Concepts and Facilities

Version 6.0.1

October 2026

ADABAS & NATURAL

This document applies to Version 6.0.1 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 © 2021-2026 Software GmbH, Darmstadt, Germany and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software GmbH product names are either trademarks or registered trademarks of Software GmbH 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 GmbH and/or its subsidiaries is located at https://softwareag.com/licenses.

Use of this software is subject to adherence to Software GmbH's licensing conditions and terms. These terms are part of the product documentation, located at https://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 GmbH Products / Copyright and Trademark Notices of Software GmbH Products". These documents are part of the product documentation, located at https://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 GmbH.

Document ID: NOW-ONOPCONCEPTS-601-20261023

Table of Contents

Preface v
1 About this Documentation 1
Document Conventions 2
Online Information and Support 2
Data Protection 3
2 Introducing Entire Operations 5
More than a Scheduler 7
Summary of Benefits 11
User Interfaces 11
3 Overview of Entire Operations Components 15
4 Entire Operations Maintenance Objects 17
Entire Operations Object Relationship 18
Users and Owners 21
Job Networks 23
Jobs 25
JCL 28
Logical Conditions 29
Events and Actions: End-of-Job Checking 33
Mailboxes, Message Sending 35
Resources 38
Calendars 38
Schedules 40
Symbol Tables and Symbols 41
Object Versioning 42
Operating System Server Nodes 42
5 Entire Operations Monitor 45
Monitor Subtasks 46
Running the Monitor as a Batch Task 47
6 Master Database and Active Database 49
Master Database 50
Active Database 51
Run Numbers 51
Job Process Control 52
7 Operating System 55
8 Tools and Utilities 57
Administration Functions 58
Object-Specific Maintenance Functions 58
Process Control and Monitoring Functions 59
Entire Operations Utilities 60
Import and Export Functions 61
9 APIs and User Exits 63

Preface

Introducing Entire Operations	Gives an overview of Entire Operations features provided to define, control and schedule job networks.
Overview of Entire Operations Components	Provides a graphical overview of main Entire Operations components, such as network maintenance objects, the Monitor, databases, tools and utilities.
Entire Operations Maintenance Objects	Describes Entire Operations maintenance objects, such as owners, networks, jobs, resources schedules and symbol tables.
Entire Operations Monitor	Describes the Monitor, the major process component of Entire Operations.
Operating System	Refers to operating systems on which Entire Operations can process job networks.
Master Database and Active Database	Describes where Entire Operations stores network definitions and keeps data of activated networks.
Tools and Utilities	Lists the Entire Operations tools and utilities available for system administration, import/export operations, and controlling and monitoring network processing.
APIs and User Exits	Refers to accessing Entire Operations from other applications and perform Entire Operations functions.

1 About this Documentation

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

Document Conventions

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

Online Information and Support

Product Documentation

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

Product Training

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

Tech Community

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

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

Product Support

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

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

Data Protection

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

2 Introducing Entire Operations

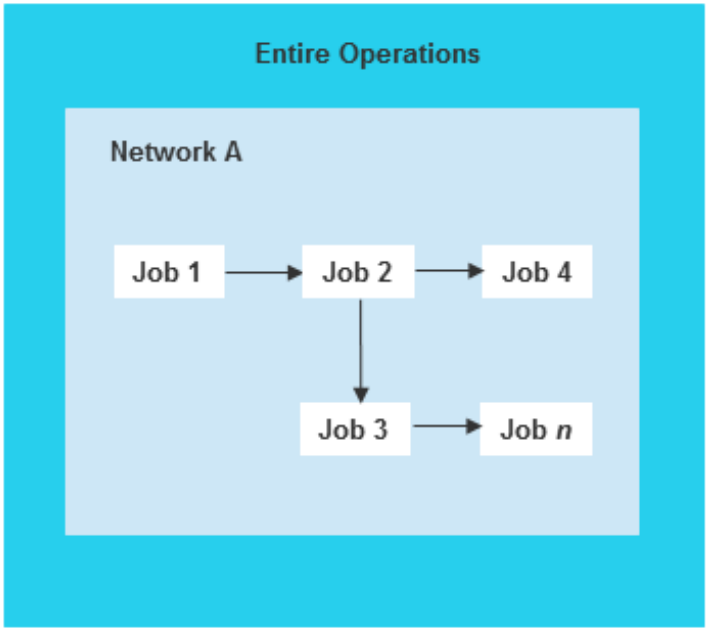
- More than a Scheduler 7
- Summary of Benefits 11
- User Interfaces 11

Entire Operations (product code: NOP) is Software GmbH's system for the automated control and scheduling of job networks. It provides all the functions required to define any type of background processing.

Entire Operations provides comfortable **character and graphical user interfaces** that guide the user with task-oriented features.

Entire Operations requires no modification to underlying operating system or to any of the sub-systems installed at your site. Existing JCL can be put under Entire Operations control unchanged, allowing a smooth transition of your existing production control methods to automatic scheduling.

Entire Operations schedules and controls job networks:



Standard security packages such as RACF, ACF2, CA-TOP SECRET or SECOS are supported, allowing Entire Operations to honor existing security profiles.

For the execution of batch jobs and scripts, Entire Operations uses clearly defined interfaces to installed spooling systems or equivalent operating system utilities.

More than a Scheduler

In addition to the core functionality of Entire Operations described in the previous sections, Entire Operations offers the following features:

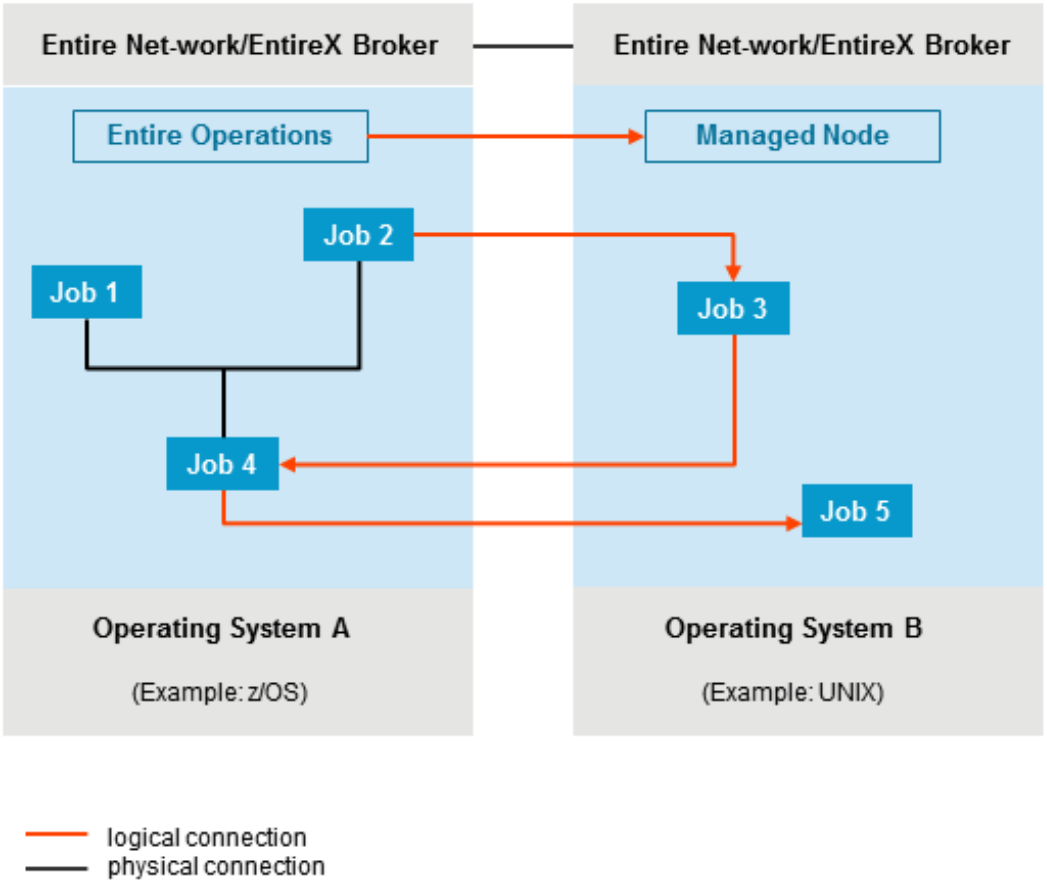
- Control across Operating Systems
- Entire Operations in a Multiple Operating System Environment
- Flexible Process Control
- Integration of Other Applications
- Integration of People in Automated Operations

Control across Operating Systems

- **Mainframe environments:**
  Entire Operations can be installed on z/OS, running with TP monitor/online systems, such as Com-plete, CICS, TSO, IMS TM, *open*UTM or TIAM.
- **UNIX:**
  Entire Operations can also be installed on UNIX platforms. It controls production processes on UNIX systems , in conjunction with a mainframe or UNIX environment. You can manage your production from any operating system on which Entire Operations can run or use Entire Operations to control several UNIX and Windows machines.
- **Windows:**
  Entire Operations controls the production processes under Windows.

Background processes running in heterogeneous operating system environments can also be controlled and monitored by Entire Operations if the computers are interlinked with Software GmbH's communication service Entire Net-Work. In such a distributed environment, job networks can consist of processing steps that execute on different operating systems.

Entire Operations in a Multiple Operating System Environment



While the Entire Operations **Monitor** program handles the distribution and decentralized execution of processing steps, this type of distributed processing can still be monitored and controlled centrally from a single point of management.

Flexible Process Control

Entire Operations allows you to define variable processing steps. For example, executable job control can be built dynamically according to current system conditions such as disk space, content of system queues and availability of specific files. In other words, job control can "read" a current situation and adapt accordingly.

Natural programs and user exits allow you to map any conceivable decision criterion and put all relevant data at your disposal, thanks to Natural interfaces to all commonly used data management and operating systems - even and especially in heterogeneous computer networks.

It is then no longer necessary to think of all possible problem situations and spend time in defining remedial action before processing starts. All you need to do is specify strategies (= programs) that recognize and resolve problems as they occur.

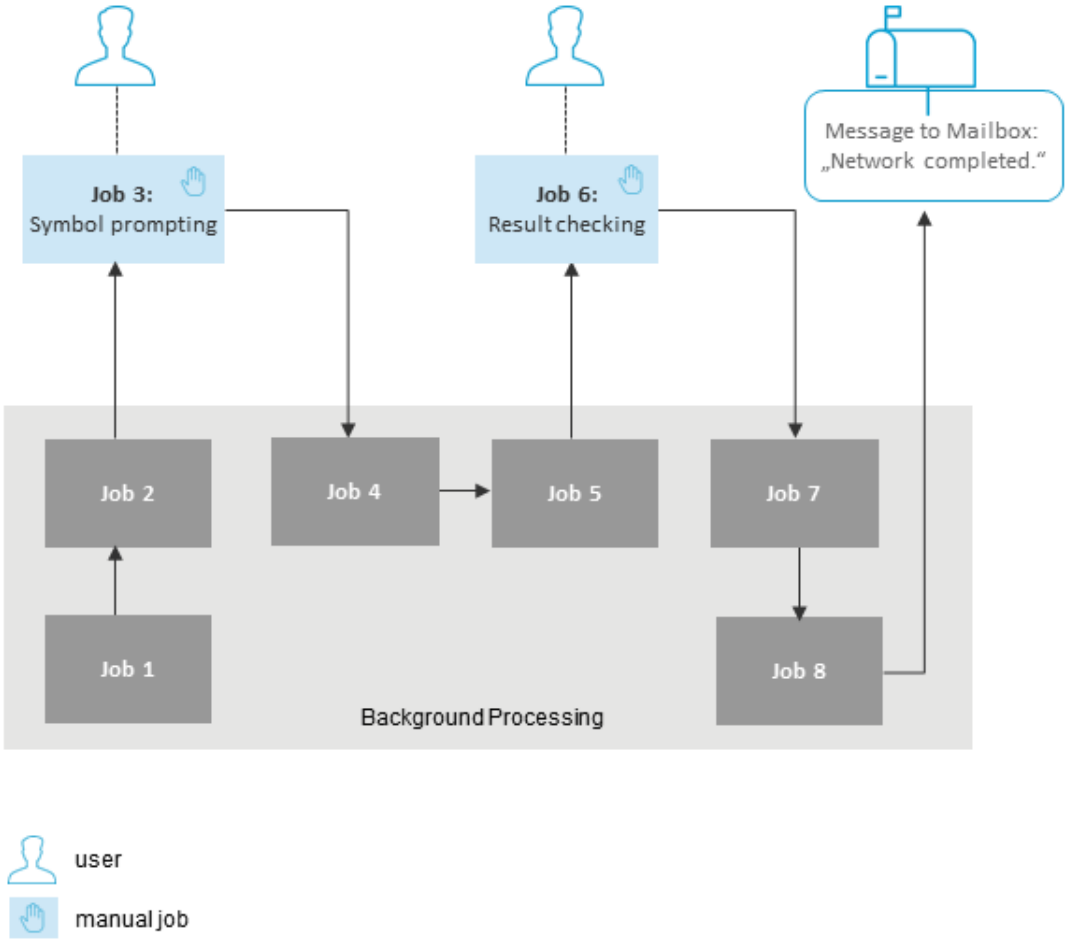
Integration of Other Applications

Entire Operations provides the option to integrate other applications. If any other products from the Entire Systems Management line are installed, Entire Operations recognizes and includes them. These are, for example:

- Entire Output Management
- Entire Event Management
- Entire System Server
- Natural ISPF

Integration of People in Automated Operations

Aside from this technical integration capability, Entire Operations can also integrate people by providing its own **mailbox concept** to combine background processing with user interaction:



At specified times during background processing, messages or prompts can be sent to such mailboxes. This has the twin effect of halting processing and informing all users with access to the mailbox of the situation. These users can react as the situation demands, for example, check a result or enter a symbol when prompted to continue processing.

This mechanism enables selected employees to provide input relevant to their department, while background processing as a whole remains under central control.

Summary of Benefits

The basic advantages of using Entire Operations to automate your data processing tasks can be summarized as follows:

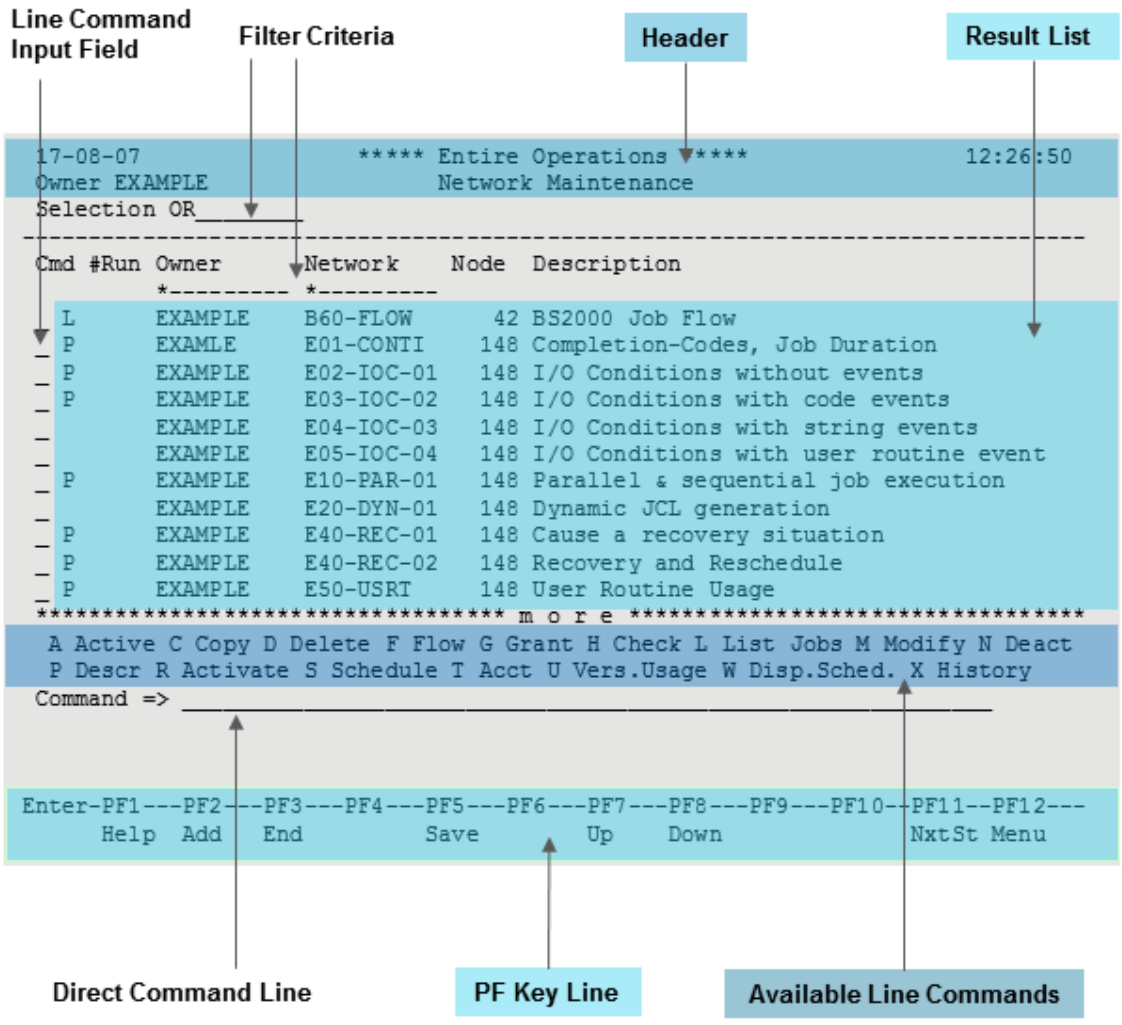
- Transparent support of several computer nodes, even in heterogeneous environments comprised of z/OS, BS2000, UNIX and Windows operating systems;
- Available in many mainframe and UNIX environments supported by Natural, for example: Com-plete, CICS, TSO, IMS TM, *open*UTM and TIAM;
- Available in English and German;
- Existing JCL runs unchanged under Entire Operations;
- No modifications are necessary to the operating system;
- Use of dynamically-built JCL or scripts, thus integrating the latest information from the operating system or any available database at execution time;
- Integration of online users into batch network processing through the concept of mailboxes;
- Open interface to user applications: information from Entire Operations can be included in any business application and users can provide input data necessary for daily or future production runs.

User Interfaces

Entire Operations provides both a menu-driven character user interface (CUI) and a graphical user interface (GUI). The Entire Operations GUI Client is used to perform Entire Operations functions on a mainframe or a UNIX platform from the PC.

Entire Operations (product code: NOP)

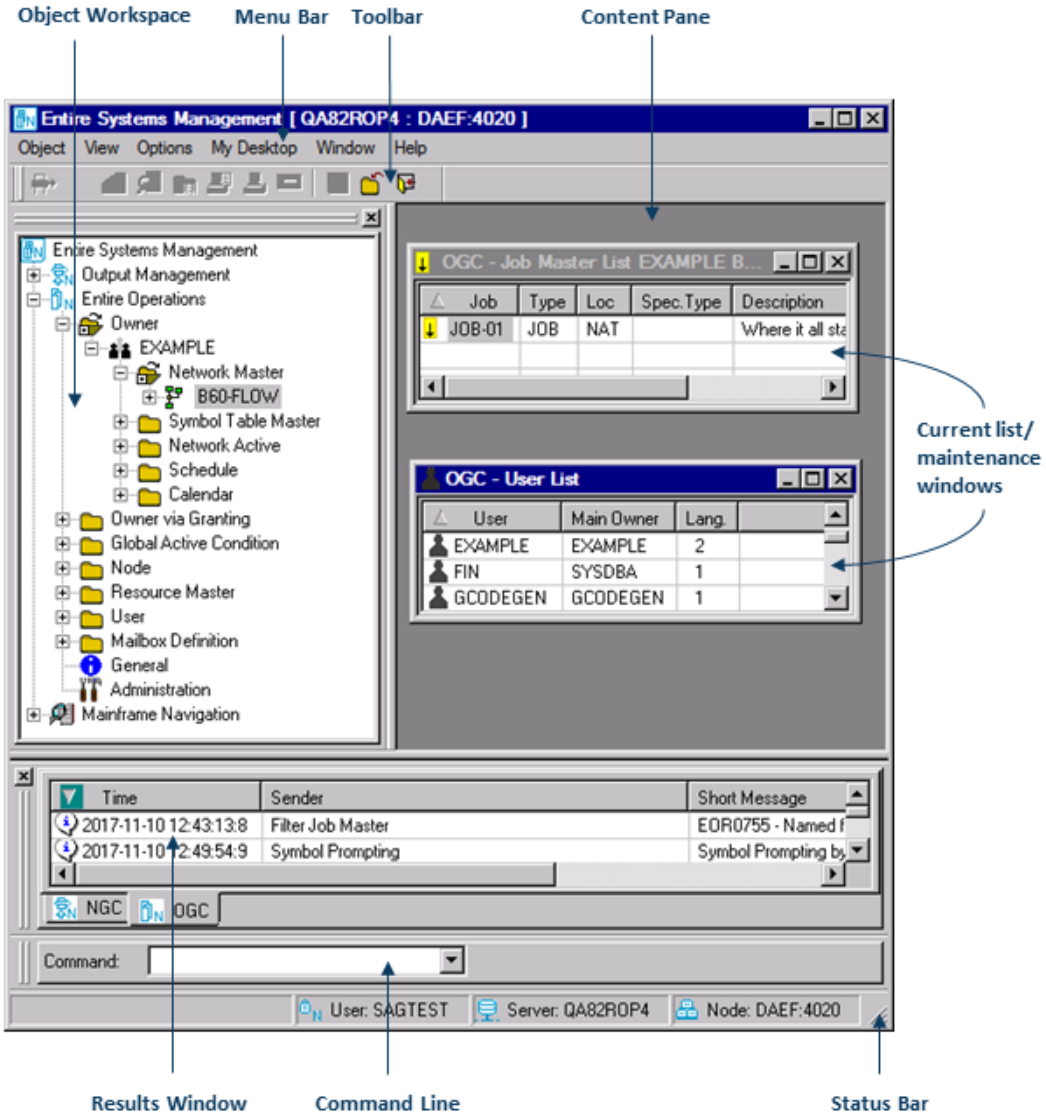
The main elements that are typical for an Entire Operations CUI screen are indicated in the following example illustration of a **Network Maintenance** screen:



For detailed information, see *Main Screen Elements and Functions* in the *User's Guide*.

Entire Operations GUI Client (product code: OGC)

Entire Operations GUI Client provides a graphical user interface with a main application window:



For detailed information, see the section *Elements of the Main Application Window* in the *User's Guide*.

This section covers the following topics:

- English and German

- Entire Operations Online Help

English and German

Entire Operations is available in English and German, including all items in the user interfaces, help information and error messages.

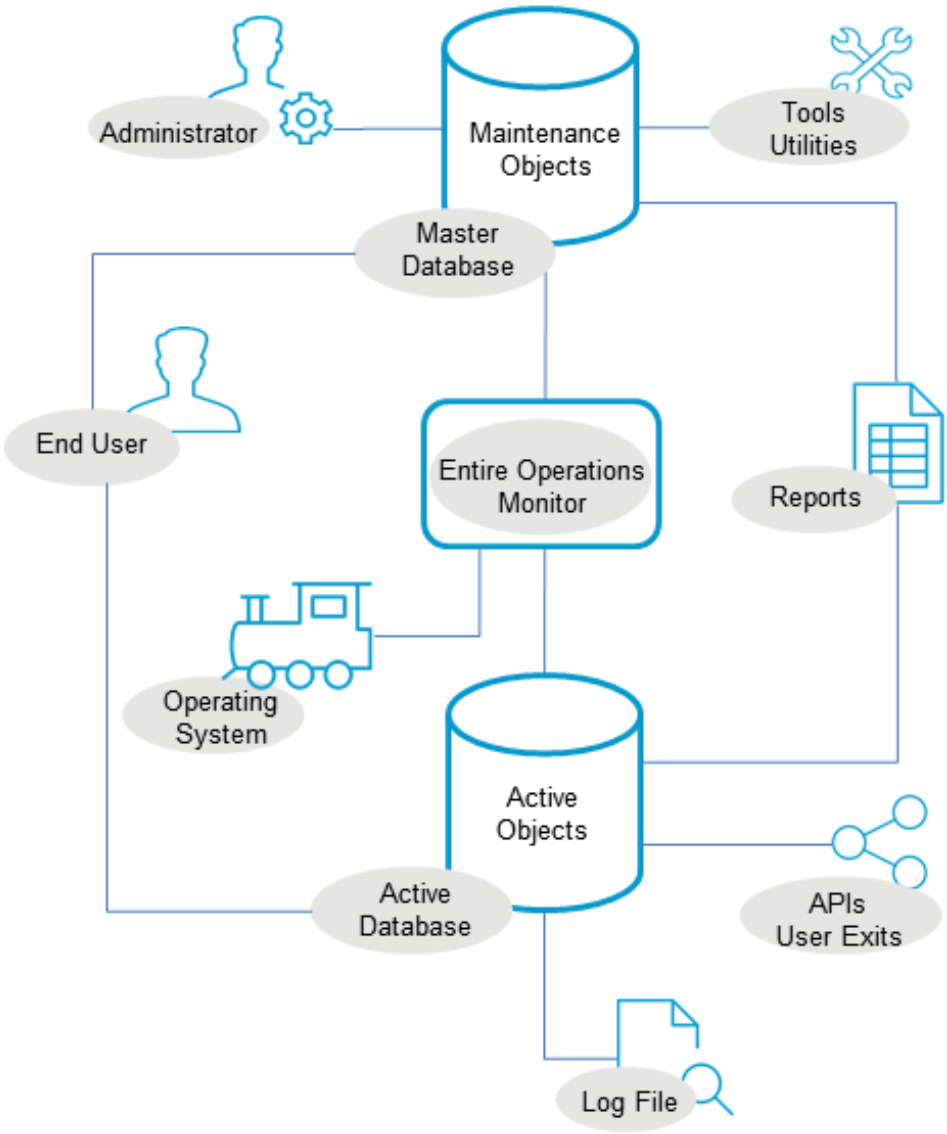
Each user can set the language individually.

Entire Operations Online Help

For detailed information, see *Online Help and System Messages* in the *User's Guide*.

3 Overview of Entire Operations Components

Entire Operations consists of the following components:



Note: You can click on an oval (gray) symbol in the diagram above for more information on the respective component.

All components are described in the remainder of this document.

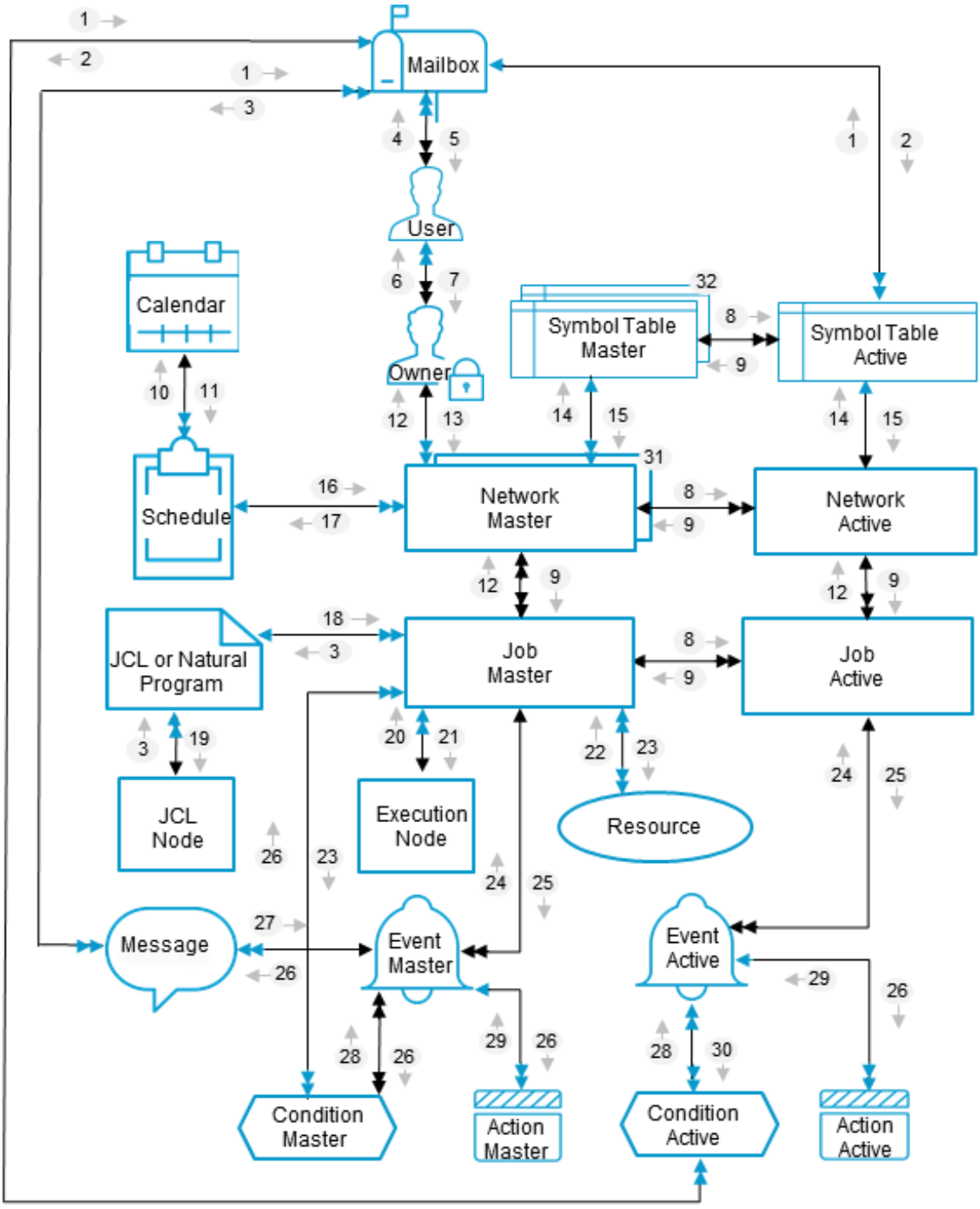
4 Entire Operations Maintenance Objects

- Entire Operations Object Relationship 18
- Users and Owners 21
- Job Networks 23
- Jobs 25
- JCL 28
- Logical Conditions 29
- Events and Actions: End-of-Job Checking 33
- Mailboxes, Message Sending 35
- Resources 38
- Calendars 38
- Schedules 40
- Symbol Tables and Symbols 41
- Object Versioning 42
- Operating System Server Nodes 42

The objects that can be maintained in Entire Operations to process and control job networks are described in this section.

Entire Operations Object Relationship

The following diagram shows all Entire Operations maintenance objects and indicates their relationship and dependencies and the conditions required to maintain and run a job network:


Mailbox
User
Calendar
Owner
Schedule
Symbol Table Master
Symbol Table Active
Network Master
Network Active
JCL or Natural Program
Job Master
Job Active
JCL Node
Execution Node
Resource
Message
Event Master
Event Active
Condition Master
Action Master
Condition Active
Action Active

Legend:

▶	must be only one
▶	none or at most one
▶▶	must be at least one, can be more than one
▶▶	none or one or many
→	direction indicator
1	is sent to
2	contains prompting for
3	contains
4	uses or can use
5	contains messages for
6	authorizes
7	can choose
8	lends properties to
9	is composed of
10	is based on
11	is the basis of
12	belongs to
13	owns
14	obtains parameters from
15	contains values for
16	schedules
17	is scheduled based on
18	is executed as
19	resides on
20	is platform for
21	runs on
22	is preliminary condition for

23	depends on or can depend on
24	determines result of
25	is checked for
26	triggers
27	is set according to
28	is set or reset according to
29	is performed according to
30	sets
31	A network master object can have several versions (see *Object Versioning*).
32	A symbol table master object can have several versions (see *Object Versioning*).



Note: You can click on a symbol in the diagram above for more information on the respective object.

Users and Owners

Each Entire Operations user is assigned a user ID for the purpose of security, profiling, message switching and logging.

Each user ID is associated with a user profile which contains authorizations. A user profile can be modified by an authorized user (e.g. the system administrator).

This section covers the following topic:

- Owners
- Example of Owner Assignment

Related Topics:

- *User Maintenance - Administration* documentation
- *Using Owners - User's Guide*

Owners

Owners enhance security and ease of use by grouping user IDs and associating job networks to these owners.

The concept of owners 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 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.

A user can be authorized to use several owner names: switching owner names means selecting another group of job networks, which can then be maintained.

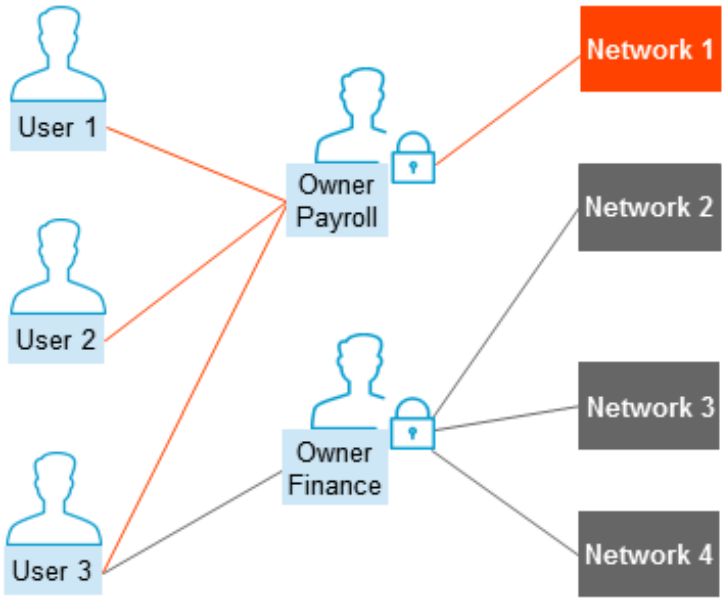
Note: In special cases, a user can be authorized to access networks belonging to other owners. The owner SYSDBA is authorized to access the networks of all owners. See also *Granting Definition: Authorizing Other Users or Owners to Access a Network* in the *User's Guide*.

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 title of most system windows.

Example of Owner Assignment

The following figure provides an example of the relationship between users, owners and job networks:

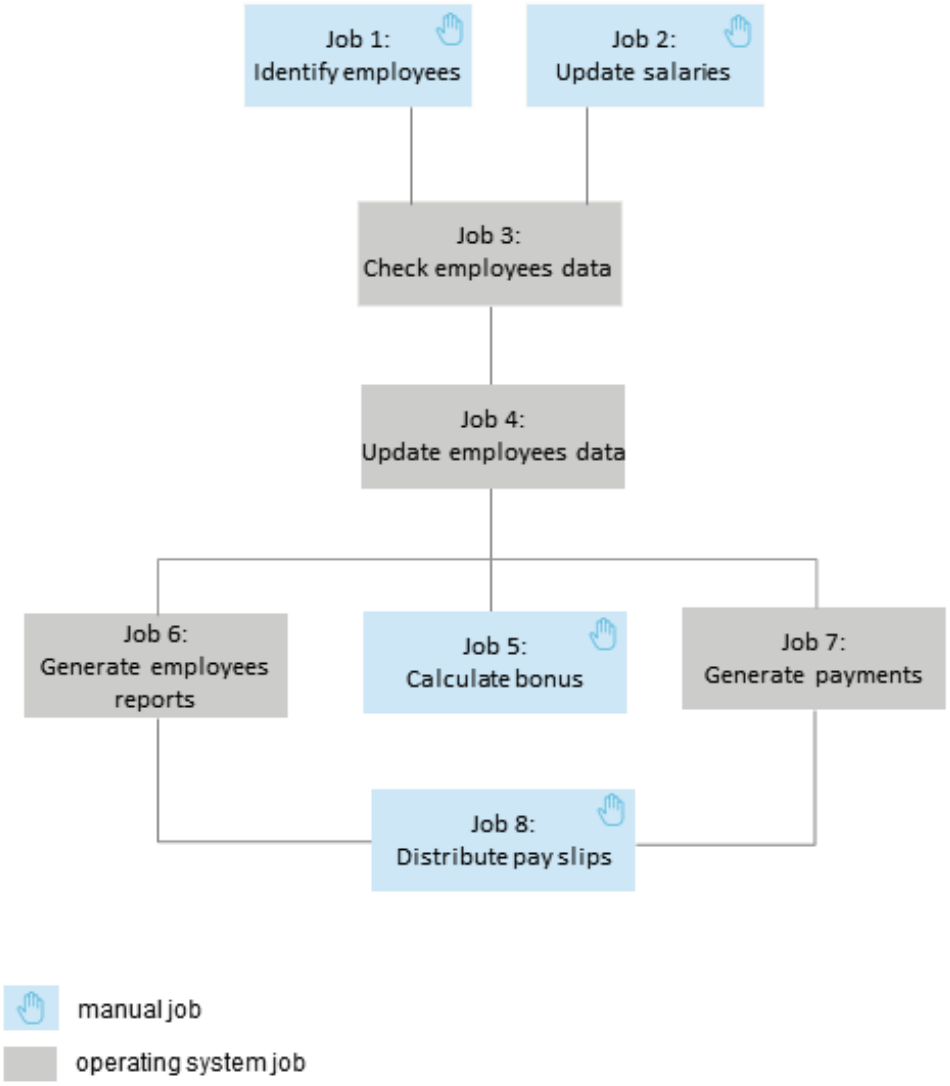


All three users may maintain networks belonging to owner Payroll, while User 3 is also authorized to access networks belonging to owner Finance.

Job Networks

In Entire Operations, a job network is a group of jobs, tasks, scripts or processes that may or may not be interrelated and which are scheduled for work according to a job network schedule table. Thus, a job network can represent any unit of business work in production work flow.

You can even integrate into these job networks manual tasks, which are to be performed at fixed times by data center personnel. The following figure illustrates an example of a job network for the automatic generation of pay slips and summary reports as it might be implemented in a payroll department:



In the normal case, a job network consists of a chain of jobs linked together by certain dependencies (e.g. in the simplest case: If Job 1 finishes OK, start Job 2). These dependencies are expressed by logical conditions.

A network is the smallest unit that can be activated automatically by Entire Operations. By using calendars, networks can automatically be scheduled for activation on workdays only. Apart from this, you can manually activate networks at any time.

Several active copies (or activations) of a network can work in parallel, since Entire Operations identifies each copy uniquely by its run number, which is automatically assigned to each network at activation time.

You can use a job to define subnetworks within a main network. This allows you to build nested networks.

This section covers the following topics:

- Network Master Definition
- Network Scheduling

Related Topics:

- *Defining a Subnetwork - User's Guide*
- *Network Maintenance - User's Guide*
- *Defining a Network Schedule - User's Guide*
- *Schedule Maintenance - User's Guide*

Network Master Definition

Before Entire Operations can activate a job network, this job network must be defined as a network master to the **master database**. You can add a short description for easy identification when reviewing networks. You can specify default nodes for JCL and job submission on the network level. You can also specify a symbol table on the network level. Node and symbol table are then used as defaults for all jobs contained in the job network but can be overwritten for each job individually.

Network Scheduling

Scheduling a job network means defining dates and times at which it is to run. A job network is activated (copied into the active database) according to its scheduled starting time and additional schedule parameters defined for the network.

While processing a network, the Entire Operations **Monitor** checks whether all prerequisites for a job are fulfilled (time, resources, input conditions). According to the result it will automatically start the job and keep track of it.

Jobs

The job is one of the basic objects of Entire Operations. Entire Operations regards a job as a user-defined task that is performed by JCL instructions and job IDs, scripts or files as supported by the operating system, Entire Operations subnetworks or dummy jobs, or Natural programs. For all types of jobs supported by Entire Operations, see ***Job Types*** described in the following section.

This section covers the following topics:

- Job Master Definition

- Job Scheduling
- Job Types
- Job Attributes
- Jobs in a Multi-Machine Environment

Related Topics:

- ***Job Process Control***
- *Job Maintenance - User's Guide*
- *Scheduling a Job - User's Guide*
- *Schedule Maintenance - User's Guide*

Job Master Definition

A job is defined by (a logical) name, type and location (PDS member, Natural library, etc.) as a job master to the **master database**.

You can add, delete and modify jobs in the network. You can add a short description for easy identification when reviewing the jobs in the network. You can also specify, for example, a node and symbol table for job execution to override the defaults defined for the job network.

Job Scheduling

The activation of each job depends on the scheduling parameters defined for it. As in the case of job networks, you can define time frames.

For jobs, however, you can additionally specify whether a scheduled job that did not run (e.g. due to hardware problems) is to be rescheduled (i.e. the job is scheduled to run twice at the next scheduled time). You can also specify a number of days to determine how long a job can reside at most in the active queue before it is executed.

You can also specify an estimated job running time which the Entire Operations **Monitor** uses to calculate starting and end time for the job. This can prevent a job from running if a predefined deadline would be exceeded.

The job scheduling parameters also provide a facility to notify specified users if the start of job fails to meet the defined deadline starting time.

Job Types

Entire Operations supports the following types of jobs:

- Standard jobs of the operating system (z/OS and BS2000);
- Started Tasks (z/OS);
- Shell scripts of the UNIX operating system;
- BAT files and PowerShell scripts on Windows systems;
- Other scripting environments on UNIX and Windows (e.g.: Perl, Windows Scripting Host);
- Command-line oriented executables on UNIX and Windows;
- FTP jobs;
- Natural programs;
- Data file generation;
- Windows services;

In addition, for non-operating system jobs there are:

- Dummy jobs to create time windows for non-operating system jobs or Boolean connections for single conditions.
- Subnetwork jobs to define a subnetwork within a main network. This allows you to build nested networks.

For more information about job types, see the section *Available Job Types* in the *User's Guide*.

Job Attributes

Each job in the network is defined by a series of identifying attributes:

- (Logical) name;
- Job type;
- Node for JCL location;
- Node for job execution;
- Time frame;
- **End-of-Job** information;
- User ID under which job is run.

Such a job can be contained in several job networks.

Jobs in a Multi-Machine Environment

When Entire Operations is used in a multi-machine environment, the location of a job (i.e. the location of its contents) and the location of its execution node can differ: at activation time Entire Operations reads the job information from the source node and executes it on the target node.

Jobs in a network can be interlinked by using ***Logical Conditions***.

See also the *z/OS: JES2 /*ROUTE Statement* described in the *User's Guide*.

JCL

A job needs job control language (JCL) instructions to perform a task. Exceptions are dummy jobs and subnetwork jobs that are used for special purposes within Entire Operations.

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 covers the following topics:

- Dynamic JCL Generation

Related Topics:

- *Defining and Managing JCL for a Job - User's Guide*
- *Dynamic JCL Generation (JCL Location MAC) - User's Guide*

Dynamic JCL Generation

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* in the *User's Guide*). This is referred to as dynamic JCL generation and only applies to jobs with macro sources (JCL location MAC) in Entire Operations.

Logical Conditions

The use of logical conditions is the central concept of Entire Operations. Logical conditions are used to describe job or network dependencies. A logical condition can be set by any event that occurs during job processing. This event must occur before Entire Operations can proceed to the next step.

When a job network is activated, each logical condition is assigned a **run number**. This run number enables Entire Operations to distinguish between the same event that occurs during different network activations.

Logical conditions can be used in two different ways:

- As input conditions;
- As output conditions.

This section covers the following topics:

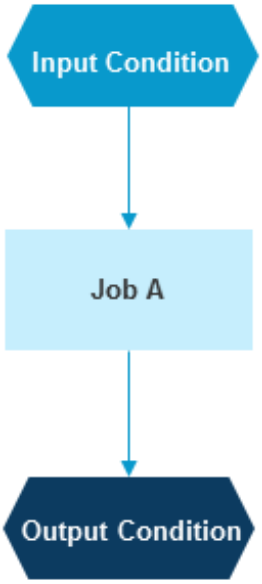
- Input and Output Conditions
- Jobs Linked by Input and Output Conditions
- Job Network with Logical Conditions

Related Topics:

- For more detailed information on conditions, see the sections *Defining and Managing Job Conditions* and *Defining and Managing End-of-Job (EOJ) Checking and Actions* in the *User's Guide*.
- For information about prerequisite checking, see the section *Prerequisite Check before Job Submission*.

Input and Output Conditions

The following figure illustrates the concept of input and output conditions in relation to a job:



All input conditions must be fulfilled before a job can be submitted (prerequisite condition). You can define any number of input conditions for a job.

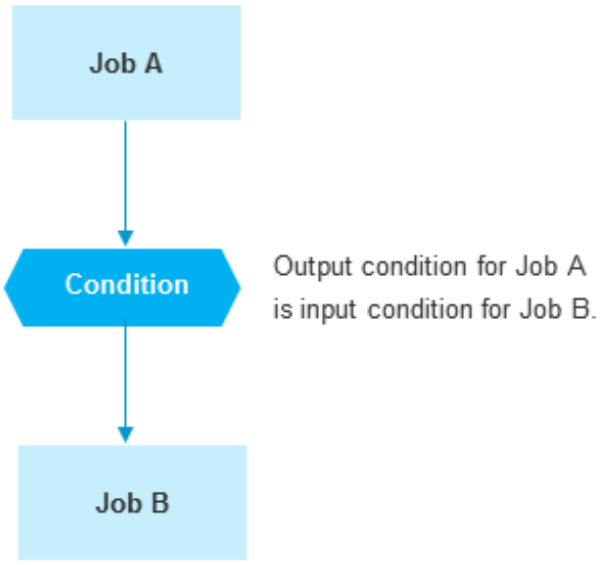
An output condition can be set or reset according to the result of predefined events (either automatically given by Entire Operations or user-defined). As part of **End-of-Job** analyses, Entire Operations checks for the occurrence of such events. Several output conditions can be set or reset for each event at the job or even job step level.

Related Topic:

Defining and Managing Job Conditions - User's Guide

Jobs Linked by Input and Output Conditions

Jobs in a job network are linked by defining an output condition of one job as an input condition for the next job, as illustrated by the figure below:

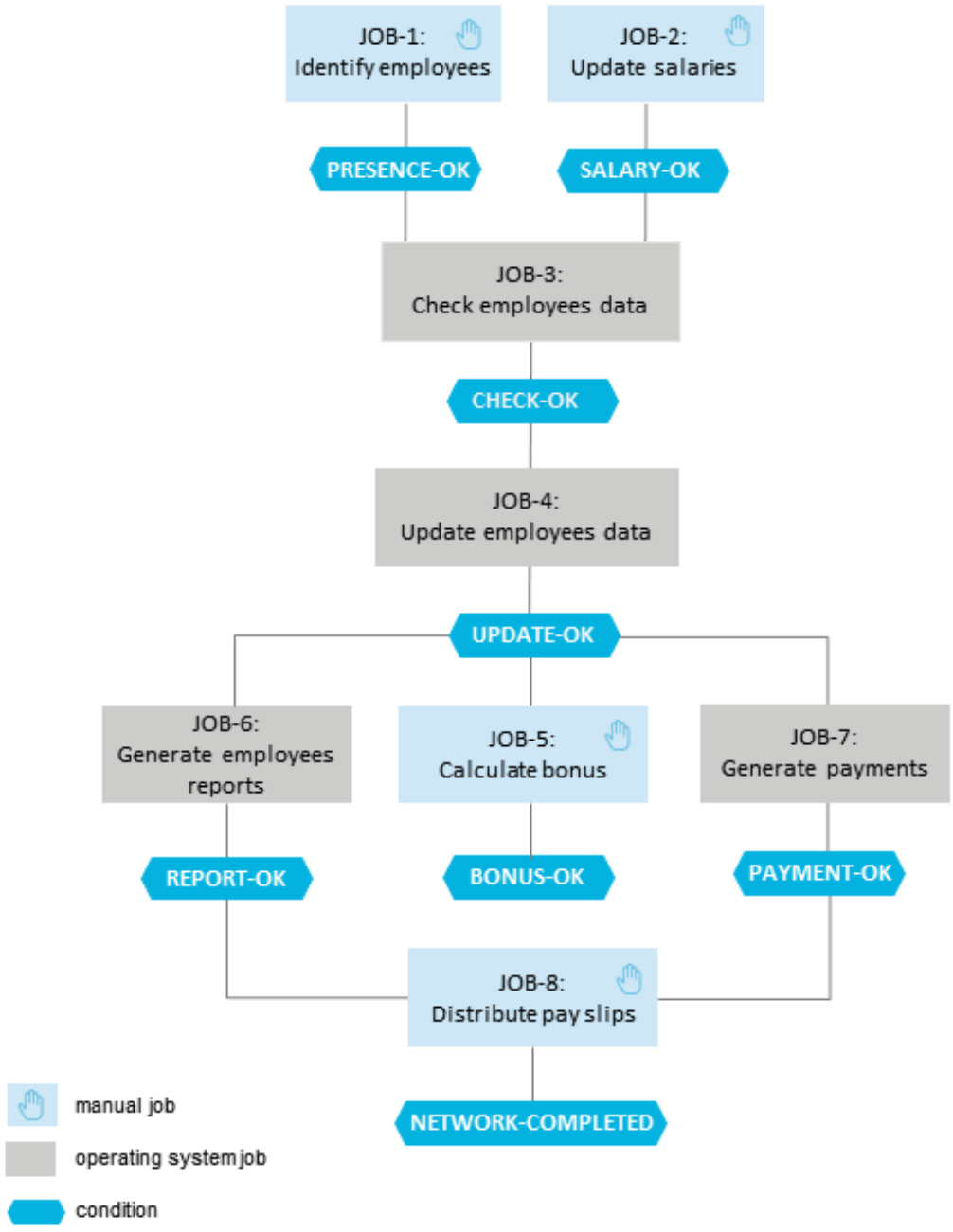


A specified event occurred as a result of Job A. This sets the condition, signaling to Entire Operations that Job B can be started.

You can also link jobs together which belong to different job networks or which are executed on different computer nodes.

Job Network with Logical Conditions

The following figure is an example of the job dependencies for the job network of a payroll department:



The following table gives an overview of the job dependencies (logical conditions) that link the jobs illustrated above:

Job Number	Input Condition	Output Condition
JOB-1	n/a	PRESENCE-OK
JOB-2	n/a	SALARY-OK
JOB-3	PRESENCE-OK	CHECK-OK
	SALARY-OK	
JOB-4	CHECK-OK	UPDATE-OK
JOB-5	UPDATE-OK	BONUS-OK
JOB-6	UPDATE-OK	REPORT-OK
JOB-7	UPDATE-OK	PAYMENT-OK
JOB-8	REPORT-OK	NETWORK-COMPLETED
	PAYMENT-OK	

For example, Entire Operations will not start JOB-7 (Generate payments) until the input condition UPDATE-OK is fulfilled (this condition is also defined as the output condition for JOB-4).

This job flow is completely independent of the operating system platforms on which the individual processing steps run.

Events and Actions: End-of-Job Checking

End-of-Job checking refers to the process of how Entire Operations recognizes the job status on job completion.

This section covers the following topics:

- End-of-Job Events
- End-of-Job Actions
- Default Checks

Related Topics:

Defining and Managing End-of-Job (EOJ) Checking and Actions - User's Guide

End-of-Job Events

When the job has just completed, Entire Operations searches for the occurrence of user-defined events. Such an event can be any of the following:

- A return code is received in a specific job step;
- A return code is received in any job step;
- A string is found in the job protocol or output;
- A Natural user exit is executed which determines the End-of-Job status by returning a certain condition code. This routine can:
  - examine the job protocol or output itself,
  - read data produced by the job,
  - perform system functions,
  - send messages.

End-of-Job Actions

You can define how Entire Operations reacts on each specified event. Such system action can consist of any of the following, for example:

- Automatically set or reset the output conditions associated with this event (see the section ***Output Conditions***);
- Send a message (see *Defining Notification Messages* in the *User's Guide*) to a specific operating system user, the system console, an Entire Operations mailbox or an e-mail address;
- Cancel or print job protocol and output;
- Perform a recovery (if a job or job step failed);
- Pass files to Entire Output Management (NOM) (see the *User's Guide*).

Default Checks

Depending on the operating system where the job was executed, Entire Operations performs some default checks to determine the job result. For z/OS systems, for example, system abends or JCL errors will automatically be detected. These default checks will be executed for each job, regardless of whether specific user-defined checks were requested for a job.

For detailed information, see the section *Operating System Dependent Defaults for Event Checking* in the *User's Guide*.

Mailboxes, Message Sending

Within Entire Operations, mailboxes serve to send network- and job-related messages and requests to users and/or groups of users. These messages can be used to inform users about the current status of the job network or to request some data needed for further execution.

The messages and requests listed for a mailbox are generated by the Entire Operations **Monitor** because of either automatically generated processing information or user-written messages and requests defined for jobs. Requests can require user interaction to continue job processing, such as setting a condition or performing symbol prompting.

Using the concept of mailboxes:

- tasks can be made dependent on logical conditions and can also set logical conditions;
- mailboxes can be assigned to these logical conditions to specify who should be informed about them.

This section covers the following topic:

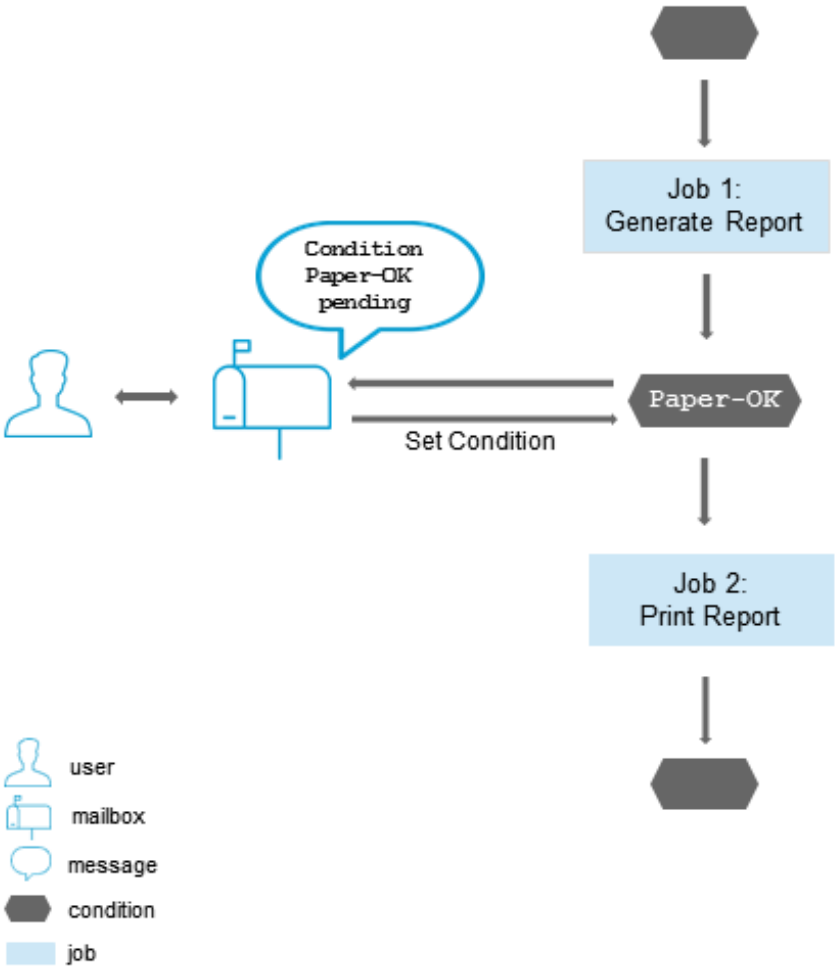
- Example Scenario - Main Mailbox Concept
- Example Scenario - Concept of Single and Multiple Mailbox Users

Related Topics:

- *Working with Mailboxes - User's Guide*
- *Mailbox Definition - Administration* documentation

Example Scenario - Main Mailbox Concept

The concept of mailboxes allows you to integrate manual actions into the job network as illustrated in the following example of a mailbox for paper supply:



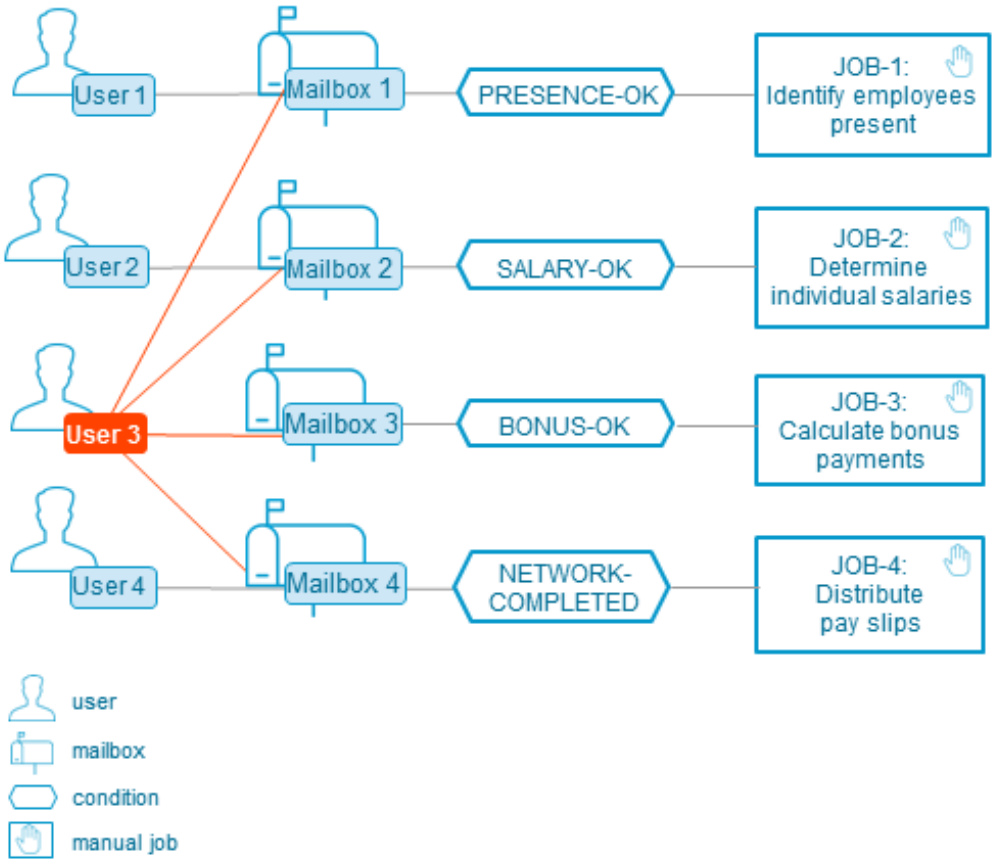
Explanations:

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).

Example Scenario - Concept of Single and Multiple Mailbox Users

The following figure illustrates the concept of mailboxes when one or more users are linked to a mailbox using the example of the payroll department job network:



Explanations:

User 1 (who could be in data collection) is notified that the condition PRESENCE-OK is not fulfilled.

He can then take the necessary steps by completing personnel attendance data and confirming this in his mailbox, when finished.

User 3 (who could be an assistant or backup of the personnel manager) is notified of any unfulfilled condition and can thus supervise the running of the whole job network. Users 3 and 4 are notified when the network has ended (NETWORK-COMPLETED) and the pay slips can be distributed.

Resources

Resources can reflect real resources, or they can be fictitious. The existence of real resources can be determined using Entire System Server features. For example, you can examine the size of free space on any available disc, the presence of any cataloged data set or the actual number of running jobs.

Entire Operations resources are only meaningful within Entire Operations and are used to further regulate the job flow. They can be used to determine the job sequence within the job flow or prevent jobs from running in parallel if all other prerequisites (e.g. time windows or logical conditions) are fulfilled. For example, if a user defines a certain quantity of a resource as a prerequisite for a specific job, then this job will not run until this amount is available.

In our example of the payroll department job network (***Job Network with Logical Conditions***), we could prevent Job 6 (Generate employees reports) and Job 7 (Generate payments) from running in parallel by defining a resource (e.g. CPU time) with an initial quantity of 100, and defining this resource for each job with a required value of 60. The jobs will then run sequentially.

For detailed information, see *Handling Prerequisite Resources for a Job* in the *User's Guide*.

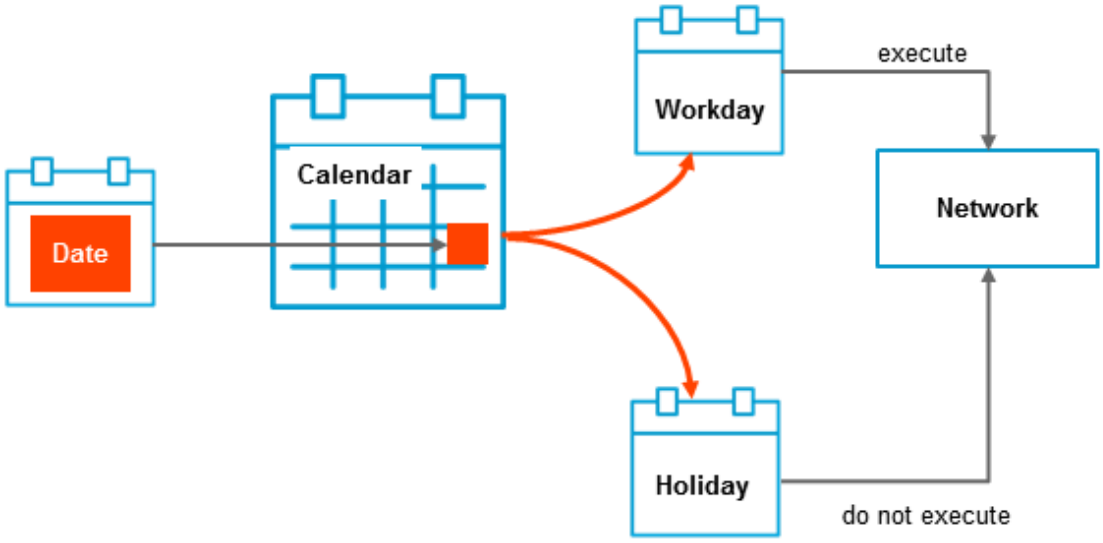
Calendars

Calendars of any number can be defined to the system and easily modified online. They can apply to a dedicated owner or to the whole system.

The only purpose of having calendars in Entire Operations is to distinguish between workdays (working days) and holidays (non-working days). Entire Operations does not perform any activities on holidays. Entire Operations accounts for holidays by not activating a network if the scheduled date is marked in the calendar as a holiday.

You can determine whether a job network scheduled for execution on a holiday should be activated before or after the scheduled day or whether it should be cancelled.

On the other hand, if you want a job network to run, for example, each Friday, you do not need a calendar.



This section covers the following topics:

- Calendar Definition

Related Topics:

- *Calendar Maintenance - User's Guide*

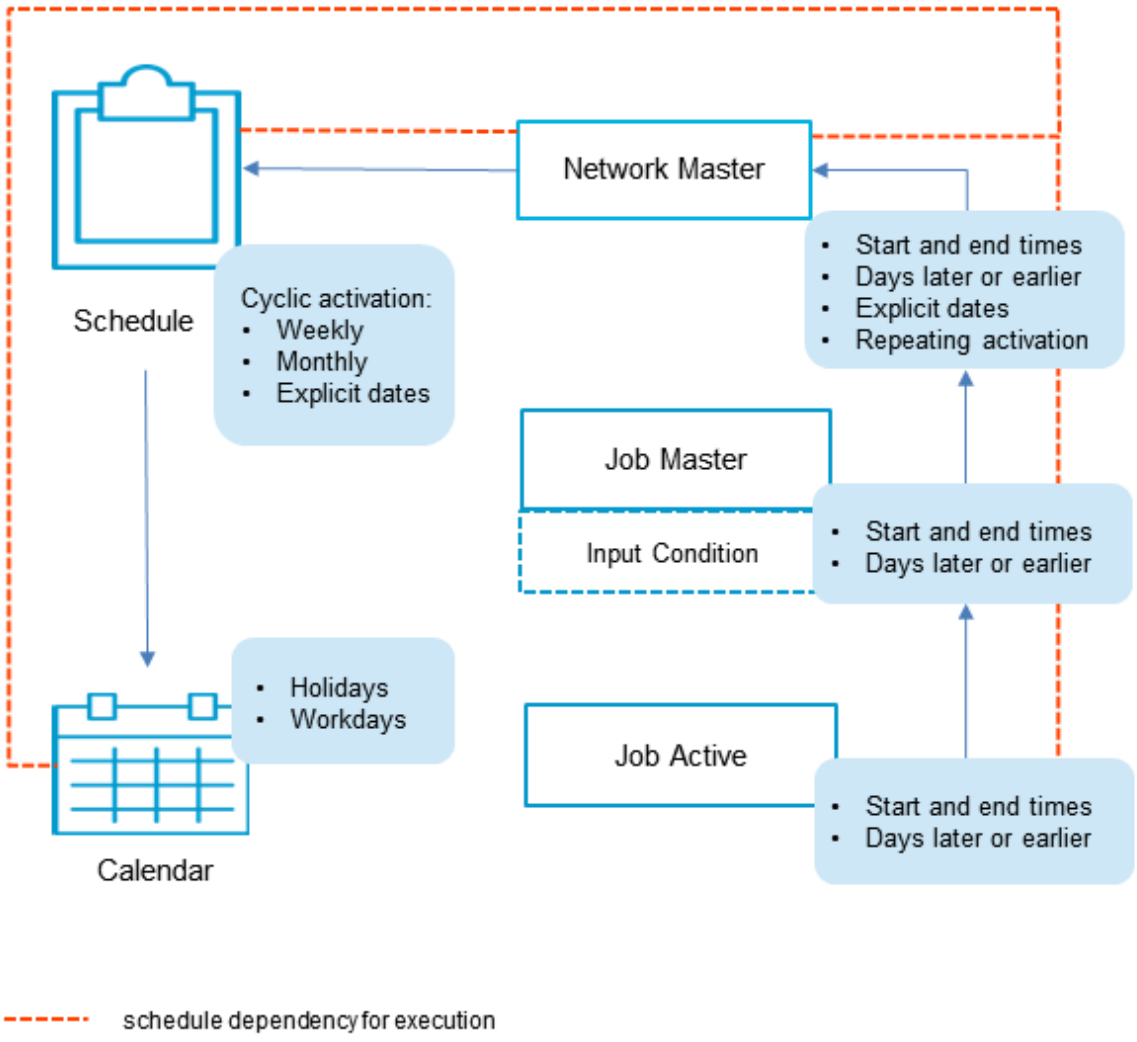
Calendar Definition

Calendars are referenced by schedule tables which are defined in the network maintenance facility. 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).

A calendar is defined by name and year. Defining new calendars or modifying existing ones consists of specifying or marking holidays (non-working days).

Schedules

Schedules contain the planned execution dates of job networks. They can contain periodic and/or explicit schedule dates as indicated in the following graphic:



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 workday of a month).

Related Topics:

- *Schedule Maintenance - User's Guide*

Symbol Tables and Symbols

Symbol tables are user-defined tables, each containing a list of symbol names together with their current value. These tables are used during **dynamic JCL generation**. The benefit of using a symbol table is that it must be created only once but can be referenced in a huge number of jobs.

You can define any number of symbol tables and use them just by specifying their name in the definition of the appropriate job networks.

Symbols can be defined for prompting during or before job network activation.

Each network activation has its own active copy 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 from the symbol table. You can use two escape characters to determine whether this replacement should take place at JCL generation time or at job submission time.

There are also many predefined symbols (see the *User's Guide*) available within Entire Operations.

Symbols are searched for in several tables (like in STEPLIBs) as described in *Symbol Table Types and Symbol Search Order* (*User's Guide*). Symbols can contain other symbols recursively; system variables can be used to construct symbol values.

A symbol table belongs to an owner. Any number of symbol tables can be associated with an owner. You can update all symbol tables of all owners for which you are authorized.

Symbols can be examined and modified by APIs (Application Programming Interfaces) from any Natural application. Scheduling such a program as part of an Entire Operations job network makes it possible to modify active symbol tables even during the execution of the job network.

Related Topics:

- *Symbol Table and Symbol Maintenance - User's Guide*

Object Versioning

With Entire Operations you can use different versions of objects of type **job network** and **symbol tables**.

The versioning of job networks and symbol tables is optional.

For each job network and for each symbol table you can decide individually whether you like to use versions, or not.

You may use versioning:

- to archive previous network versions, so that they can be activated manually any time later;
- to archive previous symbol table versions;
- to prepare new network versions or symbol table versions for future use;
- to define which version is to be used for scheduled network activations.

For detailed information on object versioning, see *Maintaining Job Network Versions* and *Versioning of Symbol Tables* in the *User's Guide*.

Operating System Server Nodes

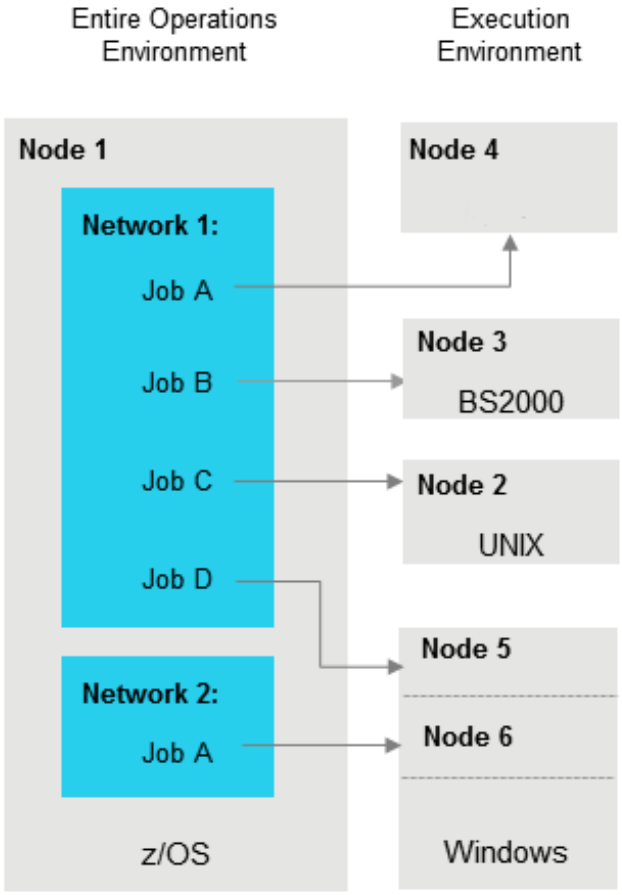
Nodes are Entire System Servers and refer to machines on which requests to the operating system are executed. They are distinguished by numerical identifiers in the same way as database IDs distinguish between different Adabas databases. Within Entire Operations, each machine is assigned a node. More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

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

Communication paths between otherwise isolated nodes are provided by the Software GmbH 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:



Related Topics:

- *Definition of Nodes - Administration* documentation

5 Entire Operations Monitor

- Monitor Subtasks 46
- Running the Monitor as a Batch Task 47

The Monitor is the heart of Entire Operations. The Entire Operations Monitor is a complex program that is activated 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 performs the following functions:

- Automatically activates scheduled networks (copies them to the active database);
- Checks time windows for job or job network execution;
- Checks input conditions and resources;
- Submits jobs according to their (internal) priority;
- Keeps track of jobs in the various queues of the operating system;
- Analyzes the **End-of-Job** status of each job that terminated, determines which events have occurred and triggers appropriate system actions (set logical conditions, send messages, start programs);
- Logs all important information;
- Cleans up the **active database**.

The functions of the Entire Operations Monitor can be distributed to various subfunctions (subtasks). When using subtasks, certain processing steps can run in parallel and multi-processor environments can thus be used to optimize performance. Distributing typical Monitor functions is the responsibility of the system administrator.

In technical terms, there are two ways of running the Monitor: as one or several subtasks or as a batch task.

Related Topics:

- *Monitor Defaults - Administration* documentation

Monitor Subtasks

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

Subtasks under z/OS

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

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 and UNIX* 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, BS2000 or UNIX.

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.

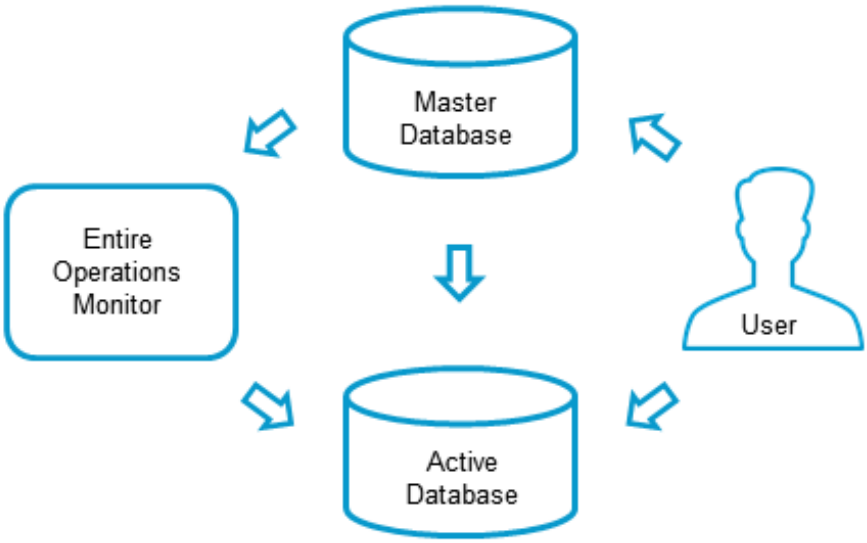
The system administrator can define a time interval between Monitor task cycles. At the beginning of a task cycle, the Monitor "wakes up" and checks the Entire Operations work queues, performing any necessary actions such as submitting jobs and analyzing the **End-of-Job** status of each job that terminated. The shorter the wait time between task cycles, the shorter the interval between job termination and its End-of-Job analysis. The price for this is increased overhead due to Monitor reactivation.

6 Master Database and Active Database

- Master Database 50
- Active Database 51
- Run Numbers 51
- Job Process Control 52

All definitions and information concerning any user, job network, job, and scheduling information are stored on the master database. It also contains all information pertaining to defined logical conditions, resources, calendars and symbol tables.

When a job network is activated, all definitions and information are copied to the active database.



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

Related Topics:

- *Activation of Networks or Jobs - User's Guide*
- *Maintaining Active Jobs and Networks - User's Guide*

Master Database

All information stored on the master database can be maintained online.

The master database is an Adabas file. This automatically provides features such as user synchronization, data integrity, data compression, auto-extension and auto-restart capability. The stored objects can be maintained online. These objects are:

- User profiles;
- Job network definitions;

- Job definitions;
- Input and output conditions;
- Resource definitions;
- Schedule tables;
- Calendars;
- Symbol tables.

Active Database

A network is activated either automatically by the Entire Operations **Monitor** according to its scheduled date, or manually by the user on demand. The active database may thus contain several copies of the same job network, each identified by a different **run number**.

The following information is stored:

- Current definition of scheduled job networks and their current symbol tables;
- Active JCL library (this means that all JCL information is copied from external storage media to the active database);
- Current status of input and output conditions;
- Current job status.

The active database can be accessed, and its information can be modified in the same way as the master database. Changes to any object in the active database are only valid for the selected run of the network and do not affect the network and job definitions stored on the master database. This enables you to make modifications which are valid only for a specific production cycle.

Run Numbers

Active objects in Entire Operations are identified by a run number, which is assigned to them automatically during the creation of an active object. Active networks or jobs are created during a network activation or job activation.

- Run numbers are unique on the job network level.
- Run numbers can be assigned to planned activations too. In the planning phase, no active jobs exist for a given run number.
- There is no guarantee that the numbering of network activation is ascending with the activation times.

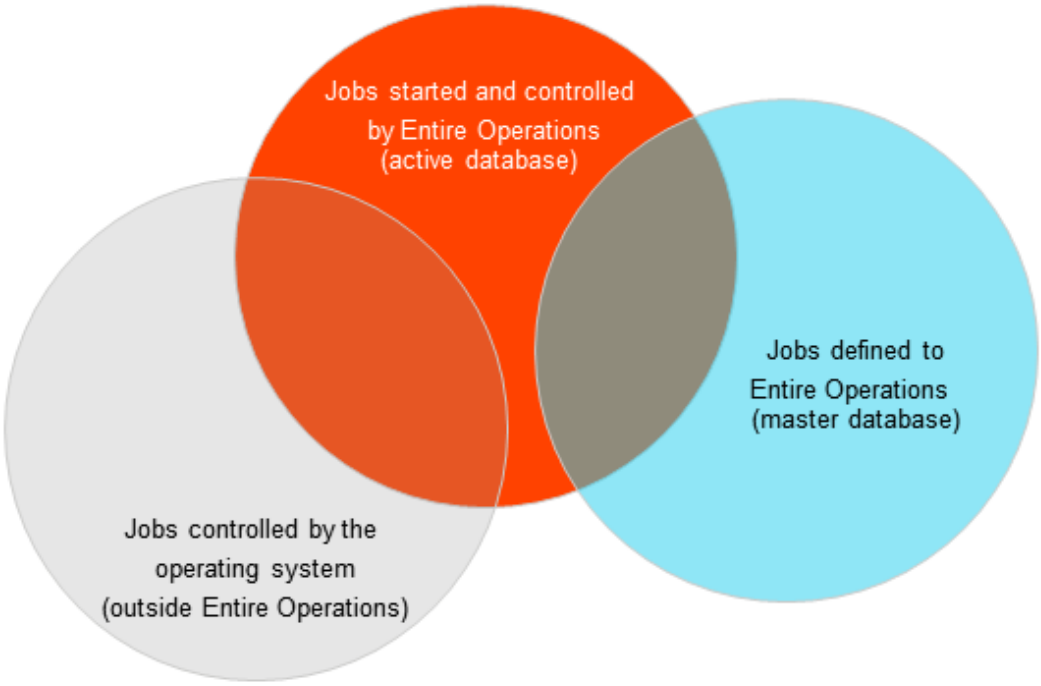
Job Process Control

You can distinguish between the following groups of jobs that can or cannot be controlled by Entire Operations:

- **Jobs in the Entire Operations master database**
  Inactive jobs defined to Entire Operations
- **Jobs in the Entire Operations active database**
  Jobs that are started and controlled by Entire Operations including jobs that are not submitted to the operating system

  For example, Entire Operations dummy-type jobs or Natural programs are not submitted to the operating system but executed by Entire Operations.
- **Operating system jobs**
  Jobs that are controlled by the operating system including jobs not defined to Entire Operations

This is illustrated in the following graphic:



- The orange circle contains all jobs started and controlled by Entire Operations. The blue circle contains all jobs defined to Entire Operations but not started. Entire Operations cannot control jobs that are not defined to it, and that are not started by it.
- The intersection of the orange and blue circle indicates Entire Operation jobs that are ready to be started by Entire Operations.
- Jobs outside Entire Operations (contained in the gray circle) run on the computer outside the control of the Entire Operations **Monitor**. On the other hand, Entire Operations holds information on jobs that run outside the control of the operating system.
- The intersection of the gray and orange circle indicates Entire Operations jobs submitted to the operating system.

7 Operating System

Entire Operations initiates job processing on the underlying operating systems: jobs, tasks, scripts or Natural programs are started. Several operating systems can be served simultaneously by one Entire Operations **Monitor**.

The Monitor sends the appropriate requests to the Entire System Server execution node. With Entire Net-Work, the Monitor can do this across the whole network.

Related Topics:

- *Operating System Server Nodes*

8 Tools and Utilities

- Administration Functions 58
- Object-Specific Maintenance Functions 58
- Process Control and Monitoring Functions 59
- Entire Operations Utilities 60
- Import and Export Functions 61

Administration Functions

Entire Operations provides administration functions to maintain user permissions and profiles, perform global maintenance functions and specify default system settings.

Related Topics:

- *Administration* documentation

Object-Specific Maintenance Functions

Entire Operations provides object-specific maintenance functions as indicated for the individual objects in the section ***Entire Operations Maintenance Objects***.

Diagram Function for Network Maintenance

Entire Operations GUI Client provides a diagram function that represents a network (master and active) and can be used to maintain all objects that belong to a network. The maintenance functions provided in the diagram correspond to the functions available in the tree view of the **main application window**.

Related Topics:

Viewing and Maintaining a Job Network Diagram - User's Guide

Edit Functions

Entire Operations provides edit functions to create, view or modify any of the following:

- Master JCL in an operating system file or a Natural object.
- Active JCL in the Entire Operations system file.
- Macros sources for **dynamic JCL generation**.
- Natural programs and subprograms that run as jobs in job networks or execute as user exit routines.
- Online documentation (text descriptions) of networks, jobs and job events.
- Job protocols, job output and listings (read-only view).
- The Entire Operations system log (read-only view).

Related Topics:

- *Editing Master JCL and Natural Sources - User's Guide*
- *Handling Macro Sources for Dynamic JCL Generation - User's Guide*
- *Documenting Your Networks - User's Guide*
- *Writing and Viewing Online Documentation for a Job - User's Guide*
- *Creating and Viewing Online Documentation for Events - User's Guide*

Process Control and Monitoring Functions

Entire Operations provides several features to control and monitor network processing.

- Logging
- Reporting
- Diagram Function for Network Control

Logging

Entire Operations provides a log file which records every event during network processing. It also records all manual changes to active jobs, such as JCL corrections or updates of symbol tables. This information is available online and can be used for reporting and statistical purposes.

The log information can be selected according to job network and job, as well as according to dates and times, or even users. The log reports can also be exported into a sequential file, thus providing the opportunity for further analysis with any other tool and according to any criteria.

Related Topics:

- *Log Information - User's Guide*

Reporting

Entire Operations provides reports with system information useful for keeping track of operations, for making scheduling decisions or even for planning future production cycles.

The reports cover the following:

- Information on all jobs, selectable according to terminated jobs, abended jobs and jobs which have not been started. Date ranges and network names can be specified to further narrow selection down. Reports on all jobs include all events, activation time, messages, termination status etc. All job reports are sorted according to log time.
- Network information, selectable according to whether a short overview is required or a longer report that includes detailed information on network components. In all network reports, in-

formation is provided on network and job definition, all input conditions and resources, as well as **End-of-Job** handling, including output conditions. The more detailed reporting option additionally displays all text descriptions available on the network, job and event level.

- Schedule overview of selected or all networks, consisting of a list of jobs to be scheduled within a specified range of dates. This can either be done for time ranges belonging to the past to get a list of all unsatisfied network activations or you can request the Network schedule overview for future production periods to get information for forecasting and planning purposes.
- Cross-reference information on the use of Entire Operations objects.

All reports can be viewed online and printed or stored for archiving purposes.

Using Entire Operations import/export functions, the contents of the master database can be unloaded to a sequential file, too. This feature is intended for migration and transport purposes, but you can also use it to build up your own reporting system.

Related Topics:

- *Log Information - User's Guide*
- *Reporting - User's Guide*
- *Cross-References - User's Guide*

Diagram Function for Network Control

Entire Operations GUI Client provides a diagram function that represents a network (master and active) and can be used to monitor all objects that belong to a network, see whether they are connected and check network execution.

Related Topics:

- *Viewing and Maintaining a Job Network Diagram - User's Guide*

Entire Operations Utilities

Entire Operations provides utilities that 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.

Related Topics:

- *Entire Operations Utilities - User's Guide*

Import and Export Functions

Entire Operations provides import and export functions that can be used for migration purposes and mass updates.

Related Topics:

- *Import/Export Functions* documentation

9 APIs and User Exits

Entire Operations provides Application Programming Interfaces (APIs) that are used for accessing Entire Operations data and performing Entire Operations functions from any Natural application.

Entire Operations provides user exit routines that can be used for updating databases, conducting plausibility checks during network activation, handling files and accessing operating system functions.

Related Topics:

- *API Routines - User's Guide*
- *User Exits - User's Guide*