



SYSTEMS MANAGEMENT

Entire Operations

Concepts and Facilities

Version 3.2.1

 **SOFTWARE AG**



Order Number: NOP300-006ALL

This document applies to Version 3.2.1 of Entire Operations and to all subsequent releases. Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

© January 2001, Software AG
All rights reserved

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

Table of Contents

Concepts and Facilities - Overview	1
Concepts and Facilities - Overview	1
What Is Entire Operations?	2
What Is Entire Operations?	2
Introduction	2
Entire Operations Schedules and Controls Job Networks	2
More than a Scheduler	3
Control Across Operating Systems	3
Entire Operations in a Multi-CPU Environment	3
Intelligent Process Control	4
Entire Operations Viewer	4
Integration	4
Summary of Benefits	5
Entire Operations Objects	7
Entire Operations Objects	7
Entire Operations Object Relationship Diagram	7
Owners and User IDs	8
Users, Owners and Job Networks	8
Job Networks	9
Jobs	9
Job Types	10
Job Attributes	10
Job in a Multi-CPU Environment	10
Logical Conditions	10
Logical Condition Overview	11
Input and Output Conditions	11
Jobs Linked by Input and Output Condition	11
Job Network with Logical Conditions	12
Mailboxes	13
Concept of Mailboxes	14
Resources	14
Calendars	15
Schedules	15
Symbol Tables	16
Entire Operations Components	17
Entire Operations Components	17
Master Data Base	17
External JCL	18
Active Database	18
Monitor	19
Operating System	19
User Interface	19
Entire Operations User Interface	20
Uniform Screen Layout	20
Line Commands	20
Direct Commands	20
PF Keys	21

Window Technique	21
Entire Operations Online Help Facility	21
English and German	22
Color Support	22
Logging Facility	22
Reporting Facility	22
Editor	23
Entire Operations Editor - Job Description	23
Entire Operations Facilities	24
Entire Operations Facilities	24
Main Menu	24
Job Network Maintenance	25
Network Maintenance	25
Job Network Scheduling	25
Network Scheduling	25
Job Maintenance	26
Job Maintenance Screen	26
Job Scheduling	27
Job Scheduling Screen	27
Calendar Definition	28
Calendar Definition - Example	28
Logical Condition Maintenance	29
Input Conditions	29
Input Condition Maintenance	30
Output Conditions	30
End-of-Job Checking	31
End-of-Job Checking and Actions	31
Default Checking	32
End-of-Job Actions	32
Using Resources	33
Resource Definition	33
Dynamic JCL Generation	33
Example 1: Dynamic JCL in an OS/390 Environment:	34
Example 2: Dynamic JCL in a BS2000/OSD Environment:	35
Example 3: Dynamic JCL in a UNIX Environment:	37
Editing System Objects	38
Generating Reports	39

Concepts and Facilities - Overview

This documentation covers the following topics:

- **What Is Entire Operations?** Describes the purpose of Entire Operations, which is Software AG's software system for the automated control and scheduling of job networks and which provides all the functions required to define any type of background processing.
- **Entire Operations Objects** Explains Entire Operations Objects, such as Owners and User IDs; Networks; Jobs; Logical Conditions; Mailboxes; Resources; Calendars; Schedules, and Symbol Tables.
- **Entire Operations Components** Provides information on NOP components like the Master Data Base; External JCL; Active Data Base; Operating System (CPU); Monitor; User Interface; Logging Facility; Reporting Facility, and Editing Facility.
- **Entire Operations Facilities** Describes a number of menu-driven Entire Operations Facilities which allow the user to define objects to the system and control and monitor network processing.

For details on how to navigate within the documentation, see [Using the Documentation](#).

What Is Entire Operations?

This section covers the following topics:

- Introduction
 - More than a Scheduler
 - Summary of Benefits
-

Introduction

Essentially, Entire Operations is Software AG's software system for the automated control and scheduling of job networks. It provides all the functions required to define any type of background processing.

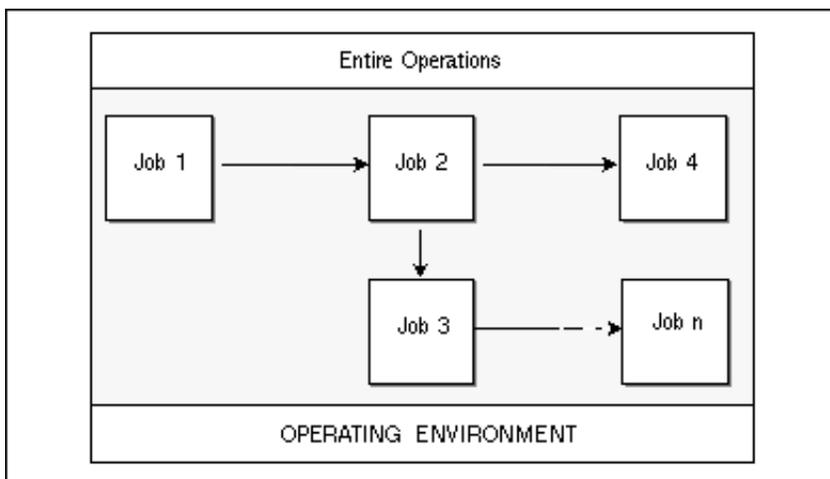
Entire Operations behaves according to the rules specified using a comfortable user interface that guides the user with all the task-oriented features of a highly modern system:

- object-oriented menus and direct commands for easy navigation
- pop-up windows for help texts and selection of items
- online tutorial
- quick execution of functions using PF keys
- Windows graphic interface (Entire Operations Viewer)

Entire Operations is available in English and German. Other languages are supported on request.

Entire Operations requires no modification to underlying operating system or to any of the subsystems 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

Have you heard it all before? Does it sound like "just another scheduling system"? Well, read on: Entire Operations offers much more:

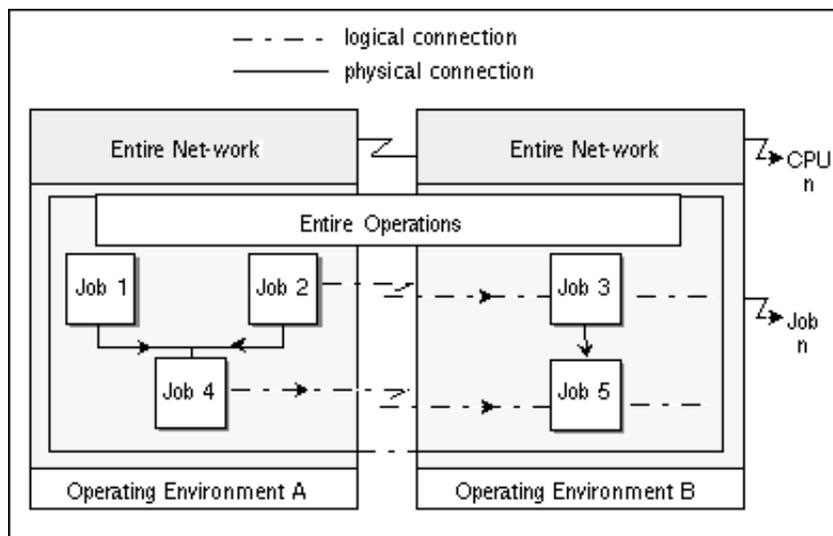
- Control Across Operating Systems
- Entire Operations in a Multi-CPU Environment
- Intelligent Process Control
- Entire Operations Viewer
- Integration

Control Across Operating Systems

- **Mainframe environments:**
Entire Operations can be installed on any of the mainframe platforms OS/390, VSE/ESA and BS2000/OSD running any TP/DC system such as Com-plete, CICS, TSO, IMS, UTM or TIAM.
- **UNIX:**
Entire Operations controls production processes on UNIX systems (HP-UX, SINIX RM, AIX), for example, in conjunction with a mainframe environment. You can also manage your mainframe production from a central UNIX node or use Entire Operations to control a group of UNIX machines.

Background processes running in heterogeneous multi-CPU configurations can also be controlled and monitored by Entire Operations if the computers are interlinked with Software AG's communication service Entire Net-work. In such a distributed environment, job networks can consist of processing steps that execute on different, even diverse computers:

Entire Operations in a Multi-CPU 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.

Entire Operations can even balance the system's workload by sharing out work dynamically. During processing, a decision can be taken based on the current load to determine on which computer the next processing step is to be executed.

Intelligent Process Control

Automating any type of operation requires thorough advance planning in order to map the processes in the language of the automation tool. Experience shows, however, that tomorrow's reality often differs from today's plan. One can, of course, anticipate a number of exceptional situations and plan accordingly; and it goes without saying that Entire Operations provides a wide range of features that enable it to react appropriately whenever such a situation arises.

However, the extent to which you can anticipate possible system situations and their consequences is limited; especially in large job network topologies, each new fork in processing adds to the system's complexity, which is neither desirable nor necessarily helpful.

Entire Operations addresses this problem by allowing 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, availability of specific files, and many more. In other words, job control can "read" a current situation and adapt accordingly.

It is here that the functional depth of Natural, Software AG's 4GL Development Environment is particularly helpful. For not only does the power of Natural allow you to map any conceivable decision criterion, but it can also put all relevant data at your disposal, thanks to its many 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 each and every possible problem situation and spend time and money in defining remedial action before processing starts. All you need to do is specify strategies (= programs) that recognize and rectify problems as they occur.

Entire Operations Viewer

In addition, you can visualize and analyze diagrams of your Entire Operations networks with its Windows component Entire Operations Viewer: simply download the job network data from your mainframe to your PC. Use Entire Connection or any file transfer program. With Entire Operations Viewer, you can now scroll your network diagrams, enlarge and print them.

Integration

Have you never been annoyed by the sheer number of applications or programs you need for your daily work? Depending on the software configuration at your site, you either have to go through tiresome logon and logoff procedures, or suspend and resume application sessions using a session manager (often to the detriment of system resources).

With Entire Operations, such session switching is a thing of the past, as Entire Operations provides the capability of integrating any other application(s) in its main menu. If any other products from the Entire Systems Management line are installed, Entire Operations recognizes them and includes them in its main menu automatically. These are:

- 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 data base 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, users can provide input data necessary for daily or future production runs.

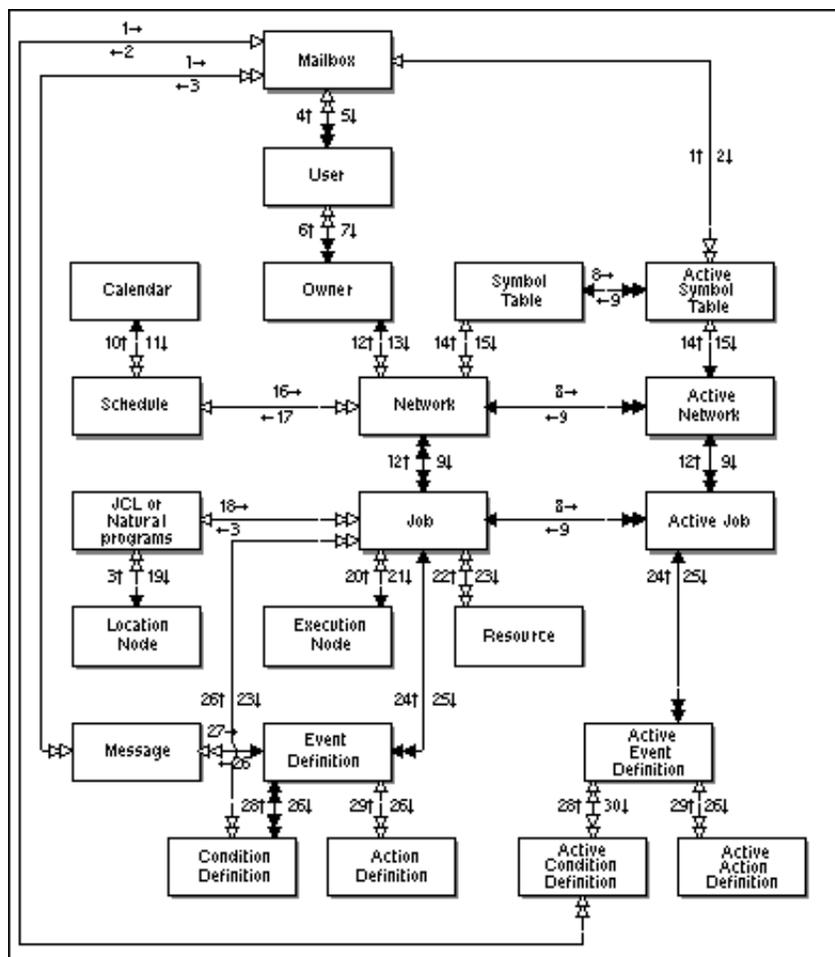
Entire Operations Objects

Working with Entire Operations involves defining and maintaining certain objects which are explained in the following subsections. These objects are:

- Owners and User IDs
- Job Networks
- Jobs
- Logical Conditions
- Mailboxes
- Resources
- Calendars
- Schedules
- Symbol Tables

The following figure is a graphic representation of these objects and their relationships:

Entire Operations Object Relationship Diagram



1 = is sent to	16 = schedules	← must be only one
2 = contains prompting for	17 = is scheduled on the basis of	⇐ none or at most one
3 = contains	18 = is executed as	⇐⇐ none or one or many
4 = uses / can use	19 = resides on	⇐⇐⇐ must be at least one, can be more than one
5 = contains messages for	20 = is platform for	
6 = authorizes	21 = runs on	
7 = can choose	22 = is preliminary condition for	
8 = lends properties to	23 = depends on / can depend on	
9 = is composed of	24 = determines result of	
10 = is based on	25 = is checked for	
11 = is basis of	26 = triggers	
12 = belongs to	27 = is set according to	
13 = owns	28 = is set or reset according to	
14 = obtains parameters from	29 = is performed according to	
15 = contains values for	30 = sets	

Owners and User IDs

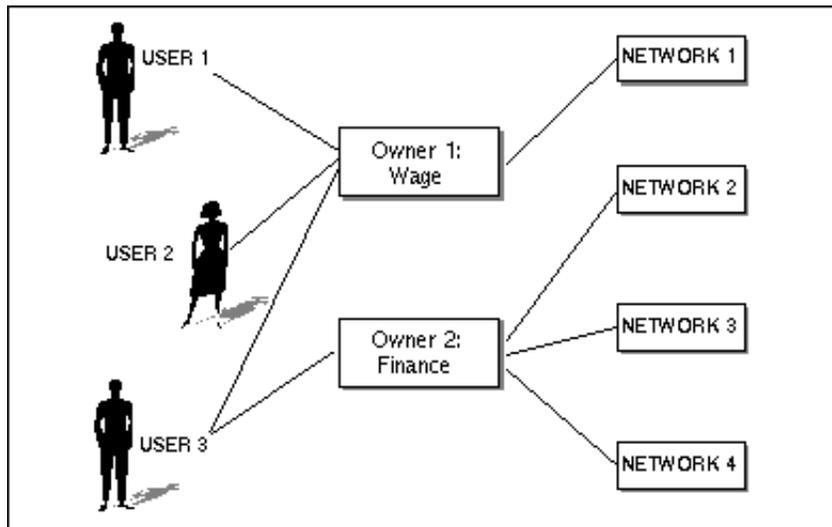
Each Entire Operations user is assigned a definite **user ID** for the purpose of security, profiling, message switching and logging.

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).

Owner names enhance security and ease of use by grouping user IDs and associating job networks to these group names. 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.

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

Users, Owners and Job Networks

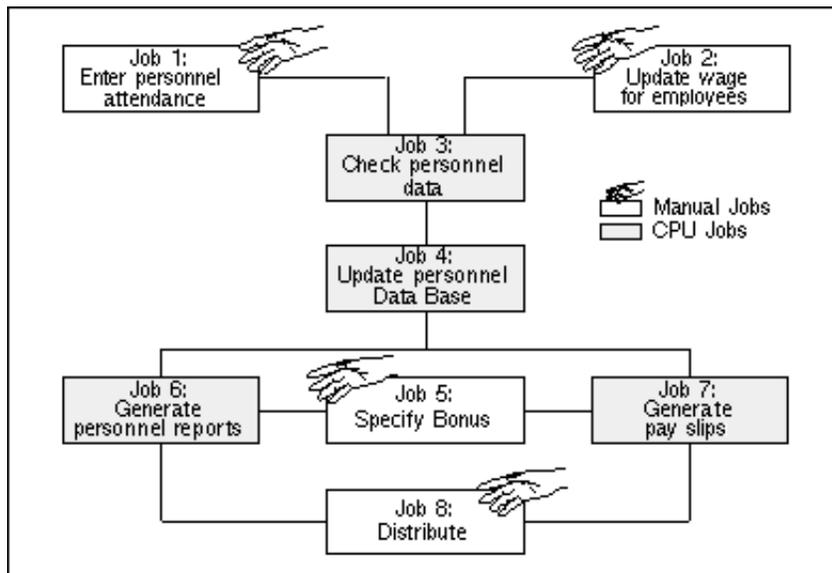


All three users may maintain networks belonging to owner **Wage**, while user 3 is also authorized to access networks belonging to owner **Finance**.

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 manual tasks in these job networks, 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 be automatically scheduled for work by the Entire Operations Monitor. An authorized user can manually activate the networks.

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

Jobs

The **job** is one of the basic objects of the Entire Operations system. A job can be a computer-driven task (CPU job) or a manual task performed by the user.

This subsection covers the following topics:

- Job Types
- Job Attributes
- Job in a Multi-CPU Environment

Job Types

Entire Operations recognizes the following **types** of CPU jobs:

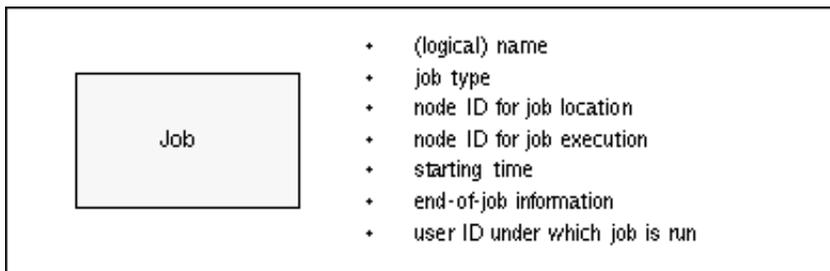
- Standard jobs of the operating system (OS/390, VSE/ESA, BS2000/OSD);
- Standard shell scripts of the UNIX operating system;
- Started Tasks (OS/390);
- Natural programs;
- Natural MACRO job (constructed with dynamic JCL);
- Cyclic jobs (for supervisory functions etc.).

In addition, for non-CPU jobs there are:

- Dummy jobs to create time windows for non-CPU jobs or Boolean connections for single conditions.

Job Attributes

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



Such a job can be contained in several **job networks**.

Job in a Multi-CPU Environment

When Entire Operations is used in a **multi-CPU 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'.

Logical Conditions

This subsection covers the following topics:

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

Logical Condition Overview

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 CPU or manual event. This event must occur before Entire Operations can proceed to the next step.

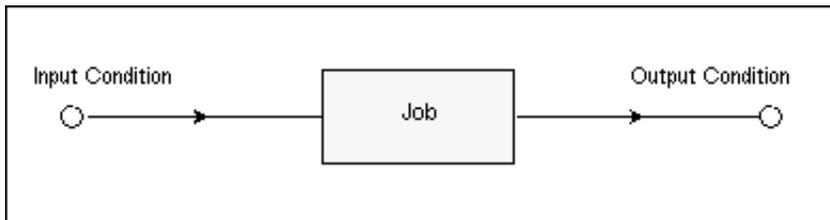
When a job network is activated, each logical condition is assigned a run number. This run number enables Entire Operations to distinguish between the same event that occurs during different network activations.

Logical conditions can be used in two different ways:

- As **input conditions**;
- As **output conditions**.

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

Input and Output Conditions

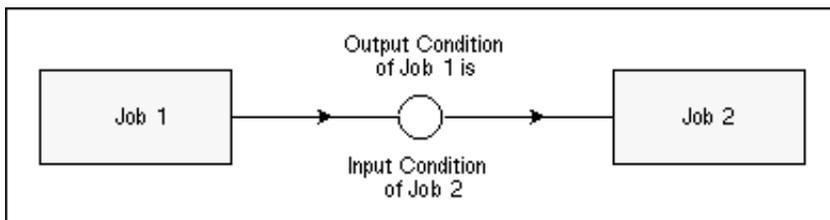


All input conditions must be fulfilled before a job can be submitted (prerequisite condition). You can define any number of input conditions for a job.

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 analysis, 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.

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:

Jobs Linked by Input and Output Condition

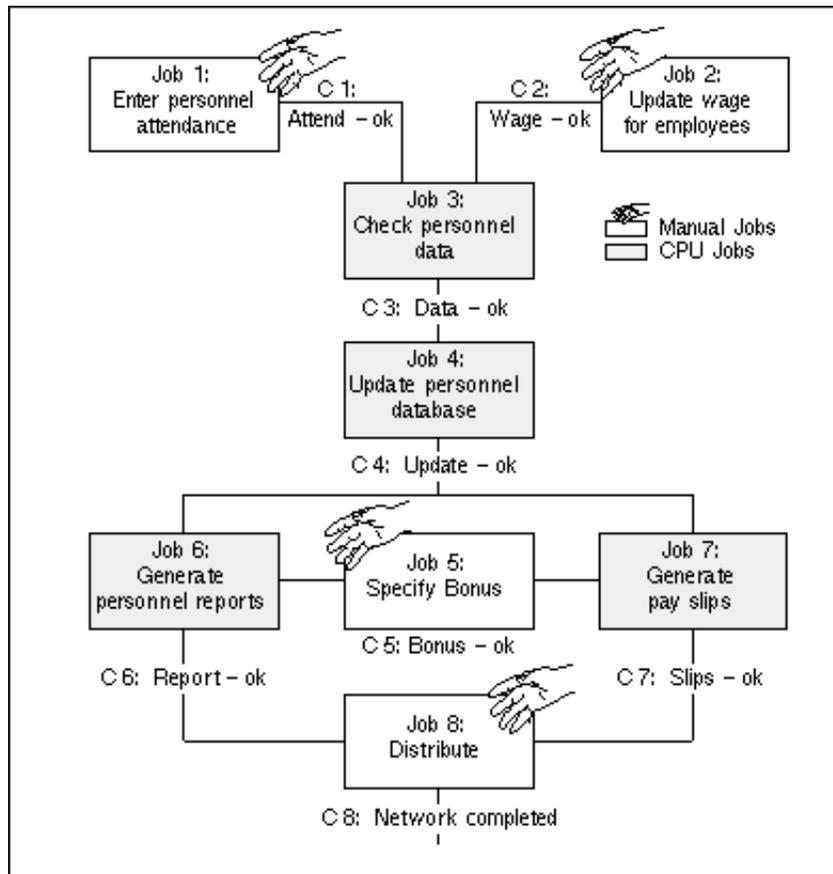


A specified event occurred as a result of **Job 1**. This sets the condition, signaling to Entire Operations that **Job 2** can be started.

Any number of logical conditions can be defined for any one job. You can even link jobs together which belong to different job networks or which are executed on different computer nodes.

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

Job Network with Logical Conditions



The following table gives an overview of the job dependencies (logical conditions) that link the jobs illustrated under Job Network with Logical Conditions.

Jobs linked by Input and Output Conditions

Job Number	Input Condition	Output Condition
Job 1		C 1: Attend - ok
Job 2		C 2: Wage - ok
Job 3	C 1: Attend - ok	
	C 2: Wage - ok	C 3: Data- ok
Job 4	C 3: Data - ok	C 4: Update - ok
Job 5		C 5: Bonus - ok
Job 6	C 4: Update - ok	
	C 5: Bonus - ok	C 6: Report - ok
Job 7	C 5: Bonus - ok	
	C 4: Update - ok	C 7: Slips - ok
Job 8	C 6: Report - ok	
	C 7: Slips - ok	C8: Network completed

For example, Entire Operations will not start **Job 6** (Generate personnel reports) until input conditions **C 4** and **C 5** are fulfilled (these input conditions are also defined as output conditions for **Jobs 4** and **5**, respectively).

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

Mailboxes

Within Entire Operations, **mailboxes** serve to send network-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.

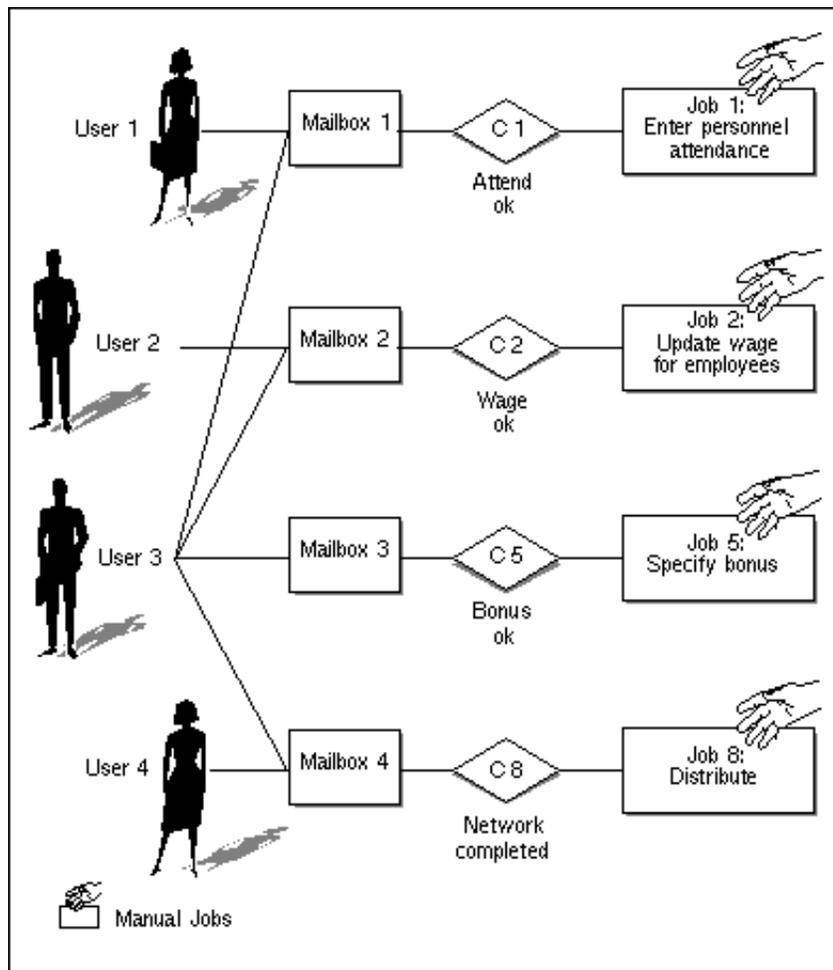
Such messages and requests can be:

- User-defined messages.
- System messages from the Entire Operations Monitor.
- Requests for logical conditions, which must be confirmed by any person. These conditions are defined as dependent on any manual action.
- Requests for pending symbol prompting for a scheduled network activation in the near future.
- Using the concept of mailboxes Entire Operations can treat non-CPU-driven tasks in the same way as CPU-driven tasks:
 - tasks can be made dependent on logical conditions and can also set logical conditions;
 - assign a mailbox to these logical conditions to specify who should be informed about them.

All users linked to a mailbox can display its list of outstanding messages and requests by issuing the direct command MAIL. With a single keystroke, the user is able to confirm the fulfillment of a condition or perform another requested action, like symbol prompting. After this the message vanishes from the list, the Entire Operations Monitor is automatically informed about the confirmation and triggers all jobs waiting for this condition.

The following figure illustrates the concept of mailboxes using the example of the payroll department job network:

Concept of Mailboxes



For example, **User 1** (who could be in data collection) is notified that **condition 1 (Attend-OK)** is not fulfilled. She can then take the necessary steps by completing personnel attendance data and confirming this in her mailbox, when finished. **User 3** (who could be assistant to 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 and the pay slips can be distributed.

Up to ten mailboxes can be associated to one user ID.

Resources

It is important to distinguish between logical resources and physical resources:

- **Logical resources** can be seen as flags which are only meaningful inside Entire Operations;
- **Physical resources** are units of the operating system itself, for example, real free disk space or the exact number of existing tape units.

To make the availability of **physical resources** a prerequisite for job submission, Entire System Server functionality can be used.

Logical resources have to be defined before they can be specified as prerequisites for job submission. The user can define and modify any number of resources to regulate the job flow. Logical resources prevent jobs from running in parallel, if all other prerequisites (e.g. time windows or logical conditions) are fulfilled.

There are two types of logical resources:

- **Reusable resources:** these are released at job completion, for example, data set names or disk work space;
- **Non-reusable resources:** these can be used once only and are therefore not released at job completion, for example, printing paper.

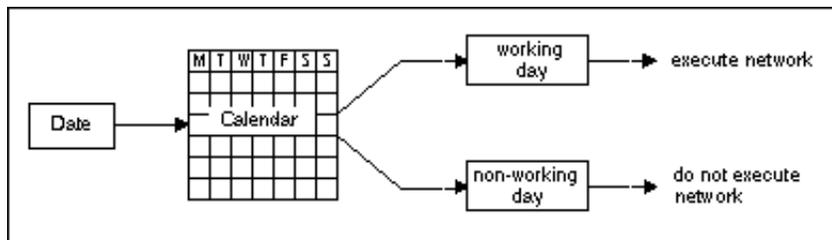
In our example of the payroll department job network (Job Network with Logical Conditions), we could prevent **Job 6** (Generate Personnel Report) and **Job 7** (Generate Pay Slips) from running in parallel by defining a resource (e.g. **CPU time**) with an initial quantity of **100**, and defining this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

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 **working days** and **non-working days**. Entire Operations does not perform any activities on non-working days. You can determine whether a job network scheduled for execution on a non-working day should be activated before or after the scheduled day or whether it should be cancelled.

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



Schedules

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 (e.g.: the last working day of a month).

Symbol Tables

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 a job network activation. The data entered at the user's terminal is then assumed by the JCL to be executed.

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 substituted by its current value from the symbol table. You can use two escape characters to determine whether this substitution should take place at JCL generation time or at job submission time.

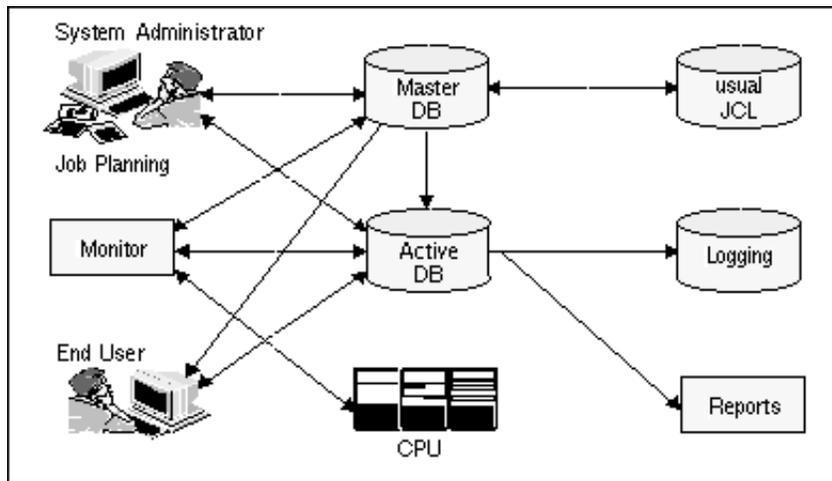
Symbols are searched for in several tables (like a STEPLIB). Symbols can contain other symbols recursively; system variables can be used to construct symbol values.

A symbol table belongs to an owner. Any number of symbol tables can be associated to an owner. The user can update all symbol tables of all owners for whom he is 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.

Entire Operations Components

Entire Operations consists of the following components:



- Master Data Base
- External JCL
- Active Database
- Monitor
- Operating System
- User Interface
- Logging Facility
- Reporting Facility
- Editor

Master Data Base

All definitions and information concerning any user, job network, job, and scheduling information are stored on the master data base. The master data base 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 in any environment such as Com-plete, CICS, IMS, TSO, TIAM and UTM as well as on UNIX. These objects are:

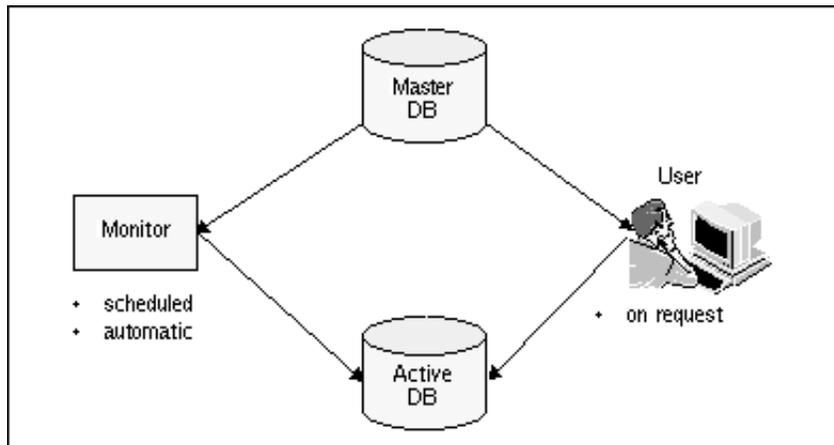
- User profiles;
- Job network definitions;
- Job definitions;
- Input and output conditions;
- Resource definitions;
- Schedule tables;
- Calendars;
- Symbol tables.

External JCL

Entire Operations can integrate JCL unchanged into job networks and the JCL can even remain in its original location. Sequential data sets and PDS, LMS and VSE/ESA libraries are supported as well as the Natural and CA-LIBRARIAN storage types. On the UNIX operating system, any number of shell scripts can be included into job control by Entire Operations.

The IMPORT function allows you to copy the JCL to the Master Data Base. This should always be used when required by particular backup criteria (access is then only possible through Natural Security) or if the JCL resides in the Master Data Base and is backed up with it.

Active Database



When a job network is activated, it is copied to the active data base. 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 data base 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 like PDS, LMS, VSE-LIBRARIAN or UNIX files to the active database)
- Current status of input and output conditions;
- Current job status.

The active data base can be accessed and its information modified in the same way as the master database. Changes to any object in the active database are only valid for the current 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.

Monitor

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 data base 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 data base);
- 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;
- Analyses end-of-job status, determines which events have occurred and triggers appropriate system actions (set logical conditions, send messages, start programs);
- Logs all important information.

The functions of the Entire Operations Monitor can be distributed to various subfunctions (subtasks). Using subtasking, 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.

Operating System

Entire Operations initiates CPU-based processing on the underlying operating system: jobs, tasks, scripts or Natural programs are started. The Monitor sends the appropriate requests to the Entire System Server execution node and with Entire Net-work can do this across the whole network.

User Interface

Entire Operations contains a menu-driven, easy-to-use user interface, which allows each user to perform all necessary functions according to his authorization profile. The system was designed so that even end users would find it comfortable to work with.

This subsection covers the following topics:

- Entire Operations User Interface
- Uniform Screen Layout
- Line Commands
- Direct Commands
- PF Keys
- Window Technique
- Entire Operations Online Help Facility
- English and German
- Color Support

The following figure illustrates a typical Entire Operations screen - the Job Maintenance screen in the Network and Job Maintenance facility:

Entire Operations User Interface

Cmd	C	R	PU	Job	Type	Description	File or Library	Member

20.04.96	*** ENTIRE OPERATIONS ***						11:39:41	
Owner	EXAMPLE			Job Maintenance			Network B60-FLOW	

—				JOB-01	MAC	Where it all starts .	EOR-T210	B60-M02
—	C1	P		JOB-012	MAC	Depending on Job-01 .	EOR-T210	E57-M01
—	C1			JOB-013	MAC	Depending on JOB-012	EOR-T210	E57-M01
—	C1			JOB-014	MAC	Depending on JOB-013	EOR-T210	E57-M01
—	C1			JOB-015	DUM	Depending on JOB-014		
—	C1			JOB-019	MAC	Depending on JOB-01	EOR-T210	E57-M01
—	C2			JOB-02	MAC	Dep. JOB-15, JOB-19	EOR-T210	E57-M02
—	C1			JOB-03	NAT	Depending on JOB-02	EOR-T210	E57-P01
—	C1			JOB-04	MAC	Depending on JOB-03	EOR-T210	E57-M01
—	C1			JOB-05	DUM	Depending on JOB-04		
—	C1			JOB-06	MAC	Where it all ends	EOR-T210	E57-M02

***** Bottom of Data *****								
A Depend. C Copy D Delete E Edit G Pregen. I Input Cond. L Resources								
M Modify O EOJ Chk + Act P Prose R Activate S Scheduling ParmS U Add.Log								
Command => _____								
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---								
Help Add End Save Up Down Menu								

Uniform Screen Layout

All Entire Operations screens have the same structure that includes:

- a **headings section** identifying the session and current function;
- a **main information section** beginning with a line command input field;
- lines containing descriptions of available **line commands**;
- a **command line** for direct access to a specific system function;
- **PF key line** describing the functions assigned to PF keys.

Line Commands

A line command is entered in the first field of a line containing data on an Entire Operations system screen and allows the user to perform a certain function on the selected item. Each screen contains a description of all available line commands;

Direct Commands

A direct command is entered in the **Command =>** line at the bottom of each system screen and allows the user to bypass the menu-driven mode of work. Any Entire Operations screen can be accessed by entering a single direct command. Direct commands have priority over line commands and PF key functions;

PF Keys

Certain maintenance, navigational and scrolling functions are assigned to PF keys. Each Entire Operations screen displays PF keys with a description of the assigned function.

Window Technique

```

27.04.96                *** ENTIRE OPERATIONS ***                15:27:38
Owner EXAMPLE          Input Conditions Maintenance              Job JOB-02
Network +-----+-----+-----+-----+-----+-----+
----- !
C Condi !                Master Input Condition Addition        ! -----
_ E60-J !
_ E60-J !      Owner    ==> EXAMPLE__                          !
_      !      Network  ==> E60-FLOW__                          !
_      !      Condition ==> BRY-1_____ Run ==> _____ !
_      !      Reference ==> _____                          !
_      !      +-----+-----+-----+-----+-----+-----+
_      !      Usage ( !                Input Condition User Routine ! : _ !
_      !      Must   !
_      !      !
_      !      Dependi !      Condition ==> BRY-1                    !
***** !      User   !      Run ==>                               ! > _ ! *****
D Delet !      File   !
_      !      User   !      Value will be determined by
_      !      Job V  !      NATURAL Library ==> _____ ! > _ !
Command !      !      User Routine ==> _____ !
_      !      -PF1---P !
Enter-PF !      Help A !      PF1 Help PF3 End          PF9 Delete ! u ! F12---
He +-----+-----+-----+-----+-----+-----+

```

Entire Operations uses the concept of windows to make it easy for the user to define and maintain system objects or to select objects from availability lists;

Entire Operations Online Help Facility

Entire Operations provides comprehensive online help which enables you to display explanatory texts for an **individual field** or for the **whole screen**.

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

Alternatively, you can enter the online help facility at the top end of the hierarchy by selecting the **Help** option on the Main Menu. This displays the main Help Menu with a list of selectable items and a short explanation of their meaning:

```

27.04.96                ENTIRE OPERATIONS HELP                15:22:15
----- Table of contents -----
Option ==>                H00000

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

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

  1  Help about Help    - The ENTIRE OPERATIONS Help System
  2  The Maps           - Functionality of ENTIRE OPERATIONS Maps
  3  Networks and Jobs - Maintenance of Networks and Jobs
  4  Calendars          - Definition and usage of Calendars
  5  Active Queue       - Display and modification of Active Jobs
  6  Symbol Tables      - Symbol table editing
  7  Logging            - Selection and display of Log Information
  8  Dynamic JCL        - How to work with dynamic JCL
  9  System             - System Administrator Utilities
 10  Active Conditions - Maintenance of Active Conditions
 11  Editor             - The ENTIRE OPERATIONS Editor
 12  Reports            - ENTIRE OPERATIONS Reporting Facility
 13  Direct Commands   - How to use the Command Line
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Quit  End

```

English and German

Entire Operations is available in English and German, including the whole user interface with maps, help information and error messages.

Color Support

Colors are extensively used to distinguish important information, warnings etc. from other information.

Logging Facility

Entire Operations provides a logging facility 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.

It is possible to define logging of JCL, job protocol and/or job-related messages for all or for some jobs.

Reporting Facility

A reporting facility provides a wide range of system information useful for keeping track of operations, for making scheduling decisions or even for planning future production cycles.

Reports are available on the basis of log information. They can be requested for all jobs or selected for a given date/time range, either for all terminated jobs or for abended jobs.

Descriptions of job networks are available in a brief outline and more detailed form.

A list of all jobs scheduled for a specific date can also be displayed. This makes it possible to forecast any future production day in advance.

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

Editor

Entire Operations includes an adapted version of the Software AG Editor which can be used to create, display or modify any of the following:

- JCL in the master data base, either in an external file or Natural member;
- JCL in the active data base;
- Natural programs and user routines;
- Online documentation of networks, jobs and any events (e.g. scratch pad information);
- Job protocols and output (in browse mode).

Entire Operations Editor - Job Description

```

Edit Prose Jb: ABC3A Nw: EORACCT----- Columns 001 072
====>                                     SCROLL====> CSR
***** ***** top of data *****
00001 Job 'ABC3A'
00002 -----
00003 This job forces a defined event at end-of-job time and executes the
00004 program 'EORCONTI' which resides on the installation library
00005 'EORnnn.LOAD'. The necessary JCL which is named 'EORE*' is stored in
00006 the installation library 'EORnnn.SRCE'.
***** ***** bottom of data *****

```

```

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      End      Rfind Rchan Up      Down      Impo      Left      Right Curso

```

The Entire Operations Editor allows you to handle all these data types in the same way in any operating system and run time environment supported.

Entire Operations Facilities

Entire Operations provides a number of menu-driven facilities which allow the user to define objects to the system and control and monitor network processing. This section gives a brief outline of the following facilities and functions:

- Main Menu
- Job Network Maintenance
- Job Network Scheduling
- Job Maintenance
- Job Scheduling
- Calendar Definition
- Logical Condition Maintenance
- End-of-Job Checking
- End-of-Job Actions
- Using Resources
- Dynamic JCL Generation
- Editing System Objects
- Generating Reports

Main Menu

Once logged on to Entire Operations, the user is presented with the Main Menu. When the user selects an option, he will see either a submenu of functions, or a list of items that can be maintained using line commands and PF keys.

```

23.04.96                *** ENTIRE OPERATIONS ***                15:30:06
Owner EXAMPLE                Main Menu                User ID GHH
-----
Main Menu                DC Solutions
1 Network and Job Maintenance    20 Entire Event Management (V132)
2 Active Job Networks            21 Entire Output Management (V134)
3 Calendar Maintenance
4 Log Information
5 Symbol Tables
6 System Administrator Services
7 Reports                Applications
8 Import/Export
9 Help                30 sysmain

Command => _____
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help                End                Owner Mail

```

The configuration of the Main Menu may vary from user to user depending on the authorizations in his user profile.

Your own applications can be included in the Main Menu as well, thus saving you tedious logging on and off. In the example above, Entire Output Management is option 21.

Job Network Maintenance

Before Entire Operations can identify and activate a job network, this job network must be defined to the system. A short description can be added for easy identification when reviewing networks. A node number is specified to define the default machine on which all jobs in the network will run. A symbol table can be specified as well in order to use dynamic JCL creation feature. As the execution node this name is used just as a default for all jobs contained in the job network and can be overwritten for each job.

Network Maintenance

```

23.04.96                *** ENTIRE OPERATIONS ***                15:33:47
Owner EXAMPLE           Network Maintenance
Selection OR_____
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
!                                                                !
!                               Network Modification                !
! Network      ==> E05-IOC-04                                Owner ==> EXAMPLE      !
! Description  ==> I/O Conditions with user routine event_____ !
!                                                    Loop existing ==> N !
! Default Values for the Jobs                                     !
! Execution Node ==> 148 MVS/ESA      Symbol Table ==> EX-ST-COMN !
! JCL Node      ==> 148 MVS/ESA                                     !
! JCL Location  ==> NAT           Sym.Table Activation Mode ==> _ !
!                                                                !
! File         ==> EOR-T1_____ !
! VolSer      ==> _____ Password ==> _____ !
!                                                                !
! --PF1-----PF3-----PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12-- ! **
! Help      End      Save Spec  Symb SP-UR DfJb Copy  MsgRe  Menu   !
+-----+-----+-----+-----+-----+-----+-----+-----+
Command => _____
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Add   End           Save           Up     Down           NxtAc Menu

```

Job 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 data base) according to its scheduled starting time and additional schedule parameters defined for the network.

Network Scheduling

```

23.04.96                *** ENTIRE OPERATIONS ***                15:41:07
Owner EXAMPLE                Schedule Definition
Network DEMO-NET
-----

Schedule Type ==> W                M Monthly   W Weekly   E Explicit Dates
Calendar      ==> EXAMPLECAL

Earliest Start ==> 19:00:00   Schedule effective from => _____ 00:00:00
Latest Start  ==> 21:00:00   ___ days later
Deadline      ==> 22:00:00   ___ days later

Number of Activations ==> _____

Activate every ==> _____ minutes
or activate at ==> _____
                _____

Command => _____
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      End      Save      Cal      Menu

```

Schedule parameters are defined by specifying a predefined calendar and/or a list of explicit dates (e.g. 01.02.96, 07.03.96 etc.), or combining a list of months with a list of days (e.g. the specification for months can be '02, 03' and for days '01, 15'. This will schedule the network for the first and fifteenth day of February and March). Holidays can be accounted for in the calendar definition.

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

Job Maintenance

While in the job network maintenance facility, the user can request a list of jobs for a particular network. He can add, delete and modify jobs in the network. A job is defined by (a logical) name, type and location (PDS member, Natural library etc.). A short description can be added for easy identification when reviewing the jobs in the network. Additionally, a node ID can be specified for execution to override the designated machine defined for the job network.

Job Maintenance Screen

```

23.04.96                *** ENTIRE OPERATIONS ***                15:42:38
Owner EXAMPLE                Job Maintenance                Network E60-FLOW
+-----+-----+-----+-----+-----+-----+-----+-----+
!                               ! er
!               Master Job Definition                !
!                               ! 2
! Job Name      ==> JOB-012___      Mod ==> SN      20.10.95 16:05 ! 1
! Description   ==> Depending on Job-01_____ ! 1
! Job Type      ==> MAC                ! 1
! Execution Node ==> 146 MVS/ESA                !
! Special Type  ==> _      Symbol Table ==> EXAM-ST1__ ! 1
! Restartable   ==> _      Suffix Symbol ==> _____ ! 2
!                               !
!                               ! Escape Characters:  Activation ==> $ ! 1
! JCL Location  ==> NAT                Submit      ==> $ ! 1
! JCL Node      ==> 148 MVS/ESA                !
! File/NatLib   ==> EOR-T210_____ ! 2
! Member        ==> E60-M01_ _____ !
! VolSer        ==> _____ (if not catlgd) Password ==> ! **
! Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8-----PF10---PF12- !
!      Help Add End Edit Save Spec Symb Impo      Copy Menu ! g
+-----+-----+-----+-----+-----+-----+
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Add End      Save      Up      Down      Menu

```

If the job is of the dynamically generated type, the symbol table used in dynamic generation must also be specified (see the subsection Dynamic JCL Generation). The user can also edit any defined JCL simply by pressing a PF key.

Job Scheduling

The activation of each job depends on the scheduling parameters defined for it. As in the case of job networks, earliest starting time can be defined.

Job Scheduling Screen

```

23.04.96                *** ENTIRE OPERATIONS ***                15:45:59
Owner EXAMPLE                Job Maintenance                Network E60-FLOW
+-----+-----+-----+-----+-----+-----+-----+-----+
!                               Scheduling Parameters                ! r
!                               !                               !
!   Job JOB-012                !                               !
!                               !                               !
!                               !                               !
!   Estimated Elapsed Time ==> _____ Min.   Average ==>   1.91 !
!   History Elapsed Time  ==>   2.01   0.03   2.01   2.00   !
!                               2.00   2.00   2.00   2.00   !
!                               !                               !
!   Earliest Start Time   ==> 00:00:00                !
!   Latest Start Time     ==> 00:00:00  ___ days later !
!   Deadline Time        ==> 00:00:00  ___ days later !
!                               !                               !
!   Schedule Dependency   ==> N                        !
!                               !                               !
!   ---PF1---PF3-----PF5-----PF9---PF10---PF11---PF12-- !
!   Help   End           Save           LMsg  Acct   ScDep  Menu  !
+-----+-----+-----+-----+-----+-----+
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Add   End       Save       Up    Down                Menu

```

For jobs, however, the user 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). A number of days can also be specified to determine how long a job can reside at maximum in the active queue before it is executed.

The user 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.

Calendar Definition

Calendars are referenced by schedule tables which are defined in the network maintenance facility. 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, the user can add, delete or update a calendar (system-wide calendars can only be modified by the system administrator).

Calendar Definition - Example

```

23.04.96          *** ENTIRE OPERATIONS ***          15:50:18
Owner EXAMPLE          Calendar EXAMPLECAL Year 1996

      January          February          March
Monday      1  8 15 22 29          5 12 19 26          4 11 18 25
Tuesday     2  9 16 23 30          6 13 20 27          5 12 19 26
Wednesday   3 10 17 24 31          7 14 21 28          6 13 20 27
Thursday    4 11 18 25          1  8 15 22 29          7 14 21 28
Friday      5 12 19 26          2  9 16 23          1  8 15 22 29
Saturday    6 13 20 27          3 10 17 24          2  9 16 23 30
Sunday      7 14 21 28          4 11 18 25          3 10 17 24 31

      April          May          June
Monday      1  8 15 22 29          6 13 20 27          3 10 17 24
Tuesday     2  9 16 23 30          7 14 21 28          4 11 18 25
Wednesday   3 10 17 24          1  8 15 22 29          5 12 19 26
Thursday    4 11 18 25          2  9 16 23 30          6 13 20 27
Friday      5 12 19 26          3 10 17 24 31          7 14 21 28
Saturday    6 13 20 27          4 11 18 25          1  8 15 22 29
Sunday      7 14 21 28          5 12 19 26          2  9 16 23 30

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

```

A calendar is defined by name and year. Defining new calendars or modifying existing ones consists of specifying or marking holidays (non-working days). The non-working days in the example above appear in **bold** type.

Entire Operations accounts for holidays by not activating a network if the scheduled date is marked in the calendar as a holiday.

Logical Condition Maintenance

Logical conditions are dependencies between jobs. They are defined using the logical conditions maintenance facility. A logical condition can be added, deleted or modified. Any number of logical conditions can be assigned to any one job. A logical condition can have either of two statuses that determine how Entire Operations is to continue processing: TRUE (condition exists) or FALSE (condition does not exist).

All conditions are identified by name and a reference date to allow the Entire Operations monitor to distinguish between the same event occurring on different dates. Dates can be specified as relative dates or explicit dates. All relative dates are converted to real dates when the job is put in the active queue. There are two ways of using logical conditions:

- As input conditions;
- As output conditions.

Input Conditions

Input conditions must be fulfilled before Entire Operations can submit an active job. In order to link two jobs, an input condition must also be defined as an output condition for the preceding job. An input condition can be fulfilled by a CPU event or manually by the user.

Input Condition Maintenance

```

27.04.96          *** ENTIRE OPERATIONS ***          14:20:58
Owner EXAMPLE          Input Conditions Maintenance          Job E05-J02-UR
Network +-----+-----+-----+-----+-----+-----+
----- !
C Condi !          Master Input Condition Modification          ! -----
m E05-U !
_ EX-1 !          Owner          ==> EXAMPLE_____          !
_ EX-2 !          Network          ==> E05-IOC-04          !
_ EX-3 !          Condition          ==> E05-UR01-NOK_____          Run ==> _____ !
_          !          Reference          ==> RUN_____          !
_          !          !          !          !          !
_          !          Usage (mark with Y or N)          !          !
_          !          Must Exist: Y   Exclusive: _   Destroy after usage: _          !
_          !          !          !          !          !
_          !          Depending on          !          !
***** !          User Routine          ==> _          Multiple Suffixes ==> _          ! *****
D Delet !          File Existence          ==> _          Mailbox          ==> _          !
          !          User Sw.(BS2000) ==> _          Symbol Value          ==> _          !
          !          Job Var.(BS2000) ==> _          !
Command !          !          !          !          !
          ! -PF1---PF2---PF3-----PF5-----PF9---PF10---PF12--          !
Enter-PF ! Help Add End Save Xref ScDep Menu ! F12---
He +-----+-----+-----+-----+-----+-----+

```

Apart from a name and reference date, the user can also assign a mailbox to a condition. Each user ID can also be associated with up to 10 mailboxes. Entire Operations will automatically notify each user of all pending conditions assigned to any mailboxes associated with his user ID.

The user can also further specify what status the condition should be in before the job can be submitted (TRUE or FALSE), whether this job must wait until the condition applies exclusively to it (e.g. to prevent parallel running of two or more jobs with the same input condition), and whether Entire Operations is to automatically reset the condition after job submission.

Before job submission, all input conditions defined for the job are checked automatically by the Entire Operations monitor. If you want the checking to be done by a Natural user routine this routine must also be specified in the input condition definition screen.

Output Conditions

Output conditions will be maintained automatically by the Entire Operations monitor if their associated events have occurred. In this case all jobs will be started which have these conditions as input conditions. Events and output conditions are defined within Entire Operations end-of-job checking (see the subsection End-of-Job Checking below).

```

27.04.96                *** ENTIRE OPERATIONS ***                14:26:01
Owner   EXAMPLE          End-Of-Job Checking + Actions          MVS/ESA   Job JOB-01
Network E60-FLOW                               Run           Date
-----
C Action      +-----+-----+-----+-----+-----+-----+ OA
-            !                               !                               !
-            !   Job ended ok                !                               ! k
-            !                               !                               ! k
-            !                               !                               ! k
-            !                               !                               ! k
-            !   Cmd State  Condition Name    Reference    Run           !
-            !   - Set      E60-JOB1-0       RUN           !
-            !   - Set      E60-JOB2-02      RUN           !
-            !   - Reset    E60-JO15-0       RUN           !
-            !   - Reset    E60-JO19-0       RUN           !
-            !   - Reset    E60-JOB1-0       RUN           ! ****
*****        !                               !
c C P        !                               !
-            !   PRS                !                               !
-----      !                               !
A Activation !                               !
P Descr. R Re !                               ! Rtn
-            !   D Delete  M Modify  W Where used !
Command => __ ! Enter-PF1---PF2---PF3---PF5-----PF7---PF8---PF12- !
Enter-PF1---PF !           Help Add  End  Save      Up   Down  Menu  ! 2---
Help Ad +-----+-----+-----+-----+-----+-----+ u

```

As in the case of input conditions, output conditions are defined by name and reference. Additionally, the user can specify whether the output condition is to be set (to TRUE) or reset (set to FALSE) when the associated event occurs.

Up to **20** output conditions can be associated with a single event.

End-of-Job Checking

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

This subsection covers the following topics:

- End-of-Job Checking and Actions
- Default Checking

End-of-Job Checking and Actions

```

27.04.96                *** ENTIRE OPERATIONS ***                14:33:32
Owner   EXAMPLE          End-Of-Job Checking + Actions          MVS/ESA   Job JOB-02
Network E60-FLOW                                                Run        Date
-----
C Action   Step          will be checked for                      means  OA
_   P      STEP1        Condition Code = C0000                    ok
_                                               Occurrence of String Library full        not ok
_
_
_
_
_
_
***** Bottom of Data *****
_   C   P      All Checks ok
_   PRS   Any Check not ok
-----
A Activation  C Cond.  D Delete  E Edit User Rtn  J Job Var  M Modify  O Other
P Descr.  R Recovery S Sysout T Output Mgmt  U User Messages  X Action User Rtn

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

```

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 routine 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.

Default Checking

Depending on the operating system where the job was executed, Entire Operations performs some default checks to determine the job result. For OS/390 systems for example system abends or JCL errors will automatically be detected. These default checks will be executed for each job, regardless whether specific user-defined checks were requested for a job or not.

End-of-Job Actions

For each specified event, the user can define how Entire Operations has to react. Such system action can consist of any of the following:

- Automatically set or reset the output conditions associated with this event (see the subsection Output Conditions);
- Send a message to a specific user, the system console, an Entire Operations mailbox, or Software AG's office automation system Con-nect;
- Cancel or print job protocol and output;

- Perform a recovery (in case a job or job step failed).

Using 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.

Instead of this logical resources are only meaningful within the Entire Operations system. An initial quantity of a logical resource can be defined in the system administration facility.

The user can use resources to further regulate the job flow.

For example, if he 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. The user could thus specify a combination of resources to define a certain job sequence within the job flow or prevent certain jobs from running in parallel.

Resource Definition

27.04.96		*** ENTIRE OPERATIONS ***		14:39:31	
Owner EXAMPLE		Job Maintenance		Network E60-FLOW	
Cmd	C R PU Job	!		!	Member
	*-----	!	Prerequisite Resources: Job JOB-012	!	
	JOB-01	!		!	60-M02
l	C1 P JOB-012	!	Cmd Resource Name	Quantity	!
	C1 JOB-013	!			01
	G C1 JOB-014	!			01
	C1 JOB-015	!	Prerequisite Resource Definition		!
	C1 JOB-019	!			01
	C2 JOB-02	!	Resource Name ==>	_____	02
	C1 JOB-03	!			01
	C1 JOB-04	!	Required Quantity ==>	_____	01
	C1 JOB-05	!			!
	C1 JOB-06	!	Enter-PF1---PF2---PF3-----PF5-----PF12--		02
		!	Help Add End Save Menu		!

A	Depend. C Copy	!	D Delete M Modify		ces
M	Modify O EOJ Ch	!			dd.Log
Command =>	_____	!	Enter-PF1--PF2--PF3--PF5-----PF7--PF8--PF12		!
Enter-PF1---PF2---P		!	Help Add End Save Up Down Menu		PF12---
Help Add E		!			Menu

Resources are defined by name, type and quantity. There are two types of resource: reusable and non-reusable. A reusable resource is released after job completion.

Dynamic JCL Generation

When defining a job within a network, the user can specify that its JCL is to be generated dynamically either at job activation time or at job submission time.

Dynamic JCL generation is achieved using the Natural MACRO facility, an extension of the Natural programming language. This facility consists of standard Natural statements and text strings (JCL frames). The text strings can contain Natural escape characters followed by variables that will be substituted by

their current value during dynamic generation.

These current values will be taken from the so called "Symbol Tables", which are user-defined tables defining the escape characters and variable names used in the MACRO facility, as well as the current values to be substituted. The symbol table to be used is specified in the job definition screen.

If any symbol specified in the dynamic JCL is not in the symbol table indicated for the job, the symbol is searched for at substitution time (either activation or execution) in the symbol table(s) belonging to owner SYSDBA. A user can define any number of entries in a single symbol table or any number of symbol tables.

Additionally, Entire Operations passes standard variables defined in the parameter section to the dynamically generated program, such as job owner, network name, current job name and original scheduling date. The same applies to Natural system variables such as DATE, TIME, USER, etc. As these parameters can be substituted in any part of the JCL, different JCL configurations can be generated depending on time, date, user ID etc.

Entire Operations provides dynamic JCL generation for all supported platforms (OS/390, VSE/ESA, BS2000/OSD, UNIX) as shown in the following examples:

- Example 1: Dynamic JCL in an OS/390 Environment
- Example 2: Dynamic JCL in a BS2000/OSD Environment
- Example 3: Dynamic JCL in a UNIX Environment

Example 1: Dynamic JCL in an OS/390 Environment:

The following is the symbol table specified for the MACRO program:

Symbol Name	Current Value
STEPLIB	SN.SYSF.SOURCE
CLASS	G

The variable from the parameter section is assumed to have the following value:

P-OWNER	NET1
---------	------

The system variables are assumed to have the following values:

*TPSYS	COMPLETE
*DEVICE	BATCH
*INIT-USER	SN

The following is a Natural MACRO program including a parameter section and JCL with the Natural escape character (paragraph sign #) followed by variable names from the symbol table.

```
# DEFINE DATA PARAMETER USING NOPXPL-A
# LOCAL /* MUST BE CODED
# END-DEFINE
//SNMAC4 JOB ,#P-OWNER,MSGCLASS=X,CLASS=#CLASS //STEP01 EXEC
PGM=NOPCONTI,PARM='C0004' //STEPLIB DD DISP=SHR,DSN=#STEPLIB
/* DEVICE: *DEVICE, INIT-USER: *INIT-USER /* TPSYS: *TPSYS
# IF CLASS = 'G'
/* THE MSGCLASS IS REALLY 'G'
# ELSE
/* ANOTHER MSG-CLASS FOUND
# END-IF
/*
```

The resulting dynamically generated JCL will be:

```
//SNMAC4 JOB ,NET1,MSGCLASS=X,CLASS=G
//STEP01 EXEC PGM=NOPCONTI,PARM='C0004' //STEPLIB DD
DISP=SHR,DSN=SN.SYSF.SOURCE /* DEVICE: BATCH, INIT-USER: SN
/* TPSYS: COMPLETE
/* THE MSGCLASS IS REALLY 'G'
/*
```

Example 2: Dynamic JCL in a BS2000/OSD Environment:

The fields taken from the DB-INFO are assumed to have the following values after the FIND statement:

Field	Value
NUCLEUS	055
LP1	1000
NU1	100
ACCOUNT	EXAMPLE
NH1	4000
MSG	FHL
VERSION	524

The variables taken from the parameter section have the following current values:

Variable	Value
P-OWNER	OS
P-JOB	NUC055
P-EXECUTION-NODE	055

No symbol table was defined for this example job.

The following is the example JCL written using the Natural MACRO facility, including variables to be substituted from the DB-INFO view and the parameter section. Variables are preceded by the escape character paragraph sign #.

```

# DEFINE DATA PARAMETER USING NOPXPL-A
# 1 L-JOB
# 1 REDEFINE L-JOB
# 2 L-JOB-A (A3)
# 2 L-JOB-NUC (N3)
# LOCAL /* LOCAL VARIABLES START HERE
# 1 DB-INFO VIEW OF DB-INFO
# 2 NUCLEUS
# 2 LP1
# 2 NU1
# 2 ACCOUNT
# 2 NH1
# 2 MSG
# 2 VERSION /* E.G. 524
# 1 LWP (N7)
# 1 NUC (N3)
# 1 SPOOL (A10) INIT <'NOSPOOL'>
# END-DEFINE
# *
# MOVE P-JOB TO L-JOB-A
# MOVE P-EXECUTION-NODE TO NUC
# F1. FIND DB-INFO WITH NUCLEUS = NUC
/.NUC NUC LOGON #P-OWNER,#ACCOUNT
/OPTION MSG=#MSG
/REMARK
/REMARK NUCLEUS #NUC
/REMARK
/SYSFILE SYSLST = NUC NUC..LST.NUC
/SYSFILE SYSDTA = SYSCMD
/FILE ADA VERSION..MOD,LINK=DDLIB
/FILE *DUMMY,LINK=DDLOG
/FILE *DUMMY,LINK=DDSIBA
/FILE ADA NUC..ASSO,LINK=DDASSOR1,SHARUPD=YES
/FILE ADA NUC..DATA,LINK=DDDATAR1,SHARUPD=YES
/FILE ADA NUC..WORK,LINK=DDWORKR1,SHARUPD=YES
/EXEC (ADARUN,ADA VERSION..MOD)
# COMPUTE LWP = F1.LP1 * (F1.NU1 + 100)
ADARUN PROG=ADANUC,LP=F1.LP1,LU=65535,LWP=#LWP ADARUN
DB=#NUC,NU=#NU1,NC=20,TT=600,TNAE=1800 ADARUN NH= NH1
/SYSFILE SYSLST = (PRIMARY)
/SYSFILE SYSDTA = (PRIMARY)
/SYSFILE SYSOUT = (PRIMARY)
/LOGOFF SPOOL
# END-FIND

```

The resulting dynamically generated JCL will be:

```

/.NUC055 LOGON OS,EXAMPLE
/OPTION MSG=FHL
/REMARK
/REMARK  NUCLEUS 055
/REMARK
/SYSFILE  SYSLST = NUC055.LST.NUC
/SYSFILE  SYSDTA = SYSCMD
/FILE  ADA524.MOD,LINK=DDLIB
/FILE  *DUMMY,LINK=DDLOG
/FILE  *DUMMY,LINK=DDSIBA
/FILE  ADA055.ASSO,LINK=DDASSOR1,SHARUPD=YES
/FILE  ADA055.DATA,LINK=DDDATAR1,SHARUPD=YES
/FILE  ADA055.WORK,LINK=DDWORKR1,SHARUPD=YES
/EXEC  (ADARUN,ADA524.MOD)
ADARUN  PROG=ADANUC,LP=1000,LU=65535,LWP=200000 ADARUN
DB=055,NU=100,NC=20,TT=600,TNAE=1800 ADARUN NH=4000
/SYSFILE  SYSLST = (PRIMARY)
/SYSFILE  SYSDTA = (PRIMARY)
/SYSFILE  SYSOUT = (PRIMARY)
/LOGOFF  NOSPOOL

```

Note:

Any JCL generated at activation time using the MACRO language can be modified by the user until the job is actually submitted. Of course this modification is valid only for the actual network run.

Example 3: Dynamic JCL in a UNIX Environment:

The following example illustrates dynamic symbol replacement within a Bourne shell script (escape character \$):

```

#
# Bourne shell script for checking the number of users
# entered in /etc/passwd.
# If more than $USER-LIMIT entries appear,
# the script will be ended with exit 1.
#
#!/bin/sh
set -x
USER_COUNT='wc -l < /etc/passwd'
echo Number of users on node 'hostname' : $USER_COUNT
if test $USER_COUNT -gt $USER-LIMIT
then
    echo USER_COUNT_WARN
    exit 1
else
    echo USER_COUNT_OK
fi

```

The symbol table to be used should appear as follows:

Symbol Name	Current Value
USER-LIMIT	100

The result is the following executable shell script:

```
#
# Bourne shell script for checking the number of users
# entered in /etc/passwd.
# If more than 100 entries appear,
# the script will be ended with exit 1.
#
#!/bin/sh
set -x
USER_COUNT='wc -l < /etc/passwd'
echo Number of users on node 'hostname' : $USER_COUNT
if test $USER_COUNT -gt 100
then
    echo USER_COUNT_WARN
    exit 1
else
    echo USER_COUNT_OK
fi
```

Note:

Any JCL generated at activation time using the Natural MACRO language can be modified by the user until the job is actually submitted. Of course this modification is valid only for the actual network run.

Editing System Objects

The editing facility provided by Entire Operations allows the user to create, browse or edit any of the following objects:

- JCL of jobs, either in the master data base (and thus from any external storage source) or in the active data base. Changes to the JCL of active jobs are valid for the current run only and do not affect the master data base;
- Natural programs and user routines;
- Online documentation of a network, job or an event within a job (scratch pad information);
- Job protocol (for browsing only);
- Job output (for browsing only).

This enables the Entire Operations user to process such various data like, for example, UNIX scripts, CA-LIBRARIAN files or LMS files with a single Editor.

The user can enter the Editor by selecting the editing option in the maintenance screens of the appropriate object.

The Editor provides a comprehensive range of OS/390/ISPF-like commands appropriate to the particular object to be edited. For example, the STOW command will save and catalog a Natural program, but should not be used when editing a Natural MACRO program. A Natural MACRO program is saved, pre-processed and cataloged by the MACRO command.

Other word processing facilities provided by the Editor include centering, physical and logical tabulation, and text overlay.

A comprehensive online help is available by pressing a PF key from within the editing facility.

Generating Reports

Entire Operations provides a wide range of reports to assist operations at all levels. The user can enter the reporting facility either by selecting the reporting option from the Entire Operations Main Menu or by issuing the REPORT direct command from any system screen. 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, information 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 prose 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 be done either 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 in order to get informations for forecasting and planning purposes.

All reports are available online, their data can be printed as well. With this in easy way of long-time documentation is given.

Using Entire Operations Import/Export Utility the contents of the master data base can be unloaded to a sequential file too. Being intended for migration and transport purposes you can use this feature as well to build up your own reporting system.