

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
- Reserved Condition Names

For more detailed information, see the sections Logical Conditions, Input Condition Maintenance and End-of-Job Checking and Actions in the Entire Operations User's Guide.

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.

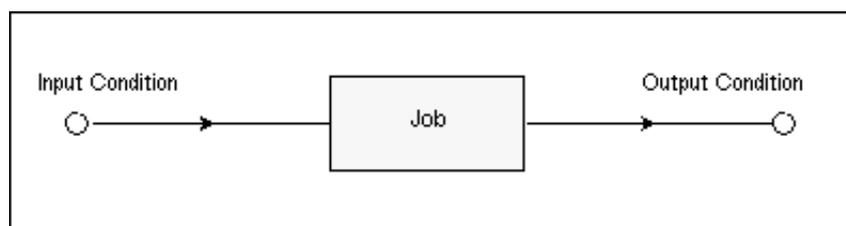
When a job network is activated, each logical condition is assigned a run number. This run number enables Entire Operations to distinguish between the same event that occurs during different network activations.

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

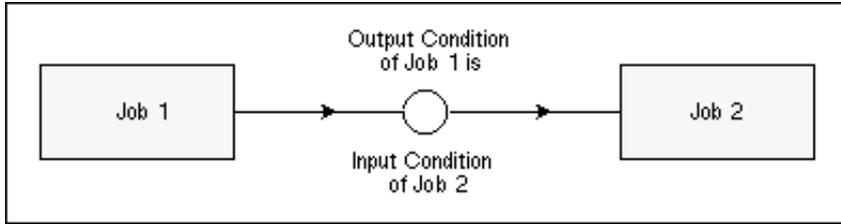


All input conditions must be fulfilled before a job can be submitted (prerequisite condition). You can define any number of input conditions for a job.

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

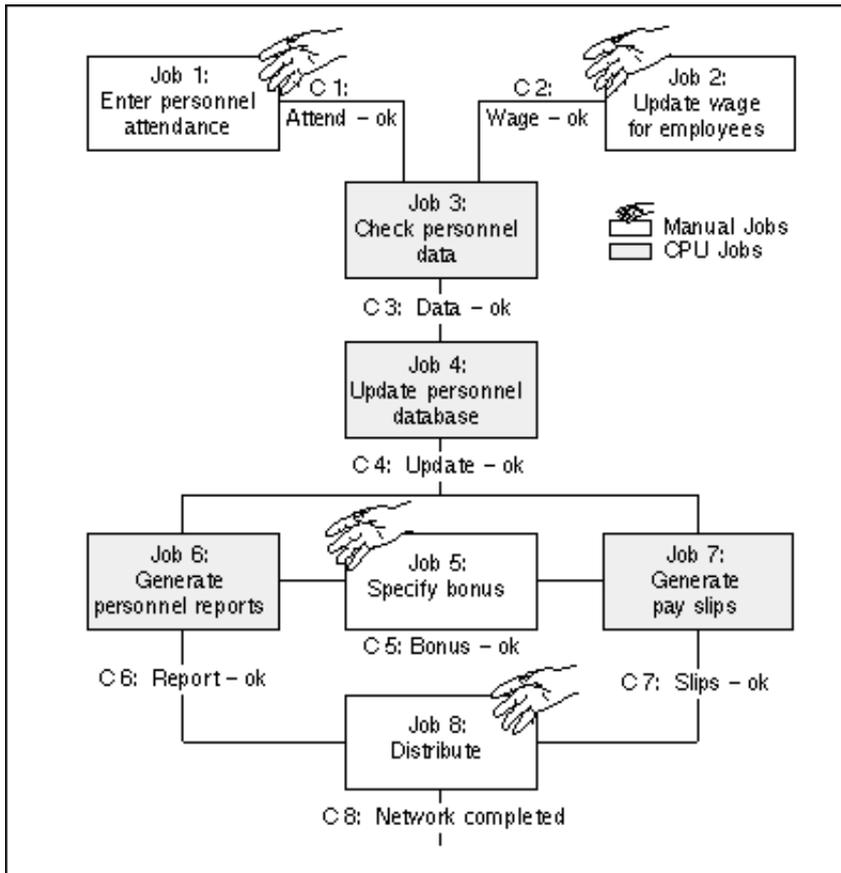


A specified event occurred as a result of **Job 1**. This sets the condition, signaling to Entire Operations that **Job 2** can be started.

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.

Reserved Condition Names

Some condition names may not be used for "common" conditions:

Condition	Explanation
NET-BEGIN NET-END NET-END-NOTOK	Used for sub-network control (see sub-networks)
NET-END-OK	To override the automatic detection of "network ended ok", you must set the reserved condition NET-END-OK at least once in your network. The whole network will then be treated as "ended ok", regardless of the real results of the jobs. This has an impact on the release of resources, which are allocated on the network level. You can force the resource release by setting this condition. See: Periods of Resource Allocation.