained in *Fields and Columns: Symbol Table Usage*.

This section covers the following topics:

- Columns: Symbol Table Usage

Columns: Symbol Table Usage

The columns contained in the **Where used Symbol Table Master** or **Active Usage Symbol Table Active window** are explained in the following table:

Column	Description				
Owner	Owner of the selected symbol table.				
Symbol Table	Name of the selected symbol table.				
Version (field)	Version of the selected symbol table. (all) is the default entry for all symbol table versions defined for the selected symbol table.				
Network	Name of the job network.				
Job	Name of the job that uses the symbol table. A dash (-) or a blank indicates that the symbol table is specified in the network definition.				
Run	If the symbol table is used by an active job, this field displays the job's run number.				
Type	Specifies whether job is active or not. Possible entries: <table border="1" data-bbox="462 982 1477 1108"> <tr> <td>Def.</td> <td>Symbol table is specified for job on the master database.</td> </tr> <tr> <td>Run</td> <td>Symbol table is specified for active job.</td> </tr> </table>	Def.	Symbol table is specified for job on the master database.	Run	Symbol table is specified for active job.
Def.	Symbol table is specified for job on the master database.				
Run	Symbol table is specified for active job.				
SymTab-Version	Version of the symbol table (if defined).				

61

Defining Symbols and Symbol Values

▪ Available Functions: Symbol Master	616
▪ Listing Symbol Masters	617
▪ Viewing a Symbol Master Definition	617
▪ Modifying a Symbol Master	618
▪ Adding a Symbol Master	619
▪ Maintaining a Symbol Active	619
▪ Defining Multiple Symbol Values	623
▪ Specifying a Range Check for Numeric Symbol Values	624
▪ Predefined Symbols	625
▪ Validating Symbol Values with a User Exit	633
▪ Global Symbol Modification Exit	633
▪ Symbol Setting triggered by the SYSOUT of a Job	633
▪ Deleting a Symbol from a Symbol Table	634

Available Functions: Symbol Master

➤ To list all functions available for a Symbol Master node

- In the object workspace, select a **Symbol Master** node and open the context menu.

The following functions are available:

Function	Shortcut	Description
List	F8	See Listing Symbol Masters .
New	CTRL+N	See Adding a Symbol Master .
Refresh	F5	See Refreshing Object Lists .
Filter	F3	See Filtering Objects .
Paste data	CTRL-V	See Pasting Objects .
Set Drag And Drop Function	---	See Drag & Drop .

➤ To list all functions available for a symbol master

- From the **Symbol Master** node, select an instance and open the context menu.

The following functions are available:

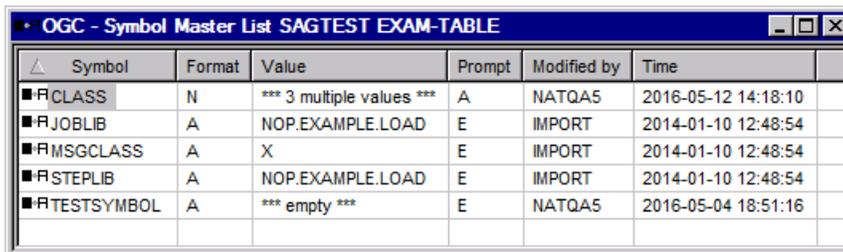
Function	Shortcut	Description
Open	CTRL+O	See Modifying a Symbol Master .
Display	CTRL+D	See Viewing a Symbol Master .
Delete	DELETE	See Deleting a Symbol from a Symbol Table .
Copy data	CTRL+C	See Copying Objects .
Set Drag And Drop Function	---	See Drag & Drop .

Listing Symbol Masters

➤ To list symbol masters

- In the object workspace, select a **Symbol Master** node and choose **List** from the context menu, or press F8.

A **Symbol Master List** window similar to the example below opens:



Symbol	Format	Value	Prompt	Modified by	Time
■ FCLASS	N	*** 3 multiple values ***	A	NATQA5	2016-05-12 14:18:10
■ FJOBLIB	A	NOP.EXAMPLE.LOAD	E	IMPORT	2014-01-10 12:48:54
■ FMSGCLASS	A	X	E	IMPORT	2014-01-10 12:48:54
■ FSTEPLIB	A	NOP.EXAMPLE.LOAD	E	IMPORT	2014-01-10 12:48:54
■ FTESTSYMBOL	A	*** empty ***	E	NATQA5	2016-05-04 18:51:16

The table contained in the window lists all symbol masters defined for the selected symbol table master (here: EXAM-TABLE).

The columns in the window are explained in [Fields and Columns: Symbol Table Master](#).

Viewing a Symbol Master Definition

➤ To view a symbol master definition

- 1 In the object workspace, select a **Symbol Master** instance.
- 2 Open the context menu and choose **Display**, or press CTRL+D.

A **Display Symbol Master** window opens with the current symbol definition.

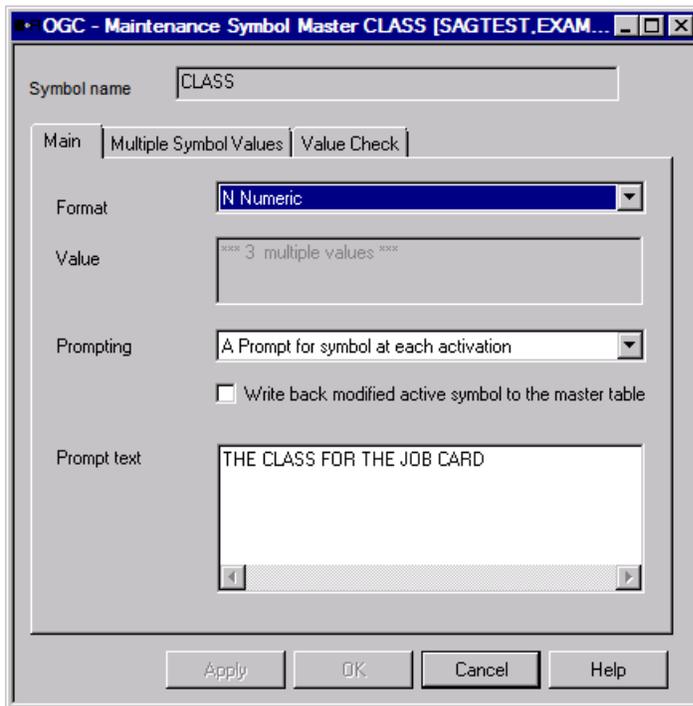
The read-only fields on the tabbed pages **Main**, **Multiple Symbol Values** and **Value Check** correspond to the fields and pages in the [Maintenance Symbol Master window](#). They are explained in [Fields: Master and Active Symbols](#).

Modifying a Symbol Master

➤ To modify a symbol master

- 1 In the object workspace, select a **Symbol Master** instance.
- 2 Open the context menu and choose the **Open** function, or press CTRL+O.

A **Maintenance Symbol Master** window similar to the example below opens:



- 3 Make your changes. The fields on the tabbed pages **Main**, **Multiple Symbol Values** and **Value Check** are explained in *Fields: Master and Active Symbols*.
- 4 Choose **OK** when you are finished.

The symbol master is modified, stored and now ready to use in the active network.

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 master to a symbol table

- 1 In the object workspace, select the **Symbol Master** node.
- 2 From the context menu, choose **New**, or press CTRL+N.

A **Create new Symbol Master** window opens.

The input fields on the tabbed pages **Main**, **Multiple Symbol Values** and **Value Check** correspond to the fields and pages in the **Maintenance Symbol Master window**. They are explained in *Fields: Master and Active Symbols*.

- 3 Enter a name for the symbol and make the required definitions.
- 4 Choose **OK** when you are finished.

The symbol master is created and now ready to use in the active network.

Maintaining a Symbol Active

This section covers the following topics:

- [Available Functions: Symbol Active](#)
- [Viewing a Symbol Active](#)
- [Adding a Symbol Active](#)
- [Modifying a Symbol Active](#)

Available Functions: Symbol Active

➤ To list all functions available for a Symbol Active node

- In the object workspace, select a **Symbol Active** node and open the context menu.

The following functions are available:

Function	Shortcut	Description
List	F8	See <i>Listing Objects</i> .
New	CTRL+N	See <i>Adding a Symbol Active</i> .
Refresh	F5	See <i>Refreshing Object Lists</i> .
Filter	F3	See <i>Filtering Objects</i> .
Paste data	CTRL+V	See <i>Pasting Objects</i> .
Set Drag And Drop Function	---	See <i>Drag & Drop</i> .

➤ To list all functions available for a single symbol active

- Select a **Symbol Active** instance and open the context menu.

The following functions are available:

Function	Shortcut	Description
Open	CTRL+O	See <i>Modifying a Symbol Active</i> .
Display	CTRL+D	See <i>Viewing a Symbol Active</i> .
Delete	DELETE	See <i>Deleting a Symbol from a Symbol Table</i> .
Copy data	CTRL+C	See <i>Copying Objects</i> .
Set Drag And Drop Function	---	See <i>Drag & Drop</i> .

Viewing a Symbol Active

➤ To view a symbol active

- In the object workspace, select a **Symbol Active** instance and choose **Display** from the context menu, or press CTRL+D.

A **Display Symbol Active** window similar to the **Maintenance Symbol Active** window opens.

You can view all symbol definitions on the tabbed pages **Main** and **Multiple Symbol Values**.

The fields and columns contained on the pages correspond to the field and columns of the **Maintenance Symbol Active** window. They are explained in *Fields: Master and Active Symbols*.

Adding a Symbol Active

➤ To add a symbol active

- 1 In the object workspace, select a **Symbol Active** node.
- 2 Open the context menu and choose **New**, or press CTRL+N.

A **Create new Symbol Active** window opens.

The fields on the tabbed pages **Main** and **Multiple Symbol Values** of the window correspond to the fields of the **Maintenance Symbol Active window**. They are explained in *Fields: Master and Active Symbols*.

Fill in the fields as required.

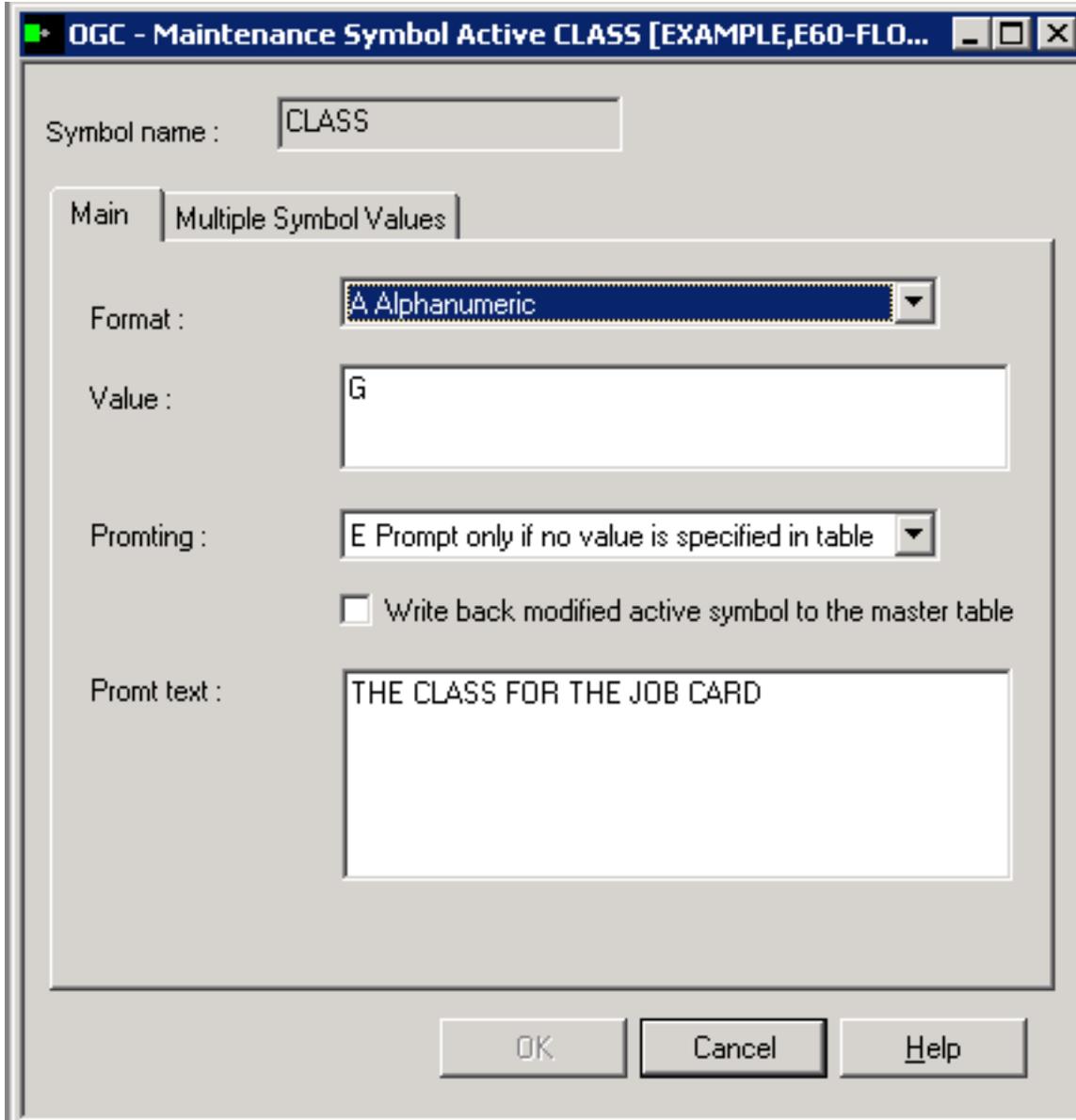
- 3 When you are finished, choose **OK**.

Modifying a Symbol Active

➤ To modify a symbol active

- 1 In the object workspace, select a **Symbol Active** instance and choose **Open** from the context menu, or press CTRL+O.

A **Maintenance Symbol Active** window similar to the example below opens:



- 2 Make your modifications. The fields on the tabbed pages **Main** and **Multiple Symbol Values** are explained in *Fields: Master and Active Symbols*.
- 3 When you are finished, choose **OK**.

The symbol active is modified and saved.

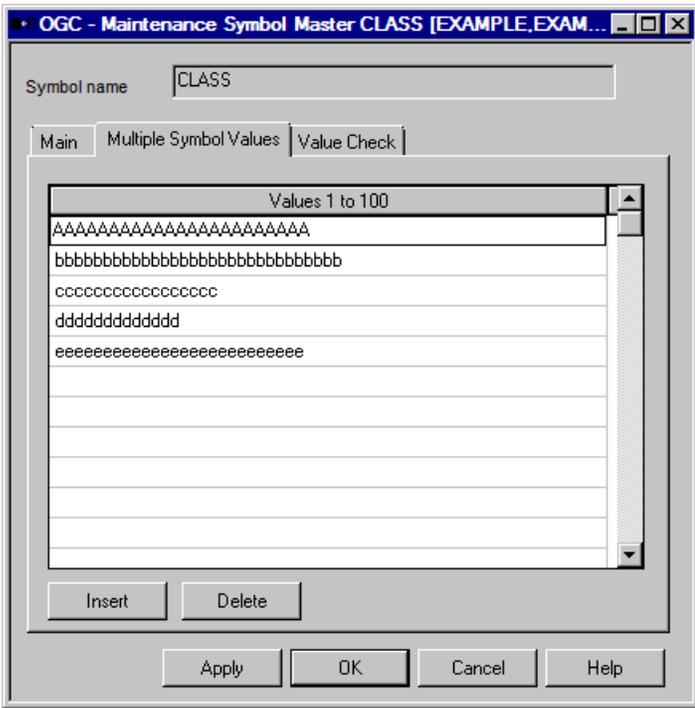
Defining Multiple Symbol Values

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 In a **Symbol Master** or **Symbol Active** window, open the tabbed page **Multiple Symbol Values**.

A **Multiple Symbol Values** page similar to the example below opens:



All values currently defined for the selected symbol are listed in a table. The table is empty if no values exist.

- 2 You can add a new value in an empty table row at the end of the table. If you want to insert an empty row, select the row above which you want to place it and choose **Insert**.

 **Note:** Incorrect value definitions (wrong format/length) or values that are not within a specified value range appear in red. For valid input values, see the description of the **Value** field and the **Value Check** page in the section *Fields: Master and Active Symbols*.

If you want to delete an entry, select the appropriate row and choose **Delete**.

This section covers the following topics:

- [Symbol Replacement](#)

Symbol Replacement

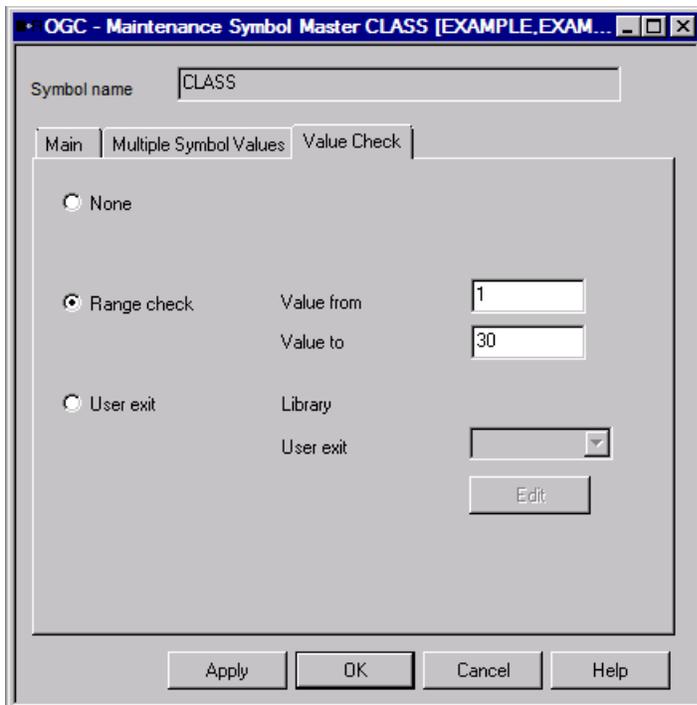
You can use the [symbol function !MV or ?MV](#) (see *Functions for Symbol Replacement*) to get single occurrences of multiple-value symbols for replacement.

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

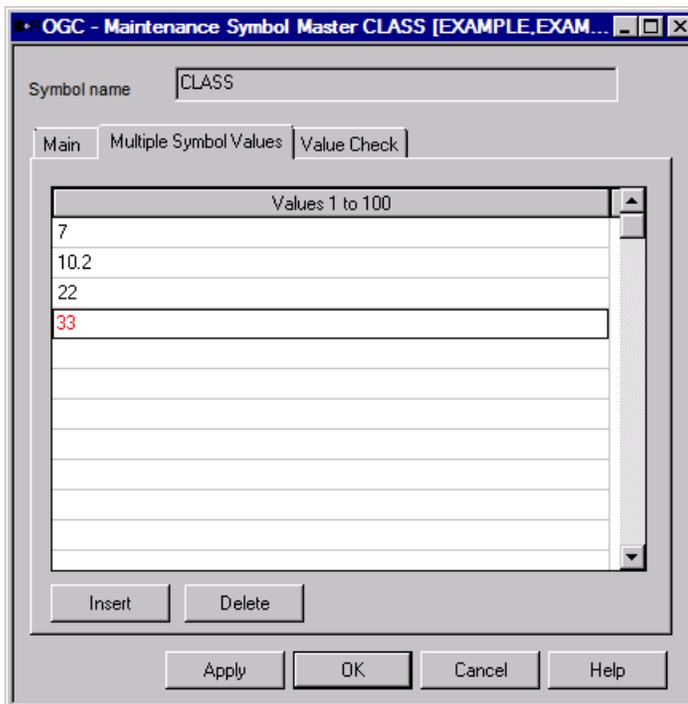
- 1 Open the **Value Check** page of a **Symbol Master** window.
- 2 Select the **Range check** button (default is **None** for no check) and enter a start number in the **Value from** field and an end number in the **Value to** field as shown in the following example of a **Value Check** page:



- 3 Choose **OK** to save your entries.

You can deactivate the check by selecting the **None** button.

- 4 If you open the **Multiple Symbol Values** page and add a new value that is not within the specified range (here: 1 to 30) the new (incorrect) value (here: 33) appears in red:



For valid input values, see the description of the **Value** field, the **Value Check** page and the **Multiple Symbol Values** page in the section *Fields: Master and Active Symbols*.

Predefined Symbols

Entire Operations provides a large number of predefined symbols that are reserved for system variables or particular system tasks. We strongly recommend not to use any of these symbols for user-defined purposes.

Predefined Symbol Ranges

All symbol names starting with an asterisk (*) and P- are reserved for predefined read-only symbols. These prefixes must not be used for user-defined symbols.

- [Predefined Symbols for Command Line Parameters](#)
- [Predefined Symbols for Mainframe Platforms](#)
- [Predefined Symbols for FTP Jobs](#)
- [Table of Predefined Symbols](#)
- [Predefined Symbols for Multiple and Parallel Activations](#)

- [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 Open the **Value Check** page of a **Symbol Master** window.
- 2 Select the **User exit** button (default is **None** for no check) and enter the name of a library and a user exit in the fields **Library** and **User exit**, respectively. The input fields are described in *Fields: Master and Active Symbols*.

Choose **Edit** if you want to modify the source of the user exit. Remember that you need to recatalog (STOW command) the source after modification.

- 3 Choose **OK** to save your entries.
- 4 Select the **None** button if you can deactivate the user exit.

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 master

- 1 In the object workspace, select a **Symbol Master** instance.

Choose **Delete** from the context menu, or press `DELETE`.

Or:

In a **Maintenance Symbol Table Master window**, select the symbol you want to delete and choose **Delete**.

A confirmation window opens.

- 2 Choose **Yes** to confirm the deletion or **No** to cancel the action.



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.

➤ To delete a symbol active

- 1 In the object workspace, select a **Symbol Active** instance.

Choose **Delete** from the context menu, or press `DELETE`.

Or:

In a **Maintenance Symbol Table Active window**, select the symbol you want to delete and choose **Delete**.

A confirmation window opens.

- 2 Choose **Yes** to confirm the deletion or **No** to cancel the action.

62 Symbol Replacement

- Symbol Escape Characters 638
- Symbol Prompting during Network or Job Activation 640
- Specifying User Exits for Symbol Modification 642
- Rules and Restrictions for Symbol Replacement 645

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 *Defaults for Network Options* 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 **Network Options** page of the **Entire Operations Defaults** function (see the *Administration* documentation) is used to define the default for all networks in your Entire Operations environment.

On the **Operating System Specials** page, you can also define the default escape characters for each operating system.

■ 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.



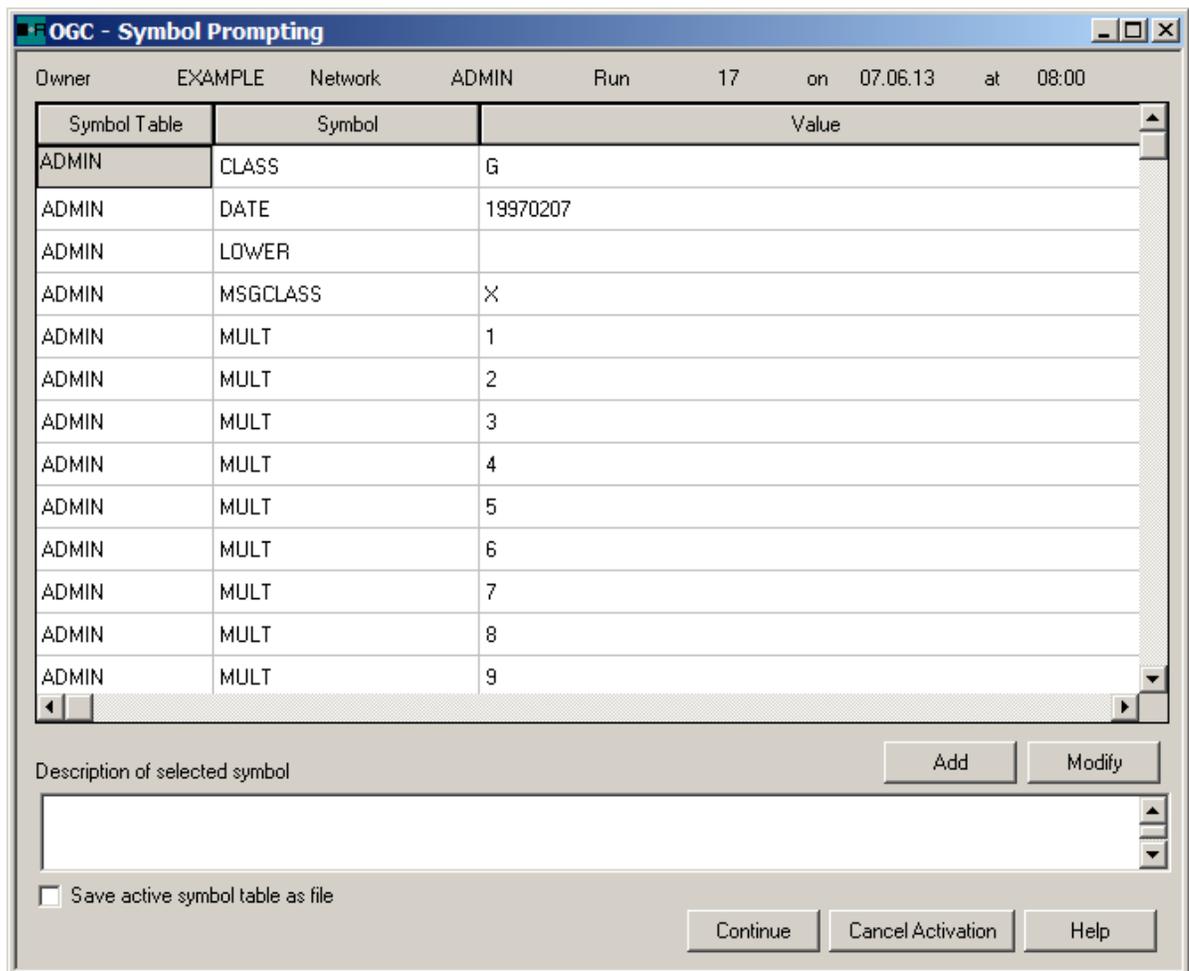
Note: Up to 100 different symbol tables can be prompted during a network activation.

If the activation is manual and if the JCL of a job contains symbols that are defined to be prompted, you are prompted for the values to be used in this activation. You can add new symbols and modify existing symbols of the active symbol table used during the network activation. Additionally, you can save the current symbols as a file.

➤ **To modify and confirm symbols and save the active symbol table as a file**

- 1 Activate a network with at least one symbol defined for prompting.

A **Symbol Prompting** window similar to the example below opens:



The window lists all the symbols requiring prompting in the symbol table and their values. The symbol values are taken from the corresponding master symbol table.

You can add new symbols, modify existing symbol values or leave them unchanged. The new values remain in the symbol table until the symbol table is modified specifically or until the next prompting after manual activation of a job network that uses the symbol table.



Note: 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 Choose **Continue** when you are finished.

If **Save active symbol table as file** is selected in order to save the active symbol table as a file, a window opens for the file specification: see [To save an active symbol table](#) in the section *Saving Symbol Table Information as a File*.

Choose **OK** when you are finished.

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 In a **Network Master** window, open the tabbed page **Symbol Prompting**:

- 2 Specify the type of symbol prompting by defining the Natural library and the respective user exit routine. The fields are explained in [Fields: Symbol Prompting User Exit](#).
- 3 Select **OK** when you are finished.

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](#)

- [Example of an Active Symbol Table List](#)

Fields: Symbol Prompting User Exit

The input fields of the [Symbol Prompting page](#) 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.	
Symbol Modification	Symbol modification option(s). Possible check box settings:	
	during prompting <i>(checked)</i>	A window prompts you to modify and/or confirm all symbols and save the active symbol table as a file. See also Example of an Active Symbol Table List .
	in Background <i>(checked)</i>	No prompt window opens. The symbol modification is managed by the user exit routine which runs in the background.
	<i>(both boxes checked)</i>	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.

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
Job active	Active symbol table referenced by the active job.	1
Network, active	Active symbol table referenced by the network.	2
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
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
Job master	Symbol table referenced by the master job.	5
Network version	Symbol table referenced in the network version.	6
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
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, 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 Master Definition** 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.

63 Functions for Symbol Replacement

- Results Returned by a Symbol Function 652
- Function !D or ?D - Date from Today Calculation 653
- Function !E or ?E - Date from Date Calculation 659
- Functions !MV or ?MV and !MM or ?MM - Access to Multiple-Value Symbols 660
- Function !TIMN or ?TIMN - Constant Time Values 662
- Function !W or ?W - Date Calculation with Reference Calendar 663

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.

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)
F18, 18	O	4 (Number of the weekday)

Format Sign	Format	Example
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)

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. ■ 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
JJJYY	39	24508
JJJYYYY	40	2452008
MM	11	10
MMDD	12	1029
MMDDYY	35	102908
MMYY	13	1008
MM.YY MM#YY	14	10.08
MMYYYY	15	102008
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.)

Format	Numeric Equivalent	Example
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
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

Format	Numeric Equivalent	Example
YY/MM/DD	32	08/10/29
YY-MM-DD	33	08-10-29

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.

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.

Parameters

The parameters are separated by a comma.

Syntax Element	Description
<i>date</i>	Date in the format YYYYMMDD.
<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: HHIISS 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 (.).</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>														
<i>period</i>	<table border="1"> <tr> <td>A</td> <td>Days.</td> </tr> <tr> <td>C</td> <td>Working days.</td> </tr> <tr> <td>W</td> <td>Week.</td> </tr> <tr> <td>M</td> <td>Month.</td> </tr> <tr> <td>Q</td> <td>Quarter.</td> </tr> <tr> <td>S</td> <td>Semester.</td> </tr> <tr> <td>Y</td> <td>Year.</td> </tr> </table> <p>The limits of the date period are only left if <offset>*<unit> is longer (or equal to) the period. If <offset>*<unit> 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.</p>	A	Days.	C	Working days.	W	Week.	M	Month.	Q	Quarter.	S	Semester.	Y	Year.
A	Days.														
C	Working days.														
W	Week.														
M	Month.														
Q	Quarter.														
S	Semester.														
Y	Year.														
<i>point</i>	<table border="1"> <tr> <td>F</td> <td>First day in period or unit.</td> </tr> <tr> <td>L</td> <td>Last day in period or unit.</td> </tr> </table>	F	First day in period or unit.	L	Last day in period or unit.										
F	First day in period or unit.														
L	Last day in period or unit.														

Syntax Element	Possible Values	
	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.
	<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.

XV Reporting

64 Reporting

- Report Types 670
- Viewing the Report Status List 672
- Generating Online Reports 674
- Viewing Report Properties and Deleting a Report 675
- Fields and Columns: Reporting 676
- Determination Date for Report Data 680
- Report Output Options 681
- Using Reports with Bar Charts 681
- User Restrictions for Reports 685
- Examples of Reports 686
- Generating Batch Reports 697

This section describes the **Reporting** 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 **Reporting dialog** 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 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).</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>
Network Schedule Overview	<p>Overview of scheduled and/or not yet executed, system-wide network activations.</p> <p>See also Example of Network Start Summary.</p>
Activation Overview	<p>Overview of network activations.</p> <p>See also Example of Activation Overview.</p>
Compare Symbol Tables	<p>(Special user authorization required.)</p> <p>Displays the results of the comparison of one or more symbol tables.</p> <p>See also Example of Compare Symbol Tables.</p>
Compare Networks	<p>(Special user authorization required.)</p> <p>Displays the results of the comparison of one or more networks.</p>

Report Type	Description
	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. 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 .
Network Start Overview (Bar Chart)	Displays the network start time and the end time of a previous network execution as a bar chart . See also Network Start Overview (Bar Chart) .
Network and Job Start Overview (Bar Chart)	Displays the network and job start as a bar chart . See also Example of Network and Job Start Overview (Bar Chart) .
Network Schedule Overview (Bar Chart)	Displays the schedule selection of a network as a bar chart . See also Example of Network Schedule Overview (Bar Chart) .

Viewing the Report Status List

➤ To view all reports submitted for processing

- 1 In the object workspace, select the **General** node and choose **Reporting** from the context menu.

Or:

In the **Command** box, type REPORTS (see *Direct Commands*) and press ENTER.

A **Reporting** window similar to the example below opens:

Creation time	Report type	Status
2016-11-17 10:29:32	Compare Networks	Aborted
2016-11-17 07:57:07	Compare Networks	Completed
2016-11-17 07:57:05	Compare Symbol Tables	Completed
2016-11-17 07:57:03	Activation Overview	Completed
2016-11-17 07:57:01	Network Schedule Overview	Completed
2016-11-17 07:56:59	Network Start Summary	Completed
2016-11-17 07:56:57	Schedule of Jobs	Completed
2016-11-17 07:56:55	Network Description (detailed)	Completed
2016-11-17 07:56:53	Network Description (short)	Completed
2016-11-17 07:56:51	Accounting Data	Completed
2016-11-17 07:56:49	Log - Networks not activated	Completed
2016-11-17 07:56:47	Log - Jobs with permanent errors	Completed
2016-11-17 07:56:45	Log - Jobs not started	Completed

A list of scheduled and ready reports appears.

(If no reports exist, the table in the **Reporting** window is empty.)

The reports listed in the table are used to manage and view all reports requested by the current user. Report generation is invoked within Entire Operations GUI Client whereas the reports themselves are generated asynchronously in the background via the Entire Operations monitor.

The report list contains the following columns:

Creation time	Date and time when a report generation was initiated arranged in descending order (latest date first)
Report type	Report type requested
Status	Indicates the progress in generating the report:
Queued	Report is queued for processing. A report is queued, for example, if it must wait until the Entire Operations Monitor has finished another task before it can handle the report generation (asynchronous processing). This allows you to continue working while the report is processed in the background.
Running	Report is being generated.

	Completed	Report has been generated successfully.
	Aborted	Report has been aborted. A probable reason for abortion is: <ul style="list-style-type: none"> ■ No report data found for processing ■ An inactive Entire Operations Monitor ■ A system error

Press F5 or choose **Refresh** or **Auto Refresh** (specifies an automatic refresh interval) from the context menu to refresh the processing status in the **Status** column.

- From the table in the **Reporting** window, you can select the report you want to generate by choosing **Save as File** from the context menu to determine the file to be created from the report: see [Report Output Options](#).

Generating Online Reports

This section provides instructions for generating a report in online mode with the **Reporting** function.

Report data is evaluated for the current day by default. If you want to change the evaluation date, use the **Determination Date** function.

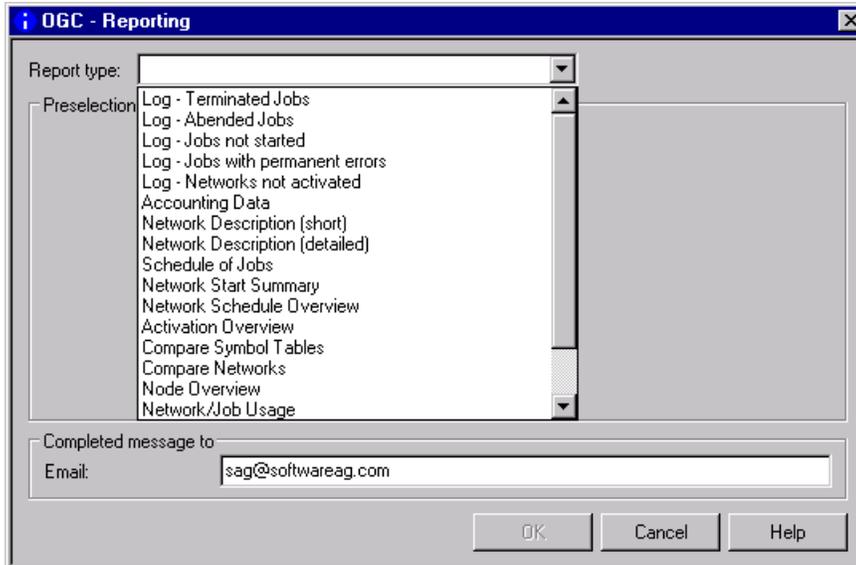


Note: The selection options provided to generate a report allow extremely flexible reporting. On the other hand, it allows you to formulate requests that will cause very many Adabas calls. For this reason, it is possible to disallow asterisk/wildcard report selection for non-administrator users: see [Range Specification and Usage Restrictions](#).

> To generate a report

- From the table in the **Reporting window**, select **Add** from the context menu.

A **Reporting** dialog opens where you choose a report from the drop-down list of the **Report type** box:



- 2 Select the required **report type** and choose **OK**.

If you choose **Cancel** instead, the selected report type is retained in the **Reporting** dialog until you open it again and choose **OK**.

Depending on the report type selected, the **Reporting** dialog provides appropriate input fields which are explained in *Fields and Columns: Reports*.

The report of the specified type is added to the table in the **Reporting** window and queued for processing.

- 3 Enter the required selection criteria and an e-mail address if you want to send a notification when the report generation has finished, and choose **OK**.

Viewing Report Properties and Deleting a Report

➤ To display the properties of a report

- Select the required report from the table in the **Reporting window** and choose **Properties** from the context menu.

A **Reporting** dialog similar to the example below appears where the properties (report type, owner, network and e-mail) are displayed:

➤ To delete a report

- Select the required report type from the table in the **Reporting** window and choose **Delete** from the context menu or press DELETE.

The report is removed from the table.

Fields and Columns: Reporting

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 in the report output file.

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 in the report file contains a dash (-).

Field/Column	Description												
Date/Time from/thru	<p>Start date/time and end date/time of the reporting period.</p> <p>Valid number range: 1 through 31.</p> <p>Default: current date and time range 00:00 - 23:59.</p> <p>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.</p>												
Run from/thru	<p>Start and end run numbers for a range of active job networks.</p> <p>Can be used only if owner and network have been selected.</p> <p>Default range: 1 - 9999 (all)</p>												
Completed message to Email	E-mail address of the user who receives an e-mail notification when the report generation completed successfully (Subject: EOR: OGC Report ended ok) or when the report generation was aborted (Subject: EOR: OGC Report ended not ok).												
Save As File	See Report Output Options .												
Type	<p>(Applies to Activation Overview only)</p> <p>Type of network activation.</p> <hr/> <p>Valid selection options:</p> <hr/> <table border="1"> <tbody> <tr> <td>All types</td> <td>All types of activations.</td> </tr> <tr> <td>Manual</td> <td>Activated manually.</td> </tr> <tr> <td>Recovery</td> <td>Activated by End-of-Job recovery processing.</td> </tr> <tr> <td>API</td> <td>Activated by the activation API. NOPUAC5N (see the section <i>API Routines</i>).</td> </tr> <tr> <td>EOJ</td> <td>Activated by an End-of-Job action.</td> </tr> <tr> <td>Scheduled</td> <td>Activated by a schedule.</td> </tr> </tbody> </table>	All types	All types of activations.	Manual	Activated manually.	Recovery	Activated by End-of-Job recovery processing.	API	Activated by the activation API. NOPUAC5N (see the section <i>API Routines</i>).	EOJ	Activated by an End-of-Job action.	Scheduled	Activated by a schedule.
All types	All types of activations.												
Manual	Activated manually.												
Recovery	Activated by End-of-Job recovery processing.												
API	Activated by the activation API. NOPUAC5N (see the section <i>API Routines</i>).												
EOJ	Activated by an End-of-Job action.												
Scheduled	Activated by a schedule.												
Compare Networks/Symbol Tables reports only:													
Show	Determines the amount of information to be shown for the compared networks.												

Field/Column	Description
	Valid selection options:
ALL	Shows all matching and differing objects and attributes.
Differences	Shows only the differing objects with the differing attributes (default).
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.
Stop	End date/time of job.
Elapsed min or Elapsed Time	Estimated run time (in minutes) of the job.
CPU Time sec	Estimated CPU time (in seconds) of the job.
<i>date to date are</i>	Line below the report table indicating the average CPU consumption. Average values of all elapsed and CPU times for all jobs in the network.
Network/Job Usage reports only:	
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.
Activation Mode	Activation mode defined for the subnetwork See also <i>Time of Activation of a Subnetwork</i> in the <i>System Overview</i> .
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:

Field/Column	Description						
	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.						
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="581 961 1101 1108"> <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)						
Bar Chart reports only:							
Run	Run number of the job						
Activation Time	Date/time when the job is activated						
Start time	Date/time when the job started						
Stop Time	Date/time when the job stopped						
Elapsed Time	Elapsed time (between job start time and job stop)						
Minimum/Maximum Elapsed Time	Minimum or maximum elapsed time per job run						
CPU Time	CPU time in milliseconds (ms)						
Stop Job	Last executed job						
Stop Message	Message of the last executed job						
Run OK?	Is true if the networks finished without errors						

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.

Determination Date for Report Data

Report data is evaluated for the current day by default. 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.

You can specify the date for which a report evaluates the data to be generated.

> To set the determination date

- 1 Select the **General** node.
- 2 Open the context menu and select **Determination Date**.

A **Determination Date** window similar to the example below opens:



- 3 Open the drop-down list box and select the required date from a calendar.

Or:

Type the required date in the input box.

- 4 Choose **OK** to save the date.

Report Output Options

The **Save as File** dialog specifies the format for the report and the location where the report is written.

Field	Explanation
File type	<p>The format used for report generation:</p> <ul style="list-style-type: none"> ■ HTML: a file in HTML (Hypertext Markup Language) format. ■ HTML5: a file in HTML5 format is required to display bar charts reports. ■ This only works if the Windows Open with function calls an HTML5-enabled browser by default. ■ CSV: a comma-separated value file for Excel. ■ XML: a file in XML (Extensible Markup Language) format without style sheet for HTML. ■ XML with style sheet for HTML: a file in XML format with a style sheet for HTML. <p>If this option is selected, a processing instruction with a reference to an external style sheet is added to the generated XML file to transform the XML file into HTML.</p>
File name	<p>The name to be used for the report file.</p> <p>If no path is specified in the File name field, the file will be saved at:</p> <pre>%LOCALAPPDATA%\Software AG\Natural\</pre>
Open with external application	<p>If this option is checked (default setting), the default Windows application associated with the specific file extension opens.</p>

The settings in the **Save as File** dialog are stored between sessions in a client-specific XML profile and are used as the default value for the next **Save as File** operation on a report of the same report type.

Using Reports with Bar Charts

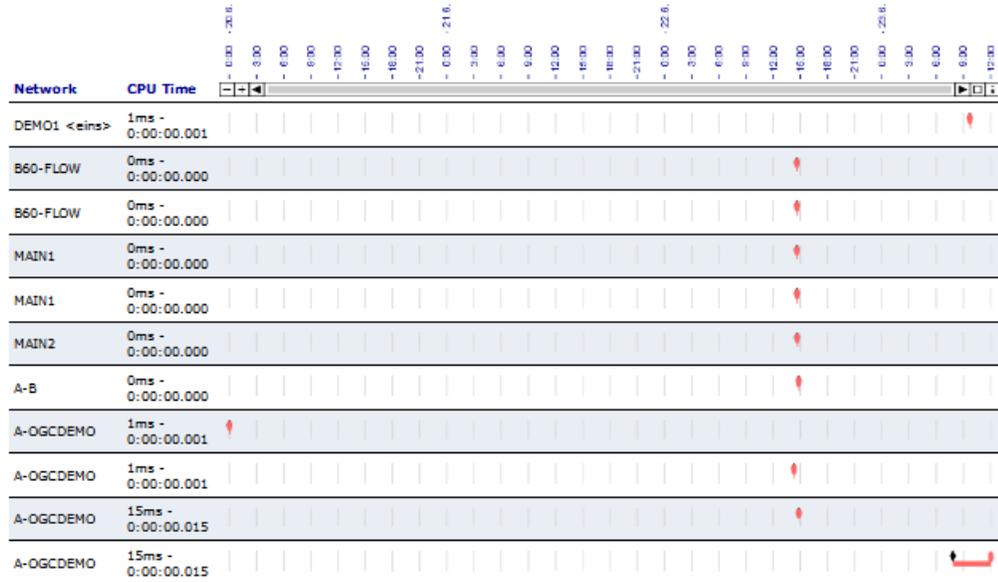
The report types **Network Start Overview (Bar Chart)**, **Network and Job Start Overview (Bar Chart)** and **Network Schedule Overview (Bar Chart)** generate the usual report columns but additionally provide an interactive diagram with a bar chart for better data visualization.

Bar charts of the report types visualize report column data as an interactive diagram. Bar charts are useful for overviewing the job flow and finding out the peak and off-peak times to balance the workload (CPU load) among the different networks.

Network Start Overview (Bar Chart) from 2015-06-20 thru 2015-06-26

Generation Date: 2015-06-23 - 12:35:00.6

- Diagram columns:
- Owner
 - Network
 - Run
 - Start Time
 - Stop Time
 - Stop Job
 - Stop Message
 - CPU Time
 - Run OK?
 - Diagram



Bar charts reports are not static but have interactive elements which are described in the following section:

- Filter Setup

Filter Setup

Filter Area:

Diagram columns: ▼ Owner Network Run
 Start Time Stop Time
 Stop Job Stop Message CPU Time Run OK?
 Diagram

By clicking on the arrow you can collapse the filter area.

Diagram columns: ▶

Changing the selection changes the displayed columns of the report:

Generation Date: 06.06.2014 - 09:39:00.7

Diagram columns: ▼ Owner Network Run
 Start Time Stop Time
 Stop Job Stop Message CPU Time Run OK?
 Diagram

Owner	Network	Run	Run OK?
INCIDENT	I5026810A	983	OK
INCIDENT	I5033788A1	8282	OK
INCIDENT	I5033788A1	8283	OK

Diagram Part:

Zoom into the diagram by selecting the time table of the diagram:

Owner	Network	Run	Run OK?
INCIDENT	I5026810A	983	OK

You can zoom into the diagram to see the diagram in more detail.



Symbol	Explanation
+ / -	You can zoom in clicking on + and zoom out clicking on -.
◀ ▶	The arrows scroll the view to left or right.
◻	You see the complete diagram again.
◻	This will show you further information/explanations.

If you move the mouse over the marker, you will see further information such as start and/or end times:

Start Time: 06.06.2014 - 09:35:00.0

Moving the mouse over the bar you see additional information about the specific network:

```

Owner:      INCIDENT
Network:    I5033788A1
Run:        8283
Start Time: 06.06.2014 - 09:35:00.0
Stop Time:  06.06.2014 - 09:35:00.0
Elapsed Time:      0ms - 0:00:00.000
CPU Time:         0ms - 0:00:00.000

```

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 **Reporting** 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)
- 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 Network Start Overview (Bar Chart)
- Example of Network and Job Start Overview (Bar Chart)
- Example of Network Schedule Overview (Bar Chart)

Example of Log - Terminated Jobs

The following is an example of the report type **Log - Terminated Jobs**:

Log - Terminated Jobs

Generation Date: 2015-06-18 - 14:22:19

Owner	Network	Job	Run	Date	Time	Message
DEMO	DEMO1 <eins>	JOB3	17	2015-06-17	17:57:00	init-user NATQA5 user NATQA
DEMO	DEMO1 <eins>	JOB3	18	2015-06-18	09:41:00	init-user NATQA5 user NATQA
DEMO	DEMO1 <zwei>	JOB1	19	2015-06-18	09:41:00	Ended ok
DEMO	DEMO2	JOB1Y	11	2015-06-18	09:42:00	Ended ok
EXAMPLE	B60-FLOW	JOB-01	267	2015-06-11	21:07:00	Dummy Job (Definition) terminated
EXAMPLE	B60-FLOW	JOB-01	268	2015-06-11	21:07:00	Dummy Job (Definition) terminated

The columns contained in the report are described in *Fields and Columns: Reports*.

Example of Log - Abended Jobs

The following is an example of the report type **Log - Abended Jobs**:

Log - Abended Jobs						
Generation Date: 2015-06-18 - 14:22:15						
Owner	Network	Job	Run	Date	Time	Message
DEMO	DEMO1 <eins>	JOB6	17	2015-06-17	17:57:00	Ended not ok
DEMO	DEMO1 <eins>	JOB6	17	2015-06-17	17:57:00	Ended not ok - STEP01 C0009 (> default C0008)
DEMO	DEMO1 <eins>	JOB6	18	2015-06-18	09:41:00	Ended not ok
DEMO	DEMO1 <eins>	JOB6	18	2015-06-18	09:41:00	Ended not ok - STEP01 C0009 (> default C0008)

The columns contained in the report are described in [Fields and Columns: Reports](#).

Example of Log - Jobs not started

The following is an example of the report type **Log - Jobs not started**:

Log - Jobs not started						
Generation Date: 2015-06-18 - 14:22:17						
Owner	Network	Job	Run	Date	Time	Message
DEMO	DEMO1 <eins>	JOB1	17	2015-06-18	10:03:00	+BDE-KKL - 0 - ABS not found
DEMO	DEMO1 <eins>	JOB1	17	2015-06-18	10:03:00	... referenced Run: -1
DEMO	DEMO1 <eins>	JOB1	17	2015-06-18	10:08:00	+BDE-KKL - 0 - ABS not found
DEMO	DEMO1 <eins>	JOB1	17	2015-06-18	10:08:00	... referenced Run: -1
DEMO	DEMO1 <eins>	JOB1	17	2015-06-18	10:13:00	+BDE-KKL - 0 - ABS not found
DEMO	DEMO1 <eins>	JOB1	17	2015-06-18	10:13:00	... referenced Run: -1

The columns contained in the report are described in [Fields and Columns: Reports](#).

Example of Log - Jobs with permanent errors

The following is an example of the report type **Log - Jobs with permanent errors**:

Log - Jobs with permanent errors						
Generation Date: 2015-06-18 - 14:22:21						
Owner	Network	Job	Run	Date	Time	Message
DEMO	DEMO1 <eins>	JOB2	18	2015-06-18	09:40:00	JCL Load - ESY5995 File not in catalog.
DEMO	DEMO1 <eins>	JOB2	18	2015-06-18	09:40:00	... TST.NOPS42.JOBS/FTPNOPCF
DEMO	DEMO1 <eins>	JOB2	18	2015-06-18	09:40:00	JCL Load Error occurred
DEMO	DEMO1 <eins>	JOB2	18	2015-06-18	09:40:00	JCL Load - ESY5995 File not in catalog.
EXAMPLE	B60-FLOW	JOB-012	274	2015-06-18	09:40:00	JCL Load - Invalid User ID NATQAS
EXAMPLE	B60-FLOW	JOB-012	274	2015-06-18	09:40:00	... SYSEORU/B60-M01

The columns contained in the report are described in [Fields and Columns: Reports](#).

Example of Log – Networks not activated

The following is an example of a **Log – Networks not activated** report:

Log - Networks not activated

Generation Date: 2015-06-18 - 14:22:31

Owner	Network	Job	Run	Date	Time	Message
INCIDENT	I5033788BA	-	25317	2015-06-16	13:00:00	No Jobs activated for Network
INCIDENT	I5033788BA	-	25318	2015-06-16	15:10:00	No Jobs activated for Network
INCIDENT	I5033788BA	-	25319	2015-06-16	17:20:00	No Jobs activated for Network
INCIDENT	I5033788BA	-	25320	2015-06-16	19:30:00	No Jobs activated for Network
INCIDENT	I5033788BA	-	25321	2015-06-16	21:40:00	No Jobs activated for Network
INCIDENT	I5033788BA	-	25322	2015-06-16	23:50:00	No Jobs activated for Network

The columns contained in the report are described in *Fields and Columns: Reports*.

Example of Accounting Data

The following is an example of the report type **Accounting Data**:

Accounting Data

Generation Date: 2015-06-18 - 14:22:25

Owner: DEMO Network: DEMO1 <eins>

Job	Run	Step	JobID	Start	Stop	Elapsed min	CPU Time sec		
JOB6	17	STEP01		2015-06-17	17:57:00	2015-06-17	17:57:00	0.00	0.01
JOB6	17		40520	2015-06-17	17:57:26	2015-06-17	17:57:26	0.00	0.01
(Network)	17			2015-06-17	17:57:26	2015-06-17	17:57:26	0.00	0.01
JOB3	18			2015-06-18	09:41:18	2015-06-18	09:41:18	0.00	0.00
JOB6	18	STEP01		2015-06-18	09:41:00	2015-06-18	09:41:00	0.00	0.01
JOB6	18		48398	2015-06-18	09:41:24	2015-06-18	09:41:24	0.00	0.01
(Network)	18			2015-06-18	09:41:18	2015-06-18	09:41:24	0.10	0.01
JOB1	19	STEP01		2015-06-18	09:41:00	2015-06-18	09:41:00	0.00	0.00
JOB1	19		48399	2015-06-18	09:41:24	2015-06-18	09:41:24	0.00	0.00
(Network)	19			2015-06-18	09:41:24	2015-06-18	09:41:24	0.00	0.00
2015-06-15 to 2015-06-21 are:							0.00	0.01	

The columns contained in the report are described under *Accounting Data/Schedule of Jobs reports only* in *Fields and Columns: Reports*.

Example of Network Description (short)

The following is an example of the report type **Network Description (short)**:

Network Description (short)

Determination Date: 2015-06-18

Generation Date: 2015-06-18 - 14:22:31

Network E40-REC-02

Owner:	EXAMPLE
Description:	E40-ST
Execution Node:	NO148
Symbol Table:	E40-ST
Schedule Times	Earliest Start:
	Latest Start:
	Deadline:
Send Late Message to:	
Long Description:	

Job E40-J01-RC

Owner:	EXAMPLE
Network:	E40-REC-02
Type:	NAT Natural
Description:	Recovery action
Special Type:	
Location:	NAT
DSN/Library:	SYSEORU
Member:	E40-J01R
Execution Node:	NO148
Symbol Table:	E40-ST
Schedule Times	Earliest Start:
	Latest Start:
	Deadline:
	Elapsed min: 0.51
Send Late Message to:	
Long Description:	
Input Conditions:	
Prerequisite Resources:	
End-of-Job Actions:	

The columns contained in the report are described in [Fields and Columns: Reports](#).

Example of Network Description (detailed)

Network Description (detailed)

Determination Date: 2015-06-18
Generation Date: 2015-06-18 - 14:22:36

Network E40-REC-02

Owner:	EXAMPLE
Description:	E40-ST
Execution Node:	NO148
Symbol Table:	E40-ST
Earliest Start:	
Schedule Times	Latest Start:
	Deadline:
Send Late Message to:	
Network E40-REC-02 -----	
Long Description:	contains the recovery job(s) for network E40-REC-01. Must NOT be activated. See the prose description of E40-REC-01 for details.

Job E40-J01-RC

Owner:	EXAMPLE
Network:	E40-REC-02
Type:	NAT Natural
Description:	Recovery action
Special Type:	
Location:	NAT
DSN/Library:	SYSEORU
Member:	E40-J01R
Execution Node:	NO148
Symbol Table:	E40-ST
Earliest Start:	
Schedule Times	Latest Start:
	Deadline:
Elapsed min: 0.51	
Send Late Message to:	
Long Description:	
Input Conditions:	
Prerequisite Resources:	
End-of-Job Actions:	

Example of Schedule of Jobs

The following is an example of the report type **Schedule of Jobs**:

Production Plan from 2015-06-20 thru 2015-06-26

Determination Date: 2015-06-23

Generation Date: 2015-06-23 - 12:19:08

Date: 2015-06-20

Owner	Network	Job	Description	Start	Elapsed Time
INCIDENT	I1042425B	XXX	xxxx		0.00
INCIDENT	I5033788BA	JOB1			0.99
INCIDENT	I5095089A	SN000101		23:50	9.92
INCIDENT	I5095089A	SN000102		23:55	9.91
NATQA5	A-OGCDEMO	J-OGCDEMO	Job for OGC sample network	12:00	0.01
NATQA5	A-OGCDEMO	J-OGCDEMO1	Job1 for OGC sample network	12:00	0.01
NATQA5	A-OGCDEMO	J-OGCDEMO2	Job2 for OGC sample network	12:00	0.00
NATQA5	A-OGCDEMO	J-OGCDEMO3	Job3 for OGC sample network	12:00	0.00
NATQA5	A-OGCDEMO	J-OGCDEMO4	Job4 for OGC sample network	12:00	0.00
NATQA5	A-OGCDEMO	J-OGCDEMO5	Job5 for OGC sample network	12:00	0.00
NATQA5	A-OGCDEMO	J-OGCDEMO6	Job6 for OGC sample network	12:00	0.00

The columns contained in the report are described under [Accounting Data/Schedule of Jobs reports only](#) in [Fields and Columns: Reports](#).

Example of Network Start Summary

The following is an example of the report type **Network Start Summary**:

Network Start Summary from 2015-06-20 thru 2015-06-26

Generation Date: 2015-06-23 - 12:19:29

Owner: INCIDENT Network: I1042163

Run	Date	Time	Job	Message
448	2015-06-22	14:17		Network Activation
448	2015-06-22	14:17	I104216331	Schedule dependency: Calendar missing
448	2015-06-22	14:17	I104216332	EOR3239 - Node 0 OpSys Class not determinable
448	2015-06-22	14:17		active Network
449	2015-06-23	07:50		Network Activation
449	2015-06-23	07:50	I104216331	Schedule dependency: Calendar missing
449	2015-06-23	07:50	I104216332	EOR3239 - Node 0 OpSys Class not determinable
449	2015-06-23	07:50		active Network

The columns contained in the report are described in [Fields and Columns: Reports](#).

Example of Network Schedule Overview

The following is an example of the report type **Network Schedule Overview**:

Schedule from 2015-06-20 thru 2015-06-23

Determination Date: 2015-06-23
Generation Date: 2015-06-23 - 12:19:12

Date	Time	Owner	Network	Run	Type
2015-06-22	00:00:00	INCIDENT	I1042163		Schedule, periodic
2015-06-23	00:00:00	INCIDENT	I1042163		Schedule, periodic
2015-06-23	16:00:00	INCIDENT	I1042163	448	Activation In Progress
2015-06-24	16:00:00	INCIDENT	I1042163	449	Activation In Progress
2015-06-20	10:00:00	INCIDENT	I1043425B		Schedule, periodic
2015-06-21	10:00:00	INCIDENT	I1043425B		Schedule, periodic
2015-06-22	10:00:00	INCIDENT	I1043425B		Schedule, periodic
2015-06-23	10:00:00	INCIDENT	I1043425B		Schedule, periodic
2015-06-22	00:00:00	INCIDENT	IS026810A		Schedule, periodic
2015-06-23	00:00:00	INCIDENT	IS026810A		Schedule, periodic

The columns contained in the report are described in *Fields and Columns: Reports*.

Example of Activation Overview

The following is an example of the report type **Activation Overview**:

Activation Overview

Generation Date: 2015-06-23 - 12:19:18

Owner	Network	Job	Run	Date	Time	Type	Message
INCIDENT	I1042425B	-	6850	2015-06-20	00:00:08	Schedule	Activation Network 06-20 00:00
INCIDENT	I1042425B	-	6851	2015-06-20	00:00:08	Schedule	Activation Network 06-20 00:03
INCIDENT	I1042425B	-	6852	2015-06-20	00:00:08	Schedule	Activation Network 06-20 00:06
INCIDENT	I1042425B	-	6853	2015-06-20	00:00:08	Schedule	Activation Network 06-20 00:09
INCIDENT	I1042425B	-	6854	2015-06-20	00:00:08	Schedule	Activation Network 06-20 00:12
INCIDENT	I1042425B	-	6855	2015-06-20	00:00:08	Schedule	Activation Network 06-20 00:15

The columns contained in the report are described in *Fields and Columns: Reports*.

Example of Compare Symbol Tables

The following is an example of the report type **Compare Symbol Tables**:

Compare Symbol Tables

Generation Date: 2015-06-23 - 12:19:20

Owner: EXAMPLE Symbol Table: DEMO <v1>

Symbol	Value	Type	Prompt	Result	Attribute	Owner	Symbol Table
FILE-1	EOR.DEMO.SRCE	A	E	Not found in		EXAMPLE	EX-ST-COMN
				Not found in		EXAMPLE	EX-VSE-1
				Not found in		EXAMPLE	EXA-SYMBOL
				Not found in		EXAMPLE	EXAM-ST1
				Not found in		EXAMPLE	E20-ST
Additional Symbol				Owner	Symbol Table		
CLASS				EXAMPLE	EX-ST-COMN		
CLASS-K				EXAMPLE	EX-ST-COMN		
CLASSA				EXAMPLE	EX-ST-COMN		
CLASSB				EXAMPLE	EX-ST-COMN		

Columns: Compare Symbol Tables

The columns contained in the report are described under [Compare Networks/Symbol Tables reports only](#) in [Fields and Columns: Reports](#).

Example of Compare Networks

The following is an example of the report type **Compare Networks**:

Compare Networks

Generation Date: 2015-06-23 - 12:19:26

Owner: EXAMPLE Network: B60-FLOW

Result	Attribute	Owner	Network
Different in	Description Loop existing Execution Node JCL Node Submit User ID Escape Act Escape Sub Modification without Prompting	EXAMPLE	B60-FLOW
JOB-01	JOB	MAC	where it all starts
		Different in	Special Type Restartable JCL Location JCL Load Mode Escape Character Submit Description File/NatLib Member JCL Node Execution Node Submit User ID Sysout User ID Member Type JCL logging Collect Sysout Default User ID Share Sysout
Additional Job		Owner	Network
JOB-1-TEST		EXAMPLE	B60-FLOW

Columns: Compare Networks

The columns contained in the report are described under [Compare Networks/Symbol Tables reports only](#) in [Fields and Columns: Reports](#).

Example of Node Overview

The following is an example of the report type **Node Overview**:

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: Reports*.

Example of Network/Job Usage

The following is an example of the report type **Network/Job Usage**:

Network/Job Usage

Generation Date: 2016-11-23 - 16:47:10

Used Job			Using Job				
Owner	Network	Job	Owner	Network	Job	Usage	Activation Mode
EXAMPLE	E01-CONTI	*	EXAMPLE	B60-FLOW	JOB-SUBNET	Subnet	D
EXAMPLE	E01-CONTI	*	EXAMPLE	B60-FLOW	JOB-01	EOJ Activation	
EXAMPLE	E01-CONTI	*	EXAMPLE	MAIN1	JOB-01	EOJ Activation	
EXAMPLE	E01-CONTI	*	EXAMPLE	MAIN2	JOB-01	EOJ Activation	
EXAMPLE	E40-REC-02	E40-J01-RC	EXAMPLE	E40-REC-01	E40-J02	Recovery	
EXAMPLE	E62-NET-B	*	EXAMPLE	E62-NET	E62-J02	EOJ Activation	
EXAMPLE	V40-REC-02	E40-J01-RC	EXAMPLE	V40-REC-01	E40-J02	Recovery	

The columns contained in the report are described under *Network/Job Usage reports only* in *Fields and Columns: Reports*.

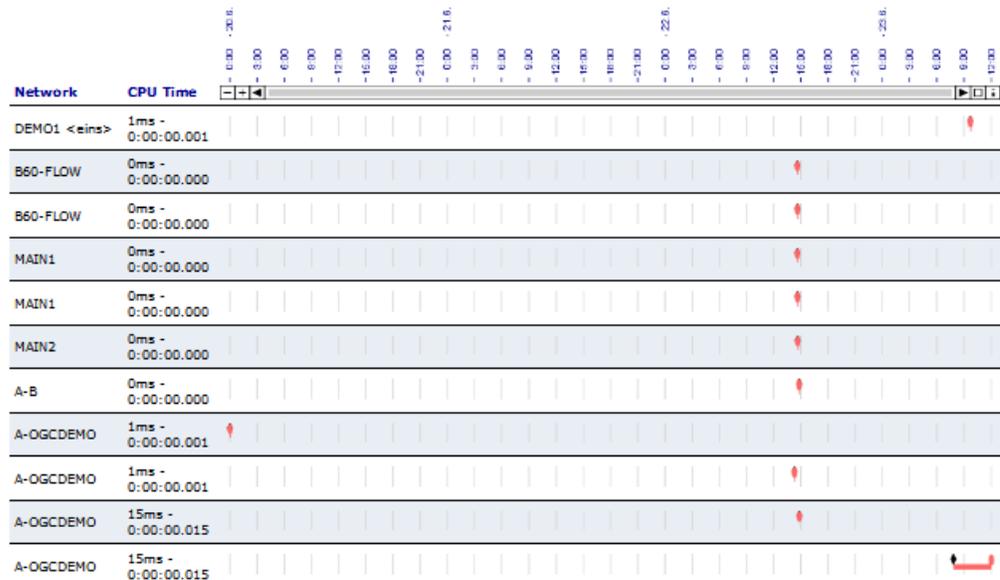
Example of Network Start Overview (Bar Chart)

The following is an example of the report type **Example of Network Start Overview (Bar Chart)**:

Network Start Overview (Bar Chart) from 2015-06-20 thru 2015-06-26

Generation Date: 2015-06-23 - 12:35:00.6

Diagram columns: Owner Network Run
 Start Time Stop Time
 Stop Job Stop Message CPU Time Run OK?
 Diagram



The columns contained in the report are described under *Bar Chart reports only* in *Fields and Columns: Reports*.

See also the section *Using Reports with Bar Charts* for information on how to handle bar charts.

Example of Network and Job Start Overview (Bar Chart)

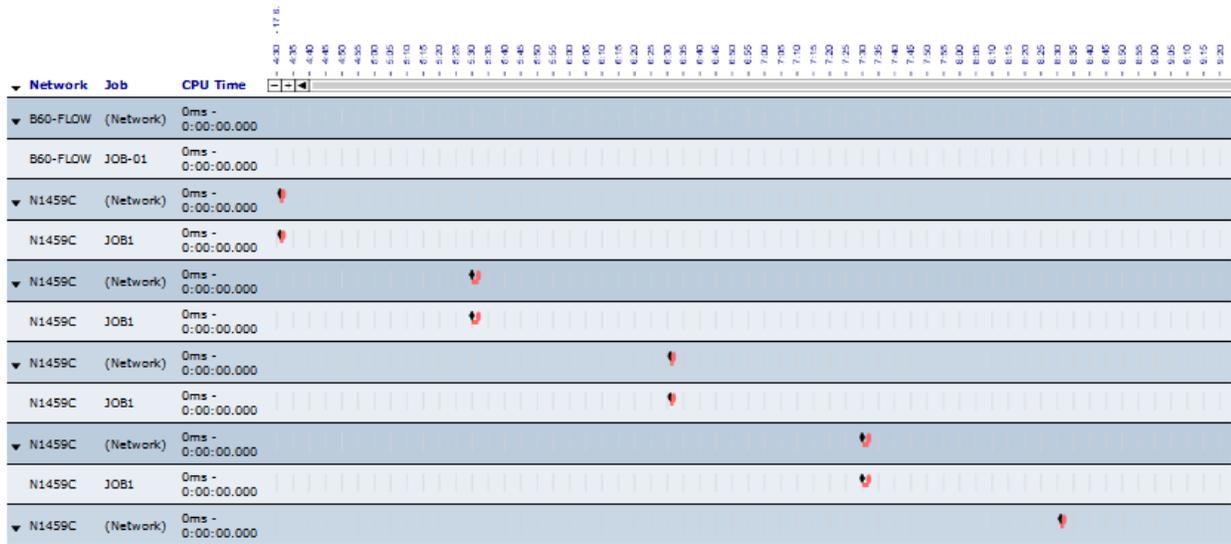
The following is an example of the report type **Network and Job Start Overview (Bar Chart)**:

Network and Job Start Overview (Bar Chart) from 2015-06-17 thru 2015-06-17

Generation Date: 2015-06-17 - 14:24:00.8

Diagram columns:

- Owner
- Network
- Job
- Run
- Start Time
- Stop Time
- Stop Job
- Stop Message
- CPU Time
- Run OK?
- Diagram



The columns contained in the report are described under *Bar Chart reports only* in *Fields and Columns: Reports*.

See also the section *Using Reports with Bar Charts* for information on how to handle bar charts.

Example of Network Schedule Overview (Bar Chart)

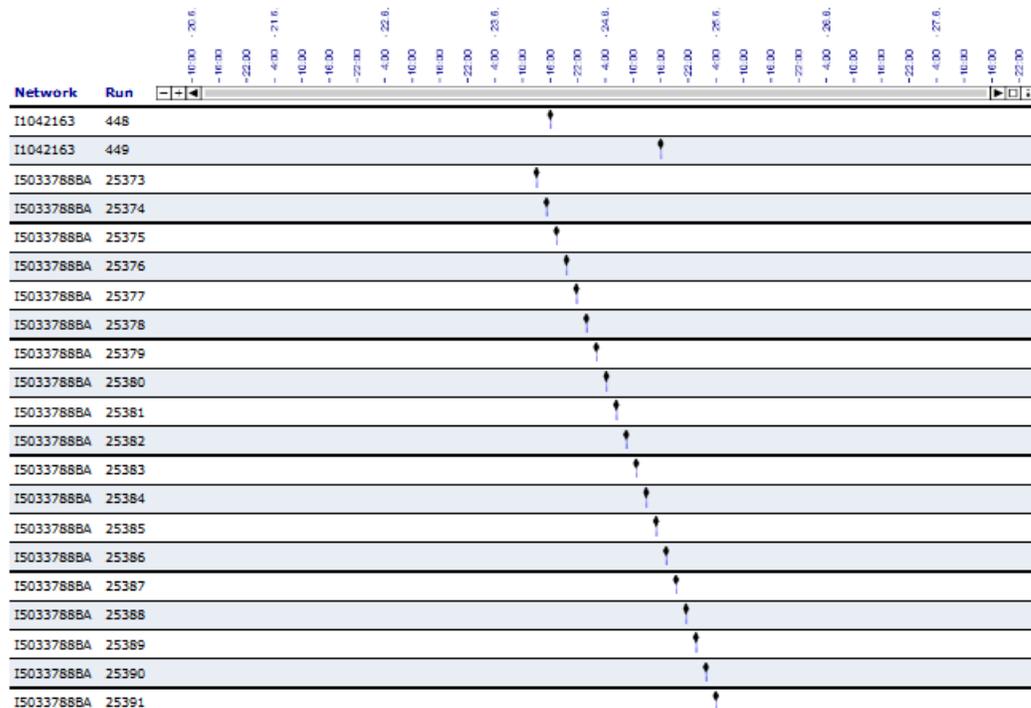
The following is an example of the report type **Network Schedule Overview (Bar Chart)**:

Network Schedule Overview (Bar Chart) from 2015-06-20 thru 2015-06-26

Generation Date: 2015-06-23 - 12:34:23

Diagram columns: ▾

- Owner Network Run
 Activation Time Minimum Elapsed Time Maximum Elapsed Time
 Diagram



The columns contained in the report are described under *Bar Chart reports only* in *Fields and Columns: Reports*.

See also the section *Using Reports with Bar Charts* for information on how to handle bar charts.

Generating Batch Reports

You can use the batch command client of Entire Systems Management to generate a report in batch mode: see *Commands for the Batch Command Client* in the section *Using Entire Operations in Batch Mode*.

XVI

Cross References

65 Cross References

- Types of Cross Reference Reports 702
- Generating Cross Reference Reports Online 703
- Fields and Columns: Cross References 709
- Examples of Cross Reference Reports 714
- Generating Cross Reference Reports in Batch 719

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 window** are described in the following table.

Report Type	Description
User 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 .
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 by default. If you want to change the evaluation date, use the [Determination Date](#) function described in the section *Reporting*.

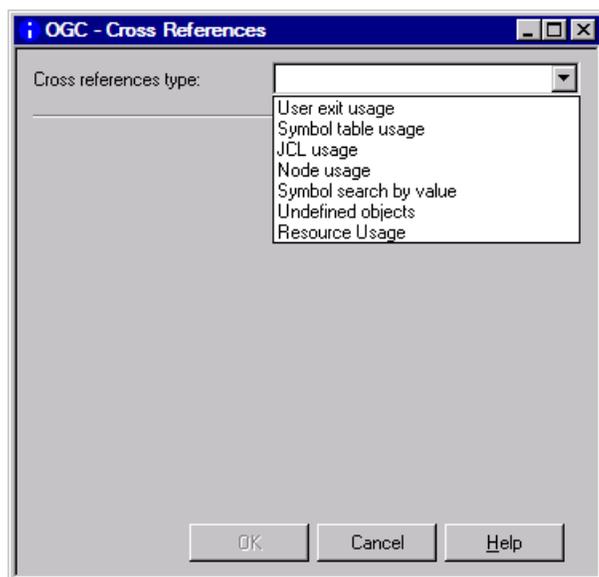
➤ To generate a cross reference report

- 1 In the object workspace, select the **General** node and choose **Cross References** from the context menu.

Or:

In the **Command** input field, type `XREF` (see *Direct Commands*) and press ENTER.

A **Cross References** window opens where you can select a cross reference type from the drop-down list box:



- 2 Select the **type of cross reference** for which you want to generate a report.

Depending on the report type selected, proceed as described in the relevant step for

[User Exit Usage](#)

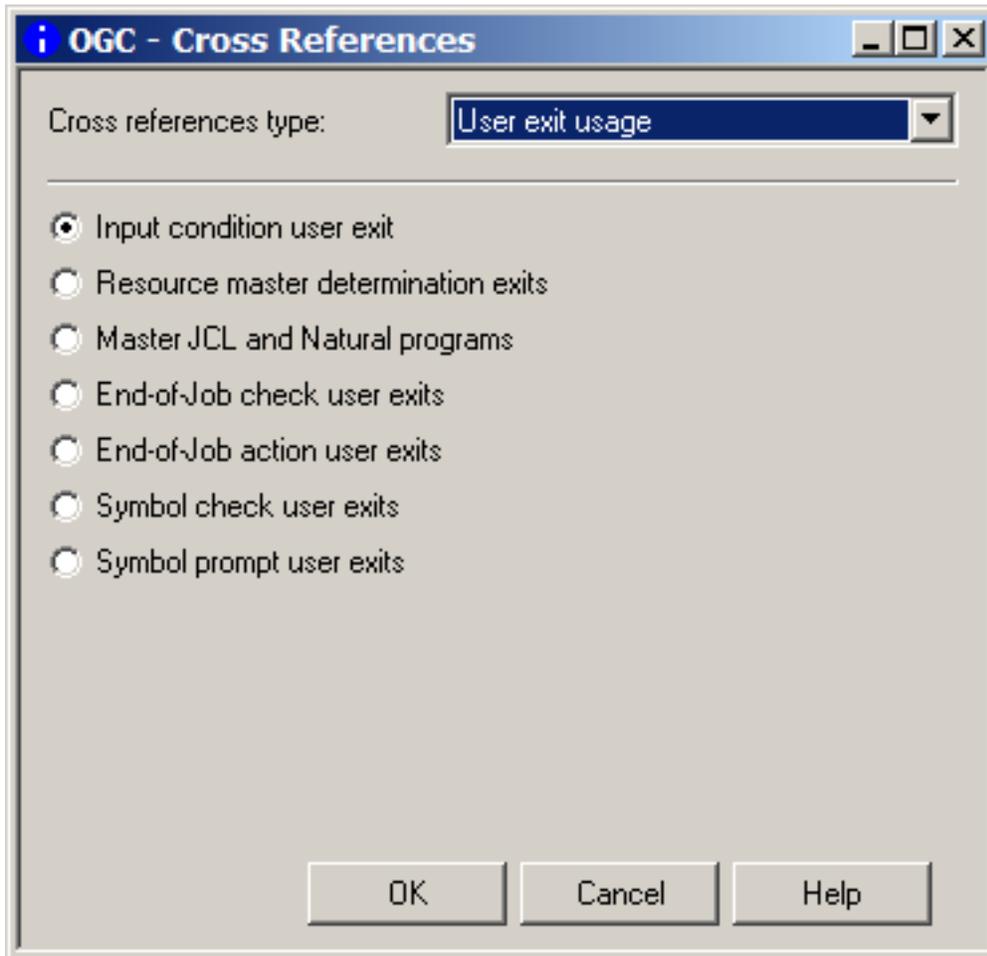
[Symbol Table Usage](#)

[JCL Usage](#)

[Node Usage](#)

[Symbol Search by Value](#)
[List Undefined Objects](#)
[Resource Usage](#)

- 3 **User Exit Usage:** If you select **User exit usage** from the **Cross References window**, a selection list of user exit types similar to the example below appears:

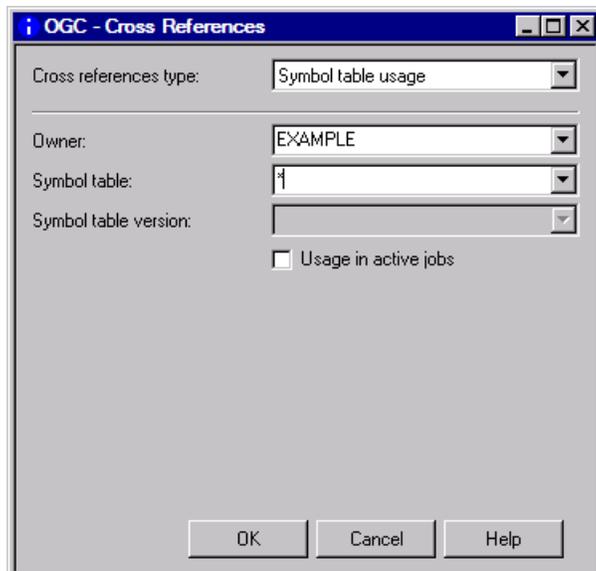


Select the required type and choose **OK**.

In the **Save as File** dialog that opens, specify the output file to be produced for the report as described in [Report Output Options](#) (section *Reporting*), and choose **OK**.

A report with a list of used user exits is generated into the output file as shown in [Example of User Exit Usage](#).

- 4 **Symbol Table Usage:** If you select **Symbol table usage** from the **Cross References window**, input fields appear as shown in the following example:

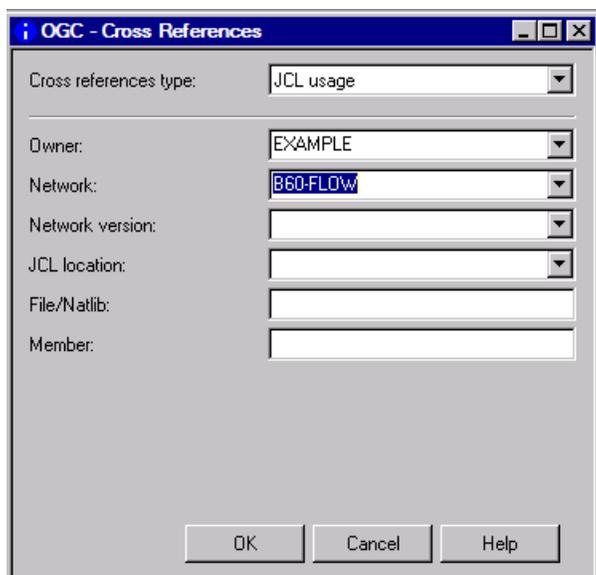


Specify the selection criteria required for the report to be produced. For valid input values, see [Fields and Columns: Cross References](#). When you are finished, choose **OK**.

In the **Save as File** dialog that opens, specify the output file to be produced for the report as described in [Report Output Options](#) (section *Reporting*), and choose **OK**.

A report with a list of used symbol tables is generated into the output file as shown in [Example of Symbol Table Usage](#).

- 5 **JCL Usage:** If you select **JCL usage** from the [Cross References window](#), input fields appear as shown in the following example:

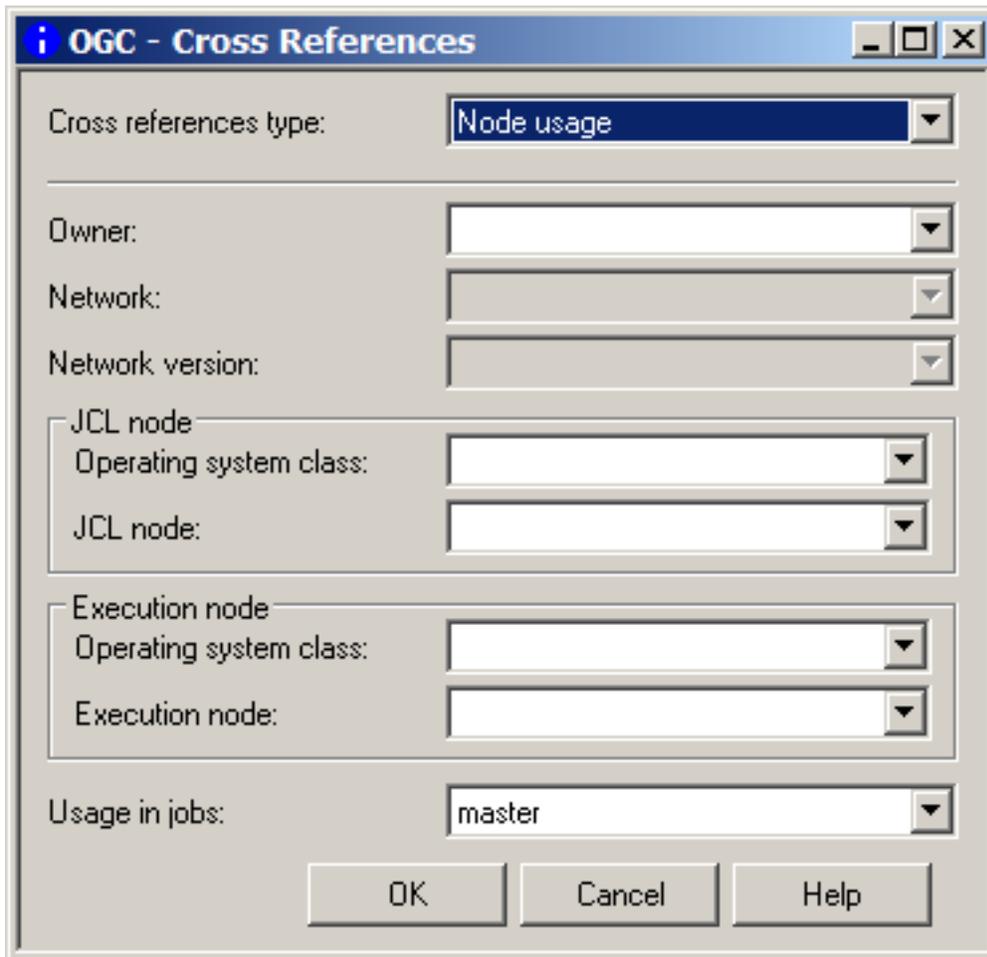


Specify the selection criteria required for the report to be produced. For valid input values, see *Fields and Columns: Cross References*. When you are finished, choose **OK**.

In the **Save as File** dialog that opens, specify the output file to be produced for the report as described in *Report Output Options* (section *Reporting*), and choose **OK**.

A report with a list of used JCL is generated into the output file as shown in *Example of JCL Usage*.

- 6 **Node Usage:** If you select **Node usage** from the **Cross References window**, input fields appear as shown in the following example:



The screenshot shows a dialog box titled "OGC - Cross References". It contains several input fields and buttons. The "Cross references type" dropdown is set to "Node usage". Below this are fields for "Owner", "Network", and "Network version". There are two sections for node selection: "JCL node" and "Execution node", each with "Operating system class" and "JCL node" or "Execution node" dropdowns. The "Usage in jobs" dropdown is set to "master". At the bottom are "OK", "Cancel", and "Help" buttons.

Specify the selection criteria required for the report to be produced. For valid input values, see *Fields and Columns: Cross References*. When you are finished, choose **OK**.

In the **Save as File** dialog that opens, specify the output file to be produced for the report as described in *Report Output Options* (section *Reporting*), and choose **OK**.

A report with a list of used nodes is generated into the output file as shown in *Example of Node Usage*.

- 7 **Symbol Search by Value:** If you select **Symbol search by value** from the **Cross References window**, input fields appear as shown in the following example:

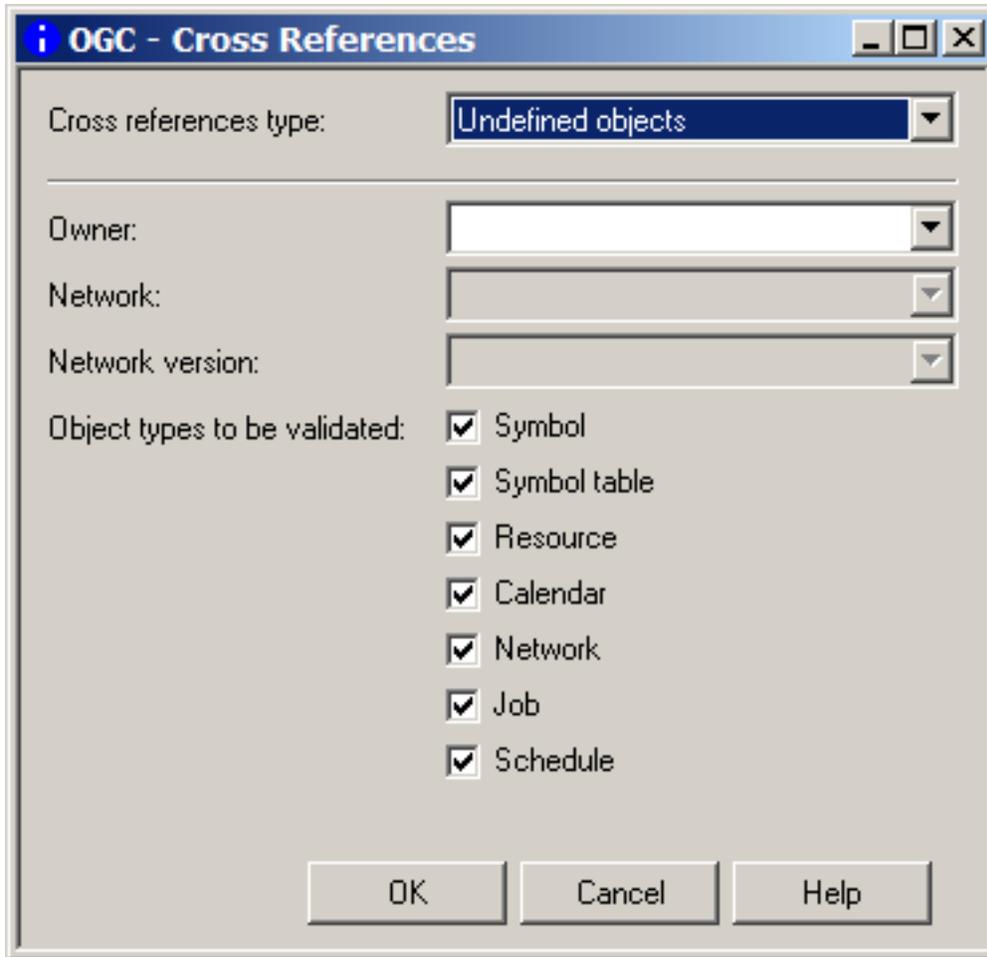
The screenshot shows a dialog box titled "OGC - Cross References". It contains several input fields and checkboxes. The "Cross references type" dropdown is set to "Symbol search by value". Below it are dropdown menus for "Owner", "Symbol table", "Symbol table version", and "Symbol". The "Usage in" section has "Master" checked and "Active" unchecked. The "Symbol value" section includes a "Mult. value index" with "From" set to "1" and "To" set to "*". The "At position" dropdown is set to "anywhere". There is an empty "Scan for" text field and an unchecked "Case sensitive" checkbox. At the bottom are "OK", "Cancel", and "Help" buttons.

Specify the selection criteria required for the report to be produced. For valid input values, see *Fields and Columns: Cross References*. When you are finished, choose **OK**.

In the **Save as File** dialog that opens, specify the output file to be produced for the report as described in *Report Output Options* (section *Reporting*), and choose **OK**.

A report with a list of used symbols is generated into the output file as shown in *Example of Symbol Search by Value*.

- 8 **Undefined Objects:** If you select **Undefined objects** from the **Cross References window**, input fields appear as shown in the following example:



Specify the selection criteria required for the report to be produced. For valid input values, see [Fields and Columns: Cross References](#).

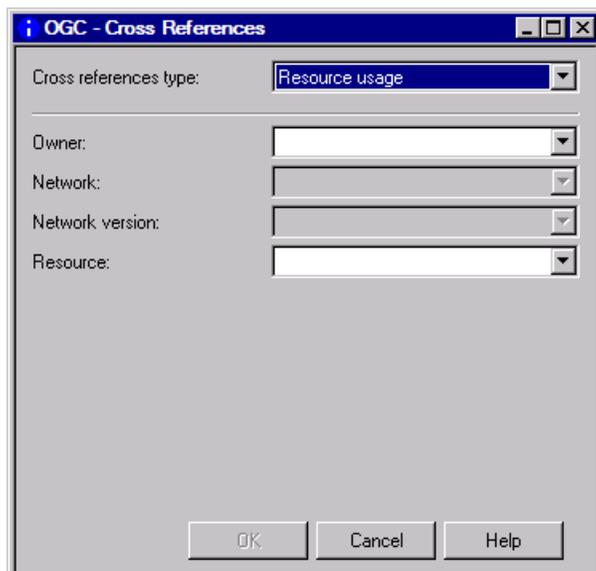
Unmark the object type(s) you do not require. All object types for which you can search are selected by default.

When you are finished, choose **OK**.

In the **Save as File** dialog that opens, specify the output file to be produced for the report as described in [Report Output Options](#) (section *Reporting*), and choose **OK**.

A report with a list of used undefined objects is generated into the output file as shown in [Example of Undefined Objects](#).

- 9 **Resource Usage:** If you select **Resource usage** from the [Cross References window](#), input fields appear as shown in the following example:



Specify the selection criteria required for the report to be produced. For valid input values, see [Fields and Columns: Cross References](#). When you are finished, choose **OK**.

In the **Save as File** dialog that opens, specify the output file to be produced for the report as described in [Report Output Options](#) (section *Reporting*), and choose **OK**.

A report with a list of used resources is generated into the output file as shown in [Example of Resource Usage](#).

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 in the report output file.

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. Enter an asterisk (*) 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

Field/Column	Explanation	
	<p>an end value (see also <i>Valid Name Specifications in Direct Commands</i>) or open a selection list of names.</p> <p>Enter an asterisk (*) to select all names.</p>	
Network version	<p>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 in Direct Commands</i>) or open a selection list of names.</p> <p>Leave the field blank to select a blank name.</p> <p>Enter an asterisk (*) to select all names.</p>	
Symbol table	<p>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 in Direct Commands</i>) or open a selection list of names.</p> <p>Enter an asterisk (*) to select all names.</p>	
Symbol table version	Version of the specified symbol table.	
Resource	<p>Name of a resource.</p> <p>This can reflect real resources or can describe a fictitious resource.</p> <p>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 in Direct Commands</i>) or open a selection list of names.</p> <p>Enter an asterisk (*) to select all names.</p>	
User Exit Usage reports only:		
Job	Name of the job using the user exit.	
Exit Library	Name of the library that contains the user exit.	
User Exit	Name of the user exit.	
Enabled	Depending on the type of user exit selected, indicates whether the user exit is enabled (Yes) or disabled (No).	
Background	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 .		

Field/Column	Explanation	
(User exit selection buttons)	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 check-box settings:	
	<i>unchecked</i>	Selects master jobs only (default).
	<i>checked</i>	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.	
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. Select a name from the drop-down list box or leave the field blank to select all names.	
Execution node	Name of an execution node or a range of names. Select a name from the drop-down list box or leave the field blank to select all names.	

Field/Column	Explanation								
Operating system class	Operating system under which the JCL or execution node is used. Select an operating system from the drop-down list box or leave the field blank to select all names.								
Usage in jobs	Jobs to be selected								
	Possible check-box settings:								
	<table border="1"> <tr> <td>master</td> <td>Selects master jobs only (default).</td> </tr> <tr> <td>active</td> <td>Selects active jobs only.</td> </tr> </table>	master	Selects master jobs only (default).	active	Selects active jobs only.				
master	Selects master jobs only (default).								
active	Selects active jobs only.								
Symbol Search by Value reports only:									
Symbol	Symbol for which to search. Enter an asterisk (*) to select all symbols.								
Usage in: Master/Active	Symbols in which to search. Select Master (default) and/or Active 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 check-box settings:								
	<table border="1"> <tr> <td>anywhere</td> <td>Anywhere within the symbol value (default)</td> </tr> <tr> <td><i>nn</i></td> <td>At the exact position: <i>nn</i> is any value in the range from 1 through 80</td> </tr> <tr> <td>at beginning</td> <td>At the beginning of the symbol value</td> </tr> <tr> <td>at end</td> <td>At the end of the symbol value</td> </tr> </table>	anywhere	Anywhere within the symbol value (default)	<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	at end	At the end of the symbol value
	anywhere	Anywhere within the symbol value (default)							
	<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								
at end	At the end of the symbol value								
Scan for	Scan value. Searches in a symbol value for a string of up to 80 characters								
Case sensitive	Mark this check box to perform a case-sensitive search.								
Undefined Objects reports only:									
This report inspects specified objects (symbol, symbol, table, resoure, 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.									

Field/Column	Explanation	
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).		
Object References	Owner	Owner of the network that references the undefined object.
	Network	Network and network version (if defined) that references the undefined object. For possible version entries , see Version Defined/Determined .
	Object: Type Name Component	Type and name of the object the references the undefined object. If relevant, the component of the referenced object is listed, for example, the input condition of a job.
Undefined Object	Owner	Owner of the undefined object. If this column is empty or if it contains (same), the owner name is identical to Owner in the Object References section.
	Parent: Type Name	Type and name of the object that references the undefined object. If the Name column is empty or if it contains (same), the object name is identical to Name in the Object References section.
	Version Defined	Version defined for the referenced Parent . Possible version entries are: (current) Current object version. (nv) Network version. (svn) Symbol table version of the network. (svj) Symbol table version of the job.
	Version Determined	A report is generated for the current date by default. If you schedule the report to run on a different date, the object version valid on this date is listed in this column. See also determination date in the section <i>General Functions and Metanode General</i> .
	Object: Type Name	Type and name of the undefined object.
	Version Defined Version Determined	Version defined for the undefined object. If a determination date is set for the cross reference report (see also <i>General Functions and Metanode General</i>), the version defined for the undefined object on the given specified is listed in Version Determined .

Field/Column		Explanation
		(void) denotes that a version is defined for the object but was not be found on the specified determination date.
Resource Usage reports only:		
Type	Type of resource	
	Possible values:	
	U	Not reusable, quantitative
	R	Reusable, quantitative
	N	Not quantitative (absolute)
Quantity - Initial	Initial quantity Total amount of the resource defined to the system	
Quantity - Used	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	
Begin	Date and time of the allocation	

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 **User Exit usage**:

Input condition user exit

Generation Date: 2017-02-12 - 12:53:19

Owner	Network	Job	Exit Library	User 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

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**:

Symbol table usage

Determination Date: 2015-05-07

Generation Date: 2015-05-07 - 16:37:02

Symbol Table Owner: EXAMPLE Symbol Table: E20-ST

Owner	Network	Run	Job
EXAMPLE	E20-DYN-01		
EXAMPLE	E20-DYN-01		E20-J01

Symbol Table Owner: EXAMPLE Symbol Table: E20-ST-VSE

Owner	Network	Run	Job
EXAMPLE	V20-DYN-01		
EXAMPLE	V20-DYN-01		E20-J01

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**:

JCL usage

Generation Date: 2015-05-01 - 11:32:43

Owner: EXAMPLE

Network	Job	Job Type	Node	JCL Location	File/Member
B60-FLOW	JOB-01	JOB	31	MAC	SYSEORU B60-M02
B60-FLOW	JOB-012	JOB	31	MAC	SYSEORU B60-M01
B60-FLOW	JOB-013	JOB	31	MAC	SYSEORU B60-M01
B60-FLOW	JOB-014	JOB	31	MAC	SYSEORU B60-M01
B60-FLOW	JOB-019	JOB	31	MAC	SYSEORU B60-M01
B60-FLOW	JOB-02	JOB	31	MAC	SYSEORU B60-M02
B60-FLOW	JOB-03	NAT	31	NAT	SYSEORU B60-P01
B60-FLOW	JOB-04	JOB	31	MAC	SYSEORU B60-M01
B60-FLOW	JOB-06	JOB	31	MAC	SYSEORU B60-M02
B60-FLOW	JOB-1-TEST	JOB	55522	NAT	SYSEXPB ACCEPX01

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**:

Node usage

Generation Date: 2015-05-01 - 12:30:02

Owner	Network	Run	Job	Job Type	JCL Node	Execution Node
EXAMPLE	B60-FLOW		JOB-01	JOB	31	148
EXAMPLE	B60-FLOW		JOB-012	JOB	31	31
EXAMPLE	B60-FLOW		JOB-013	JOB	31	31
EXAMPLE	B60-FLOW		JOB-014	JOB	31	31
EXAMPLE	B60-FLOW		JOB-015	DUM	31	31
EXAMPLE	B60-FLOW		JOB-019	JOB	31	31
EXAMPLE	B60-FLOW		JOB-02	JOB	31	31
EXAMPLE	B60-FLOW		JOB-03	NAT	31	31
EXAMPLE	B60-FLOW		JOB-04	JOB	31	31
EXAMPLE	B60-FLOW		JOB-05	DUM	31	31
EXAMPLE	B60-FLOW		JOB-06	JOB	31	31
EXAMPLE	B60-FLOW		JOB-1-TEST	JOB	55522	42

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**:

Symbol search by value

Determination Date: 2015-05-09

Generation Date: 2015-05-09 - 12:34:59

Owner	Symbol Table	Symbol	MV	Format	Network	Run	Symbol Value
EXAMPLE	EXA-SYMBOL	CLASS		A			G
EXAMPLE	EXA-SYMBOL	JOBLIB		A			NOP.EXAMPLE.LOAD
EXAMPLE	EXA-SYMBOL	MSGCLASS		A			X
EXAMPLE	EXA-SYMBOL	STEPLIB		A			NOP.EXAMPLE.LOAD
EXAMPLE	EXA-SYMBOL	UID		A			NOP

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 **Undefined objects**:

Undefined objects													
Determination Date: 2016-04-25													
Generation Date: 2016-04-25 - 11:17:24													
Object References					Undefined Object								
Owner	Network	Object			Owner	Parent				Object			
		Type	Name	Component		Type	Name	Version Defined	Determined	Type	Name	Version Defined	Determined
NATQA5	KDEMONET	Job Master	ERSETZEN	EOJ Checking & Actions		Symbol Table Master	TEST			Symbol Master	JJ		
NATQA5	KDEMONET	Job Master	ESMACC	Input Condition	NATQA5	Symbol Table Master	TEST			Symbol Master	?=)(/&%\$ "Ü		
NATQA5	KDEMONET	Job Master	MU							Resource Master	NATQA5-TEST-MULTI		

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 in the network KDEMONET of the owner NATQA5:

- The symbol JJ in the symbol table master TEST is referenced in the **EOJ Checking & Actions** defined for the job ERSETZEN.
- The symbols ?=)(/&%\$ "Ü are referenced in the symbol table master TEST as an input condition of the job ESMACC.
- The resource NATQA5-TEST-MULTI is referenced by the job MU.

The columns 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**:

Resource Usage

Generation Date: 18.05.2015 - 14:59:19

Resource	Type	Quantity		Used by		Run	Job	Begin
		Initial	Used	Owner	Network			
BOA-RES	R	20.00	5.00	XSETABO1	IMPXP2	2645	JOB2	13.05.2015 - 08:01:07.7
BOA-RES	R	20.00	5.00	XSETABO1	IMPXP2	2646	JOB2	14.05.2015 - 08:01:36.4
BOA-RES	R	20.00	5.00	XSETABO1	IMPXP2	2648	JOB2	17.05.2015 - 00:07:42.1
BOA-RES	R	20.00	5.00	XSETABO1	IMPXP2	2649	JOB2	18.05.2015 - 00:13:58.2
DEA1-KEEP-01	R	100.00	3.00	SN	RES-DEA1	21882	J03-KEEP	13.05.2015 - 08:12:50.8
DEA1-KEEP-01	R	100.00	3.00	SN	RES-DEA1	21883	J03-KEEP	13.05.2015 - 13:12:37.0
DEA1-KEEP-01	R	100.00	3.00	SN	RES-DEA1	21884	J03-KEEP	14.05.2015 - 08:12:46.9
DEA1-KEEP-01	R	100.00	3.00	SN	RES-DEA1	21885	J03-KEEP	14.05.2015 - 13:13:02.3
DEA1-KEEP-01	R	100.00	3.00	SN	RES-DEA1	21886	J03-KEEP	15.05.2015 - 08:12:43.6
DEA1-KEEP-01	R	100.00	3.00	SN	RES-DEA1	21887	J03-KEEP	15.05.2015 - 13:12:44.1
DEA1-KEEP-01	R	100.00	3.00	SN	RES-DEA1	21888	J03-KEEP	16.05.2015 - 08:12:44.6
DEA1-KEEP-01	R	100.00	3.00	SN	RES-DEA1	21889	J03-KEEP	16.05.2015 - 13:12:52.8
DEA1-KEEP-01	R	100.00	3.00	SN	RES-DEA1	21890	J03-KEEP	17.05.2015 - 08:12:43.4
DEA1-KEEP-01	R	100.00	3.00	SN	RES-DEA1	21891	J03-KEEP	17.05.2015 - 13:12:43.1
DEA1-KEEP-01	R	100.00	3.00	SN	RES-DEA1	21892	J03-KEEP	18.05.2015 - 08:12:38.1
DEA1-KEEP-01	R	100.00	3.00	SN	RES-DEA1	21893	J03-KEEP	18.05.2015 - 13:12:46.9

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

You can use the batch command client of Entire Systems Management to generate a report in batch mode: see [Commands for the Batch Command Client](#) in the section *Using Entire Operations in Batch Mode*.

XVII

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

66

Using an API Routine

- API Usage Rules and Restrictions 724
- Invoking the API Routines 725
- Example of Use of API Routines 725

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)

67

List of Available Entire Operations API Routines

▪ EORUCB1N: Check Use of BS2000 User IDs	728
▪ NOPFB2-N: Generate SYSOUT File Names for BS2000	728
▪ NOPMLA1N: Start and Stop the Monitor Activity Log	731
▪ NOPU--1N: Read Network Available to a Specific User	732
▪ NOPU--2N: Return all Usable Symbol Tables for a Network	733
▪ NOPU--3N: Get Correlation ID for an Activated Network	734
▪ NOPU--4N: Store New Event in Entire Operations System File	734
▪ NOPUAC5N: Activate Job Networks or Jobs	735
▪ NOPUAS1N: Retrieve Numbers of Active Jobs in Defined Status Ranges	738
▪ NOPUCN3N: Access Entire Operations Conditions	740
▪ NOPUCS1N: Access Calendars and Schedules	741
▪ NOPUJ14N: Import Operating System Jobs into the Active Queue	745
▪ NOPUJS2N: Job Schedule Inquiry and Modification	746
▪ NOPULW9N: Write Messages to System Automation Tools Log	748
▪ NOPUMI1N: Set/Reset Text Milestones in Master and Active Jobs	749
▪ NOPUMT3N: Expand Message Texts	750
▪ NOPUNI1N: Invalidate Entire System Server Node Table Entries	752
▪ NOPUNX1N: Entire System Server Calls to Access UNIX and Windows Files	752
▪ NOPURE2N: Handle Resource Allocations	760
▪ NOPURS1N: Access Entire Operations Master Resources	763
▪ NOPUSN2N: Inquire Calling Job or Called Network for Subnetworks	765
▪ NOPUSP3N: Display Long Texts for Symbol Prompting	766
▪ NOPUST3N: Inquire Network and Job Status, Symbol Table	768
▪ NOPUSY6N: Access Entire Operations Symbols	772
▪ NOPUVI2N: Obtain Entire Operations Version Information	777
▪ NOPUXD1N: Maintain End-of-Job User Exits for a Network	778
▪ NOPUXI0N: Add Input Condition to an Activated Job	779

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: ■ 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: ■ must be an existing run number of the job network. Function X: ■ If 0, the master network is tested. Otherwise, an active network with this run number is tested.
		out Functions A, H: ■ 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:

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. 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/net-

work the P-RC value 3 (Object used in definitions) is returned. Calendars for past years will be deleted without reference check.

- 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 run number must be defined with the special type R to prevent automatic activation.	

Parameter	Format	Use
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.
P-RC	N3	out	Return code

Parameter	Format	Use	
			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: YYYYMMDDHIISS
P-LATEST-START	A14	in/out	Latest possible start time. Format: YYYYMMDDHIISS
P-DEADLINE	A14	in/out	Deadline. Format: YYYYMMDDHIISS

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"> <tr> <td>NOP</td> <td>Writes to Entire Operations log (default).</td> </tr> </table>
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:
		■ 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.

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
		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 <code>USER</code> and <code>TIME</code> 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 <code>USER</code> and <code>TIME</code> 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.
P-UPDATE-DATE	A8	out Date of last updates. Format: YYYYMMDD.

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.

XVIII

User Exits

68

User Exits

▪ What User Exits Can Do	784
▪ Global User Exits	785
▪ Front-end User Exits	785
▪ Common User Exit Parameter Data Area NOPXPL-A	785
▪ Generation of Dynamic JCL and SYSOUT File Names (BS2000 only)	791
▪ User Exits for Resource Master Determination	792
▪ User Exits for Setting Input Conditions	792
▪ User Exits for End-of-Job Checking and Actions	792
▪ User Exits for Symbol Functions	794
▪ User Exits for Validation Checks of Symbol Values	797

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 <i>Defining End-of-Job Action User Exit</i> in the section <i>End-of-Job Checking and Actions</i> .
EJC	End-of-Job Checking See also <i>Editing End-of-Job Checking or Action User Exits</i> in the section <i>End-of-Job Checking and Actions</i> .
FSB	SYSOUT File Name Generation (BS2000) See also <i>NOPFB2-N: Generate SYSOUT File Names for BS2000</i> in the section <i>API Routines</i> .
ICO	Input Condition Value Determination See also <i>Input Condition: User Exit</i> in the section <i>Job Maintenance</i> .
MAC	Dynamic JCL Generation See also <i>Dynamic JCL Generation (JCL Location MAC)</i> in the section <i>System Overview</i> .
NAT	Natural Program Standalone under Entire Operations (NAT-type Job) See also <i>Job Types</i> in the section <i>System Overview</i> .
NVN	Network version name, syntax check See also <i>Global Exit for Version Names</i> in the <i>Administration</i> documentation.
RMD	Resource Master Determination Exit See also <i>Resource Master Determination Exit</i> in the <i>Administration</i> documentation.
SFX	Symbol Function Exit See <i>User Exits for Symbol Functions</i> .

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 --

```


XIX

Passing Files to Entire Output Management

69

Passing Files to Entire Output Management

- Basic Requirements for File Transfers to Entire Output Management 806
- Listing Files Defined for Entire Output Management 807
- Adding and Modifying File Definitions 809
- Deleting File Definitions 813
- Handing Over SYSOUT and Files to Entire Output Management 814

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 Settings** described in *Defaults for Other Settings* in the *Administration* documentation.
- The Entire Operations default setting **Copy SYSOUT File before passing it to NOM** described in *Defaults for Other Settings* 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 **EOJ Checking** page shown in the following example:

OGC - Maintenance Job Master JOB-01 [EXAMPLE,B60-FLOW]

Job name: Use symbol as execution node
 Execution node:

Description:

Job type: Special type:

Scheduling Parameters | User Defined Log Data | Long Description | OS Specials
 Main | Resources | Input Conditions | **EOJ Checking** | JCL Definition

Action	Step	will be checked for	means
		Occurrence of String 'INVALID RE	n.ok
T		Additional Actions for Job-ok	ok
AC		All Checks ok	
		Any Check not ok	

Add
 Modify
 Delete
 Edit

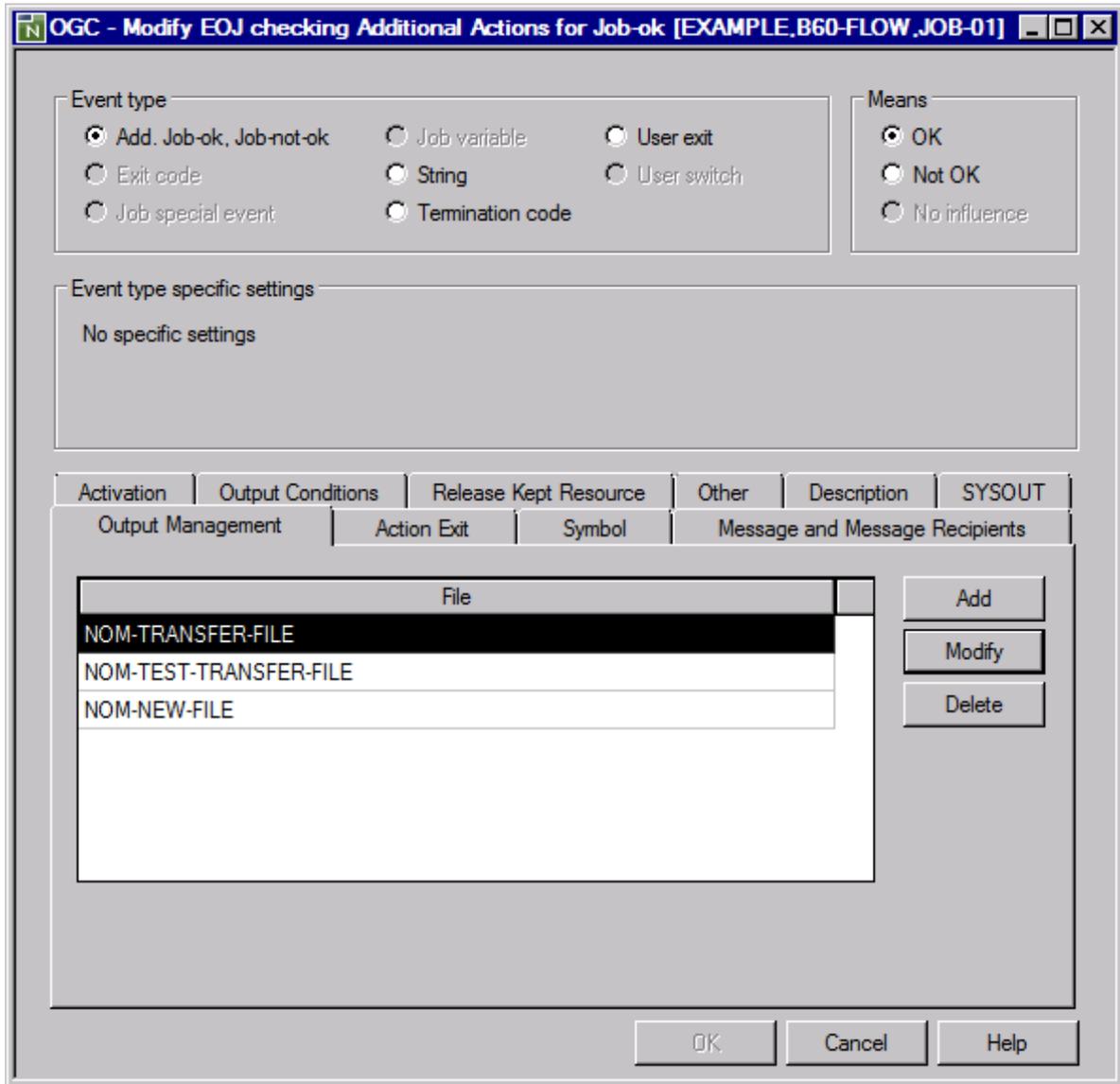
Apply OK Cancel Help

The letter **T** in the **Action** column indicates whether an EOJ action for Entire Output Management exists.

- 2 Select the table row that contains the letter **T** and choose **Modify**.

An **EOJ checking** window similar to the example below opens.

- 3 Open the tabbed page **Output Management** shown in the following example:



The page contains a list of all files defined as EOJ action for transfer to Entire Output Management.

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 **Add. Job-ok**, **Job-not-ok**, and define the extra files for these events.

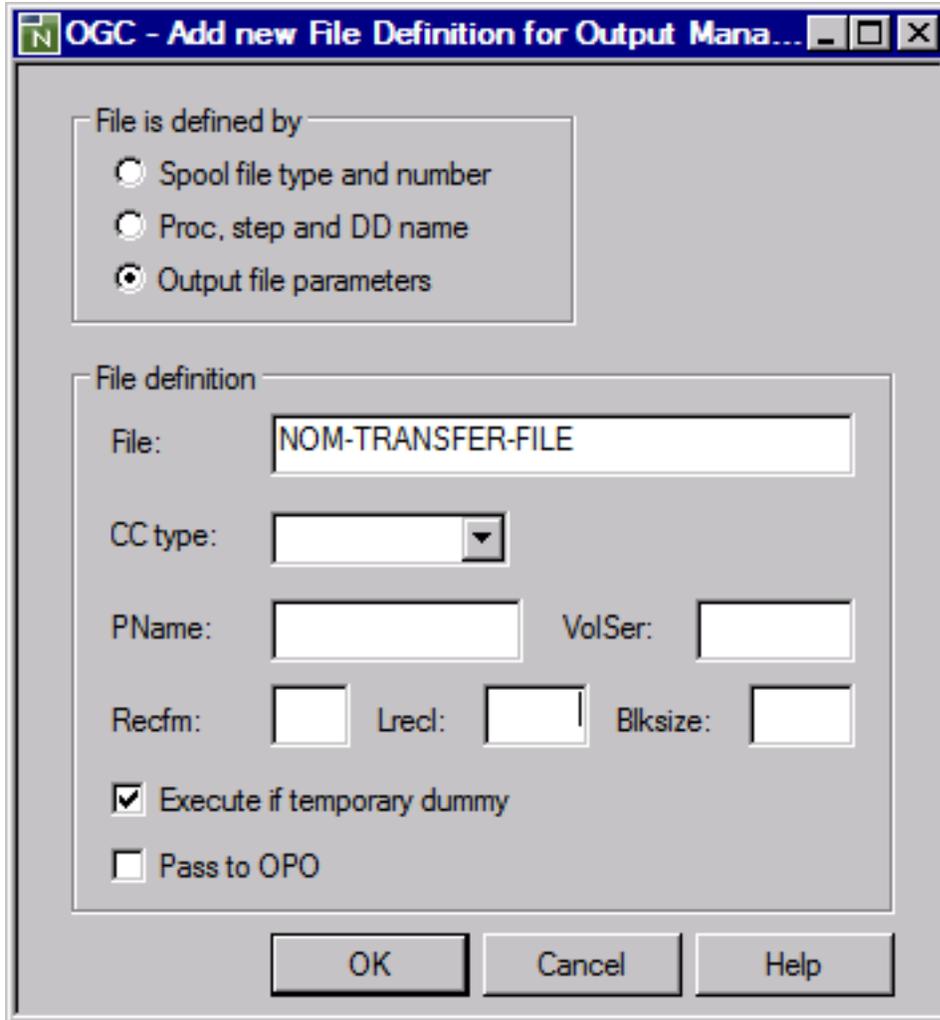
➤ To add or modify a file definition

- 1 In a **Maintenance Job Master** window, open the tabbed page **EOJ Checking**. (See also *Adding and Modifying Events and Actions* in the section *End-of-Job Checking and Actions*.)
- 2 Choose **Add** for a new EOJ action, or select an EOJ action defined for Entire Output Management (indicated by the letter **T** in the **Action** column) from the window table and choose **Modify**.

An **EOJ checking window** opens.

- 3 Open the **Output Management page**.
- 4 Choose **Add** for a new file definition, or select an existing definition and choose **Modify** for a file definition listed in the window.

A **File Definition for Output Management** window similar to the example below opens:



The fields available in the window depend on the option selected in the **File is defined by** section:

- For **Spool file type and number** and **Proc. step and DD name**, the fields are described in [Fields: Spool File Definition](#).
- For **Output file parameters**, the fields are described in [Fields: Output File Definition](#).

5 Make your definitions and choose **OK**.

The new file is assigned to Entire Output Management.

- [Fields: Spool File Definition](#)
- [Fields: Output File Definition](#)

- [Carriage Control Type \(CC Type\)](#)

Fields: Spool File Definition

The fields required to define a file of the job SYSOUT to be handled are provided in the [File Definition for Output Management](#) window, if you select **Spool file type and number** and **Proc. step and DD name**.

Field	Description
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 if you select **Output file parameters** in the [File Definition for Output Management](#) window.

Field	Description
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) .

Field	Description
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. 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.
Recfm	
Lrecl	
Blksize	
Execute if temporary Dummy	If this check box is selected (this is the default), the end-of-job action is performed even if the job executed as a temporary dummy. If this check box is not selected, the action is not performed.
Pass to OPO	Relevant for execution nodes on UNIX and Windows only. If this check box is selected, the file is passed to the Open Print Option (OPO). In this case, you cannot use wildcards. 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.
	ASA code	ASA code
	Machine code	Machine code
	EBCDIC	BS2000 EBCDIC carriage control characters
z/VSE		ASA code
	MMachine code	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 EOJ action

- 1 From a **Maintenance Job Master** window, open the tabbed page **EOJ Checking**. (See also *Adding and Modifying Events and Actions* in the section *End-of-Job Checking and Actions*.)
- 2 Select an EOJ action defined for Entire Output Management (indicated by a **T** in the **Action** column) and choose **Delete**.

A confirmation window opens.

- 3 Choose **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 From a **Maintenance Job Master** window, open the tabbed page **EOJ Checking**. (See also *Adding and Modifying Events and Actions* in the section *End-of-Job Checking and Actions*.)

- 2 Select an EOJ action defined for Entire Output Management (indicated by a **T** in the **Action** column) and choose **Add** or **Modify**.

An **EOJ checking** window opens.

- 3 Open the tabbed page **Output Management**.
- 4 Select the file you want to delete and choose **Delete**.

A confirmation window opens.

- 5 Choose **Yes** to confirm the deletion.

The selected file is removed from the tabbed page **Output Management**.

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 *Defaults for Network Options* 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

- From the [EOJ Checking](#) page, open the tabbed page **SYSOUT** and **Pass to EOM** from the **SYSOUT action** list box. 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 *Defaults for Network Options* 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.rrrr.filename.ext
```

File Name	Meaning
<i>yymmdd</i>	Year, month, day.
<i>hhisst</i>	Time, incl. 1/10 sec.
<i>rrrr</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>rrrr</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.

XX

Special Monitor Features and Batch Jobs

70 Special Monitor Features and Batch Jobs

- Monitor Start Network 822
- Day Start Network 822
- Common Start Network Considerations 823
- Cleanup in Batch Mode 823

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
```

