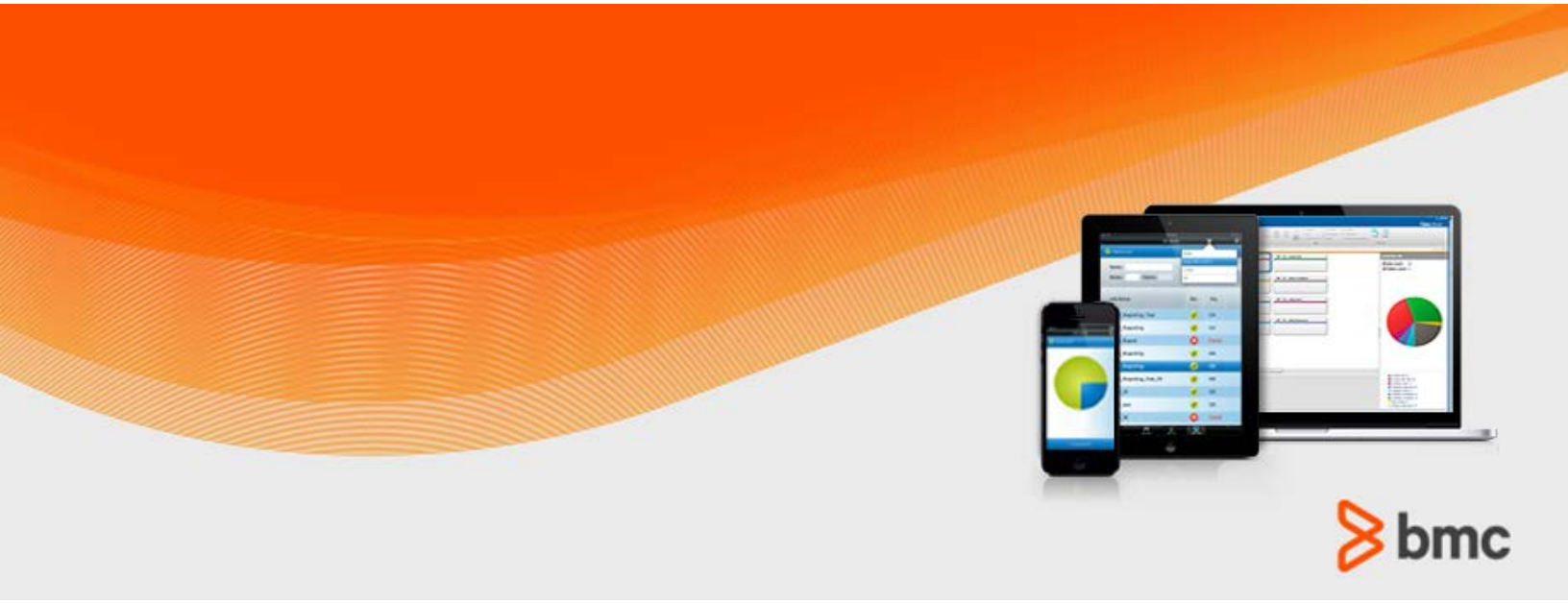
Control-M Workload Automation 9.0.00.200 Parameter Guide



May 2016

Contacting BMC Software

You can access the BMC Software website at http://www.bmc.com. From this website, you can obtain information about the company, its products, corporate offices, special events, and career opportunities.

United States and Canada

Address **BMC SOFTWARE INC**
2103 CITYWEST BLVD
HOUSTON TX
77042-2827
USA

Telephone ▪ **713 918 8800** ▪ **800 841 2031** **Fax** **713 918 8000**

Outside United States and Canada

Telephone **(01) 713 918 8800** **Fax** **(01) 713 918 8000**

For BMC Control-M Products that are licensed on the “per CPU – Full Capacity” unit of measurement and installed in an Amazon Web Services (“AWS”) or Microsoft Azure (“Azure”) cloud environment, a license is required for the total number of CPUs in each AWS or Azure instance upon which the Product is installed or which the Product manages, either remotely or locally. For AWS, one CPU is equivalent to one vCPU, as defined by AWS. For Azure, one CPU is equivalent to up to four Virtual Cores (as defined by Azure), rounded up to the closest multiple of four.

Restricted rights legend

Customer support

You can obtain technical support by using the BMC Software Customer Support website or by contacting Customer Support by telephone or e-mail. To expedite your inquiry, see "Before contacting BMC."

Support website

You can obtain technical support from BMC 24 hours a day, 7 days a week at http://www.bmc.com/support. From this website, you can:

- Read overviews about support services and programs that BMC offers
- Find the most current information about BMC products
- Search a database for issues similar to yours and possible solutions
- Order or download product documentation
- Download products and maintenance
- Report an issue or ask a question
- Subscribe to receive proactive e-mail alerts when new product notices are released
- Find worldwide BMC support center locations and contact information, including e-mail addresses, fax numbers, and telephone numbers

Support by telephone or e-mail

In the United States and Canada, if you need technical support and do not have access to the web, call 800 537 1813 or send an e-mail message to customer_support@bmc.com. (In the subject line, enter **SupID:<yourSupportContractID>**, such as SupID:12345). Outside the United States and Canada, contact your local support center for assistance.

Before contacting BMC

Have the following information available so that Customer Support can begin working on your issue immediately:

- Product information
  - Product name
  - Product version (release number)
  - License number and password (trial or permanent)
- Operating system and environment information
  - Machine type
  - Operating system type, version, and service pack or other maintenance level such as PUT or PTF
  - System hardware configuration

  - Serial numbers
  - Related software (database, application, and communication) including type, version, and service pack or maintenance level
- Sequence of events leading to the issue
- Commands and options that you used
- Messages received (and the time and date that you received them)
  - Product error messages
  - Messages from the operating system, such as `file system full`
  - Messages from related software

License key and password information

If you have questions about your license key or password, contact BMC as follows:

- (USA or Canada) Contact the Order Services Password Team at 800 841 2031, or send an e-mail message to ContractsPasswordAdministration@bmc.com.
- (Europe, the Middle East, and Africa) Fax your questions to EMEA Contracts Administration at +31 20 354 8702, or send an e-mail message to password@bmc.com.
- (Asia-Pacific) Contact your BMC sales representative or your local BMC office.

Third party Software

For the provisions described in the BMC License Agreement and Order related to third party products or technologies included in the BMC Product, see https://docs.bmc.com/docs/display/workloadautomation/Control-M+Workload+Automation+Documentation and click **Third-party software (TPS).**

Contents

1

Introduction to parameters

In the Control-M job scheduling environment, jobs are tasks that can be performed by a computer. These tasks are handled according to parameters. The parameters specified for a particular job are collectively referred to as a *job processing definition*.

NOTE: The term *job* refers to any task that can be performed by a script or command that is handled by Control-M/Server.

Job processing parameters only need to be defined once for each job. Control-M uses the job processing definition each time a job is ordered. Definitions can be modified at any time using various Control-M/EM facilities.

These definitions are created using Control-M Workload Automation and are stored in folders (job directories) according to various criteria that you determine. The folders are saved to the Job Definitions database. When the scheduling criteria of the folder are met, the jobs are passed to the Active Jobs database.

Job parameters are composed primarily of static information. However, a job processing definition may need to include information that is subject to change or was not available when the definition was created (such as the time at which a job was submitted, or the name of the batch queue to which the job was submitted.). Control-M Workload Automation enables you to replace that data with variables. You can use both pre-defined variables and those that you create.

Language support

Western European language special characters can be specified for most parameters in which free text can be entered. A list of all parameters that can contain these characters is provided in *Control-M Language Customization.*

The term "non-English characters" in the Invalid Characters section of each parameter description indicates that the parameter only supports Western characters.

The following characters are not supported for any parameter under any circumstances:

€ ‚ ƒ „ … † ‡ ˆ ‰ Š ‹ Œ Ž
‘ ’ “ ” • – — ˜ ™ š › œ ž Ÿ

Conventions for parameter descriptions

The following terms are used to describe the format of parameters:

- **Usage**
  Indicates whether a parameter is mandatory or optional.

  Some parameters are mandatory only when Control-M is installed on a certain computer. Some parameters are mandatory only when a related parameter is specified (for example, **Partition data set** and **Minimum number of tracks** must be specified together).

- **Case sensitive**
  Indicates that Control-M differentiates between values specified using various combinations of uppercase and lowercase letters.

  For example, the prerequisite condition JOB_ENDED_OK is different from the prerequisite condition Job_Ended_OK. A job waiting for JOB_ENDED_OK is not submitted if only Job_Ended_OK is found.

- **Invalid Characters**
  Indicates, when specifying the parameter, whether there are any special characters that cannot be used, embedded blanks (spaces) can be used trailing blanks (after the specified value) are ignored.

- **Variable Support**
  Indicates whether variables can be included in the value of the specified parameter.

Abbreviations and conventions

The following abbreviations are used:

Abbreviation	Description
Control-M/EM	Control-M/Enterprise Manager

The following conventions are used:

Convention	Description
key	When describing keystrokes, boldface type is used for the name of a key (for example, **F1**). When two keys are joined with "**+**" as in **Shift**+**F1**, hold down **Shift** while pressing **F1**.
Menu => Option	This represents an option selection sequence. **EXAMPLE:** Users and Groups => Groups => Add means that you first select **Users and Groups** from the menu bar. Select the **Groups** option from the submenu. Finally, select the **Add** option from the **Groups** submenu.
{ } (braces)	Braces indicate that at least one of the enclosed elements is required. **EXAMPLE: {***fileName* **|** *deviceName***|** *mediaType***}** means that you must specify one of the variables.
{Option A|Option B}	The vertical bar is used to separate choices. For example: **{AND|OR}** means that you specify either AND or OR.
[Option]	Square brackets are used to enclose parameters that are optional.
Code Samples	Format syntax, operating system terms, examples, and JCL scripts are presented in this typeface.
Messages	Messages are presented in this typeface.
Boldface	In instructions, **boldface** type highlights information that you enter. File names, directory names and paths, parameter names, and options are also in boldface.

Convention	Description
Option Symbol	A vertical bar (|) separating items indicates that you must choose one item. In the following example, you would choose a, b, or c: **EXAMPLE:** a | b | c

General parameters

General parameters contain basic information that identifies the job, describes what it does, and gives the location of the job script.

Parameter	Description
Job Type (on page 17)	Defines the type of job that determines which fields are exposed in a job definition. The value of the job type is usually based on the platform for the job run. Jobs defined as Dummy provide instructions for Control-M such as post-processing actions.
Job Name (on page 19)	Defines the name of the job processing definition and appears in the job definition and tracking displays, and enables you to identify the job, and order the job.
Description (on page 21)	(From Forecast only) Provides a description of the job in free text. A well written description can help you determine why the job was defined and how it fits into your business workflow.
What (on page 22)	Defines one or more parameters which determines what the job runs.
Command (on page 22)	Indicates an operating system command line entry to be submitted as a job. Use this parameter to specify an operating system command to execute by Control-M. The command must be specified exactly as it would be specified in a terminal for the specific computer.
Embedded Script/Embedded JCL (on page 24)	Defines a script exactly as it would be specified in a terminal for the specific computer and is part of the job definition.
Host (/Group) (on page 30)	Defines the name of a Control-M/Agent computer, remote host computer, or host group where the job is submitted.
Control-M/Server (on page 31)	Defines the name of the Control-M/Server (or Control-M for z/OS) that processes the job.
Run as (on page 31)	Identifies the user name with the authorization to execute the job. This parameter is used by the Control-M security mechanism.
Override Path (on page 32)	Specifies a temporarily-modified job script file without changing the original script file in the File Path/Member library (on page 25) and without changing the scheduling order of a folder.

Parameter	Description
Pre-execution (on page 33)	Specifies a command to run immediately before running the job defined by the What (on page 22) parameter. The return code is ignored.
Post-execution (on page 34)	Specifies a command to run immediately after running the job defined by the What (on page 22) parameter. The return code is ignored.
Run job on all hosts in group (on page 34)	Specifies that job submission details be broadcast to all agents within a defined Host Group. All available agents in the Host Group run an identical job, and each such job has a unique Order ID.
Run as Detached (on page 48)	A regular job submitted to Control-M for execution as a background process. The results of the job (the output) are analyzed by the post-processing subsystem.
Variables (on page 36)	All variables are identified by the **%%** prefix. If **%%** is included in the value for a job processing parameter, Control-M assumes that it is referring to a variable or function.
Application (on page 37)	Provides a logical name for sorting groups of jobs. This parameter is used to supply a common descriptive name to a set of related job groups. The jobs do not necessarily have to run at the same time.
Sub Application (on page 38)	Indicates the name of the Sub Application where the job belongs logically. It is a sub-category of the Application parameter. For example, the Application is Finances, and the Sub Application is Payroll.
Created by (on page 39)	Indicates the Control-M/EM user who defined the job.
Priority (on page 44)	Determines the order of job processing by Control-M in the Active Jobs database.
Critical (on page 45)	Determines whether the job is a critical-path job in Control-M, which ensures resources allocation order.
Control-D Category (on page 48)	Defines the name of the Control-D Report Decollating Mission Category. If specified, the report decollating mission is scheduled whenever the job is scheduled under Control-M.
Doc Library/ Doc Path (on page 40)	For a z/OS job, Doc Library defines the name of the library where the Documentation (description) is saved. For a non-z/OS job, Doc Path defines the name of the file path where the Documentation is saved.

Parameter	Description
Doc Member/ Doc File (on page 42)	For a z/OS job, defines the name of the member where the job Documentation (description) is saved. For a non-z/OS job, the Doc File is the name of the file where the job Documentation is saved.
URL (on page 43)	Defines the URL address where the documentation is located.
File name/Member name (on page 28)	Indicates the name of the file that contains the job script, or for z/OS jobs, the name of a member that contains one of the following in relation to the job to be executed: ▪ The JCL of the job ▪ The started task procedure ▪ Warning messages
Run as started task (on page 49)	Specifies that the job is invoked with the operator START command.
File Path/Member library (on page 25)	For non-z/OS jobs, File Path indicates the location of the file that contains the script. For z/OS jobs, Member Library indicates the location of the Member that contains the JCL, started task procedure, or Warning message.
Prevent NCT2 (on page 49)	Performs data set cleanup before the original job run.
Request NJE Node (on page 52)	Defines the node in the JES network where the job executes.
Scheduling Environment (on page 54)	Indicates the JES2 workload management scheduling environment that is to be associated with the job.
System Affinity (on page 61)	Indicates the identity of the system in which the job must be initiated and executed (in JES2).
Folder Library (on page 63)	Defines the name of the library that contains the job's folder.
Folder Name (on page 65)	Defines the name of the folder. In the Properties pane this parameter indicates the folder where the job belongs.
Folder Type (on page 63)	Indicates whether the folder type is a regular folder, a SMART folder, or a Sub Folder.

Parameter	Description
Order Method (on page 55)	Defines the method for ordering the entity as one of the following: ▪ **Automatic (Daily):** When set to Automatic, at the same time each day (known as New Day time), each Control-M/Server runs a procedure called New Day. This procedure performs a number of tasks, including scheduling the day's jobs, and running maintenance and cleanup utilities. The New Day procedures orders the folder or folder jobs. ▪ **None (Manual Order):** The folder is not automatically ordered. ▪ **Specific User Daily:** Identifier used to assign the folder to a specific User Daily job. The User Daily name (on page 61) is ordered at a specific time of the day. For load balancing purposes, the User Daily jobs are scheduled for different times, throughout the day, other than the New Day time.
User Daily name (on page 61)	Defines User Daily jobs whose sole purpose is to order jobs. Instead of directly scheduling production jobs, the New Day procedure can schedule User Daily jobs, and those User Daily jobs can schedule the production jobs. Set User Daily Name when Order Method is set to Specific User Daily.
Emergency Job (on page 61)	Determines whether the z/OS job is an Emergency job.
From program step (on page 66)	Defines the first program step for a job to begin at when it is restarted.
From procedure step (on page 66)	Defines the first procedure step for a job to begin at when it is restarted.
To program step (on page 66)	Defines the last program step for a job to stop running.
To procedure step (on page 67)	Defines the last procedure step for a job to stop running.
Service Name (on page 67)	Logical name, from a user or business perspective, for the critical service. BMC recommends that the service name be unique.
Service must complete (on page 68)	Defines the time the critical service must be completed to be considered on time. The time can be specified either by a specific hour and/or day or on a specific hour and minute after the order time.

Parameter	Description
Service Priority (on page 69)	Defines the priority level of this service, from a user or business perspective.
Batch Impact Manager Service Actions (on page 56)	Defines the automatic interventions, such as rerunning a job, displaying the critical service in BMC Service Impact Manager, or extending the service due time. The action is performed if the job finished too quickly, there is a job failure on service path, or the service finished late or too quickly.
Job Run-Time Tolerance (on page 69)	Defines the type of deviation from the average completion time for a job in the service. If the run time falls within the tolerance set, it is considered on time, otherwise it has run too long or ended too early. The values are Percentile range or Average Run Time (in percentage or by minutes).
Percentile range (on page 70)	Defines the number of standard deviations of job run in this service. Percentile Range is the default method for specifying Job Run-Time Tolerance.
Average run time +/- (on page 71)	The percentage of the time (based on the average run time for the job) or the number of minutes that the job can be early or late, and still be considered on time.

Job Type

Defines the type of job that determines which fields are exposed in a job definition. The value of the job type is usually based on the platform for the job run. Jobs defined as Dummy provide instructions for Control-M such as post-processing actions.

Additional information	Description
Usage	Mandatory
Format	Select the required value from the Job Type group.

The appropriate job template with Job Type must be selected from the job palette, or you can change the Job Type if you have not yet saved the job. The following **Job Types** identify the relevant job fields for jobs that run on particular platforms or for jobs that perform Control-M actions:

- **OS:** Defines job fields for Control-M/Server Distributed jobs that run on the UNIX or Windows platform. For more information on OS fields, see OS.
- **z/OS:** Defines job fields for Control-M for z/OS jobs that run on the Mainframe platform. For more information on z/OS fields, see z/OS.
- **BIM (BMC Batch Impact Manager):** Defines job fields for defining a batch service. You can define and order a BIM job to represent batch tasks that can seriously impact critical business services if delayed, so BMC Batch Impact Manager can provide early warning. For more information on BIM fields, see BIM.
- **Dummy:** Defines Dummy job parameters which provide instructions to Control-M, but do not run any script or command. For more information on Dummy fields, see Dummy.
- **Dummy** (z/OS): Defines Dummy job parameters which provide instructions to Control-M, but do not run any script or command. For more information on z/OS Dummy fields, see z/OS Dummy.
- **File Watcher:** The File Watcher job type defines the ctmfw (Control-M File Watcher) utility job to monitor file status and detect file processes, as described in ctmfw File Watcher utility. For more information on File Watcher fields, see File Watcher.
- **Control-M Report:** Defines Control-M Report job fields which enables you to schedule jobs that execute the emreportcli utility to generate a report based on a format using a template you define in the Control-M Reporting Facility. This enables you to schedule the generation of reports according to scheduling criteria in batch mode. For more information on Control-M Report fields, see Control-M Report.
- **OS2200:** Supports running jobs on other platforms, and enables you to submit jobs for execution on the agent platform, monitor the jobs, and perform post-processing analysis of output files. The completion status of jobs and the results of post-processing analysis are transmitted to the Control-M/Server.

  For more information on Other Control-M/Agents, see Control-M/Agent and remote hosts.
- **MAPPER:** Supports running jobs on other platforms, and enable you to submit jobs for execution on the agent platform, monitor the jobs, and perform post-processing analysis of output files. The completion status of jobs and the results of post-processing analysis are transmitted to the Control-M/Server.

  For more information on Other Control-M/Agents, see Control-M/Agent and remote hosts.
- **OS/400 <type>:** Supports running jobs on other platforms, and enable you to submit jobs for execution on the agent platform, monitor the jobs, and perform post-processing analysis of output files. The completion status of jobs and the result of post-processing analysis are transmitted back to the Control-M/Server. You can view and modify the following OS/400 type of jobs: External, Full, Multiple Commands, Program, VT.

  For more information on Other Control-M/Agents, see Control-M/Agent and remote hosts.
- **TANDEM Full:** Supports running jobs on other platforms, and enables you to define jobs, retrieve job parameters, and control jobs running on the Guardian operating system.

  For more information on Other Control-M/Agents, see Control-M/Agent and remote hosts.

- **<Application Plug-in Jobs>:** Defines job fields for Application Plug-ins such as SAP, Oracle E-Business Suite, or IBM Cognos.

Job Name

Defines the name of the job processing definition and appears in the job definition and tracking displays, and enables you to identify the job, and order the job.

Additional information	Description
Usage	Mandatory
Length	1-64 characters z/OS: 1-8 characters
Case Sensitive	Yes
Invalid Characters	▪ Single quotation marks ▪ "$", "/", "*", "?", " " .
Variable Name	%%JOBNAME
Alternate Names	▪ **Control-M/EM Utilities:** JOBNAME ▪ **Reporting Facility:** JOB NAME ▪ **Control-M/Server Utilities:** -jobname ▪ **Control-M for z/OS:** JOB NAME ▪ **Control-M/EM API:** job_name
Computer specific information	
z/OS	The Job Name parameter must be unique in the folder.
iSeries (AS/400)	The value specified for this parameter is the actual job name to be used by **iSeries (AS/400)**. It is part of the job submission command. **NOTE:** For jobs on an **iSeries (AS/400)** computer, the value specified for the **Job Name** parameter must conform to **iSeries (AS/400)** conventions for job names.

The **Job Name** can also be displayed in the job node displayed in the Control-M/EM window (depending upon options specified in the Display Net window).

This parameter is used when ordering or forcing a job, either using the Order Job (on page 154) parameter or when using the **Order/Force** windows. You can define a job without a job name in ctmcreate and ctmdefine.

Job naming standards are a must for every successful implementation of Control-M as the design phase of Control-M is predicated on solid, enforceable naming standards.While there is no set standard for job naming, a good rule-of-thumb is that all jobs start with the application moniker. Next, a few characters to describe the job's function may be included and finally a few characters to describe the specific purpose, destination or process the job performs.

EXAMPLE: AAA – for application moniker, such as DDA, SAV, MTG, LOA

TTT – for job type, such as AFT, SAP, WIN, UNX, WJM, DBA

FFFFFFF – such as POSTING, BACKUP, DBLOAD

Well-thought naming conventions are the basis for identifying the job and its function; managing access for security, change and problem management as well as reporting. Using this job naming format, Control-M access control can be designed around the job name. Control-M security can restrict or allow access to the application based on the user's role and responsibility.

For change management, users authorized to access DDA in a read-only mode can be defined, where as another user can be given full access to define, manage and monitor DDA jobs but not restart them and still, a third user can be given access to monitor and restart/override jobs but not update the jobs' definitions.

Accessing/modifying the Job Name in Variable expressions

The value of the **Job Name** parameter can be accessed using the %%JOBNAME variable. For example, this name can be included in a message that is sent using the Notify (on page 156) parameter.

The **Job Name** parameter can also be overridden when the job is ordered, for example, by using the following statement in the command line of the ctmorder utility for Control-M/Server.

EXAMPLE:-variable %%JOBNAME *newjobname*

Description

(From Forecast only) Provides a description of the job in free text. A well written description can help you determine why the job was defined and how it fits into your business workflow.

Additional information	Description
Usage	Optional.
Length	1-4000 characters
Case Sensitive	Yes
Variable Name	None
Alternate Names	▪ **Control-M/EM Utilities:** DESCRIPTION ▪ **Reporting Facility:** DESCRIPTION ▪ **Control-M/Server Utilities:** -description ▪ **Control-M for z/OS:** Description ▪ **Control-M/EM API:** description

Run as dummy

Enables you to run jobs of job types other than Dummy, as dummy jobs.

Additional information	Description
Usage	Optional
Default	unchecked
Format	Check box
z/OS	Override Path (on page 32) parameter is changed to DUMMY.

If you choose to clear the **Run as dummy** check-box, the job is restored to its' original job type. For z/OS, the job is restored to its' original Override Path and UseInstreamJCL.

What

Defines one or more parameters which determines what the job runs.

Additional information	Description
Usage	Mandatory
Format	Drop-down list
Variable name	None
Alternate Names	▪ **Control-M/EM Utilities:** TASKTYPE ▪ **Reporting Facility**: TASK_Type ▪ **Control-M/Server Utilities:** -tasktype ▪ **Control-M/EM API:** task_type
Computer specific information	▪ OS: Embedded Script (on page 24), Command (on page 22), Script (on page 24) ▪ z/OS: Member (on page 28), Embedded JCL (on page 24)
Previously Known As	Tasktype

Command

Indicates an operating system command line entry to be submitted as a job. Use this parameter to specify an operating system command to execute by Control-M. The command must be specified exactly as it would be specified in a terminal for the specific computer.

NOTE: This parameter is not relevant in z/OS environments.

Additional Information	Description
Usage	This parameter can be used only if the **What** parameter is **Command**. In this case, the **Command** parameter is required.
Length	1 through 512 characters
Case Sensitive	▪ Microsoft Windows: No ▪ UNIX: Yes
Invalid Characters	None
Variable Name	Yes. A Variable or expression can be specified as all or part of the value for this parameter.
Alternate Names	▪ **Control-M/EM Utilities:** CMDLINE ▪ **Reporting Facility**: CMD LINE ▪ **Control-M/Server Utilities:** -cmdline ▪ **Control-M/EM API:** command
Computer specific information	▪ Commands submitted to a UNIX computer are executed using the Bourne shell. ▪ For Control-M/Agent for **iSeries (AS/400)** version 2.1.3,commands must not include single quotation marks (‘ ‘).If quotation marks are necessary, use double quotation marks (**" "**) only. ▪ For Windows agents: When specifying variables in the **CMDLINE** or **COMMAND** parameters on Control-M/Agent for Windows, the variable prefix must be specified as **% % % %** instead of **% %**.

Control-M creates a temporary file that contains the command. This file is submitted for execution, and is monitored and analyzed in the same manner as a regular job.

The command can include any combination of text and variables. However, the length of the command after resolution of variables must not exceed **999** characters. For more information about variables, see Control-M Variable facility (on page 229).

EXAMPLE: OpenVMS

```
BACKUP DUA0:[000000...] MUA0:MONBKP/SAVE
```

UNIX

```
tar cvf /dev/rmt0 -c /home
```

iSeries (AS/400)

SAVLIB LIB(LIB_1 LIB_2) DEV(TAP02)

Using variables in a command)

The command in this job uses the Control-M ctmcontb utility to delete all prerequisite conditions that are more than five days old.

Variable Assignment:

%%A=%%CALCDATE %%DATE -365

%%B=%%CALCDATE %%DATE -5

Command:ctmcontb deletefrom %%A %%B

Embedded Script/Embedded JCL

Defines a script exactly as it would be specified in a terminal for the specific computer and is part of the job definition.

Additional information	Description
Usage	This parameter can be used only if the **What** parameter is **Embedded script/Embedded JCL**.
Length	64000 B
Case Sensitive	Yes
Variable Name	No. A variable or expression cannot be specified as all or part of the value for this parameter. Instead %%PARM can be used to pass parameters to the embedded script. See Job submission variables (on page 232).
Alternate Names Embedded Scripts/JCL (flag)	▪ **Control-M/EM Utilities:** USE_INSTREAM_JCL ▪ **Reporting Facility:** use In-stream Jcl ▪ **Control-M/Server Utilities:** (none) ▪ **Control-M for z/OS:** INSTREAM JCL ▪ **Control-M/EM API:** use_instream_jcl
Alternate Names Embedded Scripts/JCL	▪ **Control-M/EM Utilities:** INSTREAM_JCL ▪ **Reporting Facility:** In-stream Jcl ▪ **Control-M/Server Utilities:** embedded_script ▪ **Control-M for z/OS:** INSTREAM JCL ▪ **Control-M/EM API:** instream_jcl

File Path/Member library

For non-z/OS jobs, File Path indicates the location of the file that contains the script. For z/OS jobs, Member Library indicates the location of the Member that contains the JCL, started task procedure, or Warning message.

The path or library specified in the **File Path/Member library** parameter and the accompanying File name/Member name (on page 28) do not have to exist when the job processing parameters are defined. Control-M searches for them only before actual submission of the job.

Additional information	Description
Usage	This parameter can be used when the **What** parameter is Embedded Script/Embedded JCL (on page 24). In this case, the **Script** parameter is required. For z/OS, this parameter can be used when the **What** parameter is **Member**.
Length	▪ Microsoft Windows, UNIX, OpenVMS: 1-255 characters ▪ **iSeries (AS/400)**: 1-10 characters ▪ z/OS: 1-44 characters
Case Sensitive	Yes
Invalid Characters	▪ Blanks ▪ z/OS: Non-English characters ▪ If the job runs on any version of Microsoft Windows 2000, prohibited filename characters (such as \, /, or *)
Variable Name	%%MEMLIB
Alternate Names	▪ **Control-M/EM Utilities:** MEMLIB ▪ **Reporting Facility:** MEM LIB ▪ **Control-M/Server Utilities:** -memlib ▪ **Control-M for z/OS:** MEMLIB ▪ **Control-M/EM API:** mem_lib
Alternate Formats in other Control-M components	Alternate formats for the **File Path** parameter are listed in the Computer specific information section in this folder.
Computer specific information	
Microsoft Windows	The **File Path** parameter indicates the drive ID, and the names of the directory and subdirectories (if any).

Additional information	Description	
UNIX	The **File Path** parameter indicates the names of the directory and subdirectories (if any). One of the following symbols can be used in place of or as part of the directory and sub-directory names (these symbols are resolved at the time the job is ordered or forced): ▪ **$HOME** resolves to the home directory of the job owner. ▪ **~<username>** resolves to the home directory of the specified UNIX user.	
OpenVMS	The **File Path** parameter contains the (device:[directory]) that can be specified as the physical path or as an OpenVMS logical name.	
iSeries (AS/400)	The **File Path** parameter contains the (device:[directory]) that can be specified as the physical path or as an OpenVMS logical name.	
z/OS	Format of the parameter depends on whether the job processing definition applies to a job (or warning messages) or a started task:	
	Job	Valid values: a valid data set name of 1-44 characters or one of the following reserved values: ▪ DUMMY - for dummy jobs ▪ USER= name - for user-defined libraries ▪ GENERAL - specifies the library referenced by DD statement DALIB in the Control-M procedure.

Additional information	Description	
	Started Task	Any of the following formats can be used for the value of Mem Lib: **. taskid*, where taskid is the ID of the task The started task is activated in the computer in which the Control-M monitor is active. *cpuid*, *stcparms*, where: — *cpuid* is the ID of the computer in which the started task is to be activated — *stcparms* is started task parameters *cpuid*, where *cpuid* is the ID of the computer in which the started task is to be activated.
		Valid values for *cpuid* are: * – The computer where the Control-M monitor is active.
z/OS (*Continued*)	Started Task (*Continued*)	**Under JES2** ▪ **Nn** – where n is the JES/NJE host ID. ▪ **Mm** – where m is the computer ID. ▪ **NnMm** – where n is the JES/NJE host ID, and m is the computer ID. **Under JES3** **Lname** – where name is the logical JES name of the computer, that is, the name as used in the JES3 command *T, not the SMF system ID.

You can browse for the member that contains the JCL code, as described in Browsing remotely for the member that contains the JCL in a library, in *Using Workload Automation Guide*.

EXAMPLE: Microsoft Windows

D:\ACCOUNT\SALARY

UNIX

$HOME/ctm/salary

OpenVMS

DUA0:[SALARY.JAN.ARCH],SALARY$DIR

iSeries (AS/400)

MYLIB

z/OS

GENERAL

File name/Member name

Indicates the name of the file that contains the job script, or for z/OS jobs, the name of a member that contains one of the following in relation to the job to be executed:

- The JCL of the job
- The started task procedure
- Warning messages

Additional information	Description
Usage	Mandatory if the **Script** option is selected from the **What** parameter.
Length	▪ Microsoft Windows, UNIX, OpenVMS: 1-64 characters ▪ **iSeries (AS/400)**: 1-21 characters ▪ z/OS: 1-8 characters
Case Sensitive	Yes
Invalid Characters	▪ Blanks, \, /, or * (Blanks and / can be used when defining an OS/400 job) ▪ z/OS: Non-English characters ▪ Character masks are not supported (for example, a job with a **FR*.EXE** File Name value is not executed).
Variable name	None
Alternate Names	▪ **Control-M/EM Utilities:** MEMNAME ▪ **Reporting Facility:** MEMNAME ▪ **Control-M/Server Utilities: -**memname ▪ **Control-M for z/OS:** MEMNAME ▪ **Control-M/EM API:** memname
Computer specific information	
Microsoft Windows	Name of a command file or an executable file.
UNIX	Name of a UNIX shell script file.

Additional information	Description
Open VMS	Name of the command file. It can be specified using one of the following formats: **Without file extension** (for example, JOBFILE): Control-M adds the appropriate extension according to the **What** parameter: **.COM** for **job** (batch job), **.EXE** for **Detached** (detached process). When the file name is specified without a version specification, the last version of the file is executed. **With full file extension**, including the version specification (for example, **JOBFILE.COM;2** or **JOBFILE.EXE;5**). Using the full file extension format, it is possible to select a specific file extension and version of the command file or detached process to be executed.
iSeries (AS/400)	An executable program object.
z/OS	Name of a member whose contents are determined by the **What** parameter. ▪ JCL of the job ▪ Started task procedure ▪ Warning messages

You can browse for the member that contains the JCL code, as described in Browsing remotely for the member that contains the JCL in a library, in *Using Workload Automation Guide*.

The **File Name/Member** can be the same as or different from the job name. Each member contains only one job.

EXAMPLE: Microsoft Windows

myjob.bat, myjob.cmd, myjob.exe

UNIX

myjob

OpenVMS

MYJOB, MYJOB.COM, MYJOB.COM;2, MYJOB.EXE, MYJOB.EXE;4.

iSeries (AS/400)

MYJOB

Host (/Group)

Defines the name of a Control-M/Agent computer, remote host computer, or host group where the job is submitted.

NOTE: This parameter is not relevant in z/OS environments.

Additional information	Description
Usage	Optional **NOTE**: If this parameter is left blank, the job is submitted for execution on the Control-M/Server computer.
Length	1 through 50 characters
Case Sensitive	No
Invalid Characters	Blanks; single quotation marks.
Variable name	None
Alternate names	▪ **Control-M/EM Utilities:** HOSTID ▪ **Reporting Facility:** HOST_ID HOSTGROUP ▪ **Control-M/Server Utilities:** -hostgrp ▪ **Control-M for z/OS:** (none) ▪ **Control-M/EM API:**host_group

EXAMPLE: Enable Control-M/Server to determine the most suitable agent

Assuming that a host group called UNIX_group contains agent computers bill and diana.

The following parameter causes Control-M/Server to determine which of the two agent computers in the group is best suited to execute the job when it is submitted for execution:

Host/Host Group UNIX_group

Control-M/Server

Defines the name of the Control-M/Server (or Control-M for z/OS) that processes the job.

Additional information	Description
Usage	Mandatory
Format	Drop-down list
Variable Name	None
Alternate Names	▪ **Control-M/EM Utilities:** DATACENTER ▪ **Reporting Facility**: DATACENTER ▪ **Control-M/Server Utilities:** Not supported. Jobs created with a Control-M/Server utility are submitted to the local Control-M/Server installation ▪ **Control-M for z/OS**: When a job is created, it is submitted to the local Control-M for z/OS installation ▪ **Control-M/EM API:** control-m

Run as

Identifies the user name with the authorization to execute the job. This parameter is used by the Control-M security mechanism.

The **Run as** parameter is used by the Control-M internal security mechanism to determine operations that each user is authorized to perform. For more information, refer to Control-M security for the appropriate computer.

Additional information	Description
Usage	Mandatory
Default	User name of the current Control-M/EM user
Length	▪ 1-30 characters ▪ z/OS: 1-8 characters
Case Sensitive	Yes

Additional information	Description
Invalid Characters	▪ Blanks ▪ Computers other than z/OS: Single quotation marks
Variable Name	None
Alternate Names	▪ **Control-M/EM Utilities:** RUN_AS ▪ **Reporting Facility:** RUN AS ▪ **Control-M/Server Utilities:** -owner ▪ **Control-M for z/OS:** OWNER ▪ **Control-M/EM API:** run as
Previously Known As	Owner

Override Path

Specifies a temporarily-modified job script file without changing the original script file in the File Path/Member library (on page 25) and without changing the scheduling order of a folder.

Additional information	Description
Usage	Optional
Length	▪ Microsoft Windows, UNIX, OpenVMS: 1 through 255 characters ▪ **iSeries (AS/400)**: 1 through 10 characters ▪ z/OS: 1 through 44 characters
Case Sensitive	Yes
Invalid Characters	Blanks If the job runs on any version of Microsoft Windows 2000, prohibited filename characters (such as \, /, or *). z/OS: Non-English characters. The contents of this field must not begin with the following strings: — GENERAL — USER=

Additional information	Description
Variable Name	%%OVERLIB
Alternate Names	▪ **Control-M/EM Utilities:** OVERRIDE_PATH ▪ **Reporting Facility:** OVERRIDE PATH ▪ **Control-M/Server Utilities:** -override_path ▪ **Control-M for z/OS:** OVERLIB ▪ **Control-M/EM API:** override path
Previously known as	Override Library

The **Override Path** parameter enables the user to submit a temporarily-modified job script file without changing the original script file in the File Path/Member library (on page 25) and without changing the scheduling order of a folder.

When to use

The library containing the job's regular script file is specified in the File Path/Member library (on page 25) parameter. When temporary modifications are required, the modified script file is placed in the location indicated by the **Override Path** parameter. If the file specified by the File name/Member name (on page 28) parameter is found in the **Override Path** location, this file is submitted instead of the job script file with the same name that resides in the File Path/Member library (on page 25) location.

Canceling override

The override can be canceled by one of the following methods:

- Delete the file specified in the File name/Member name (on page 28) parameter from the Override Path location. If the job script file is not found in Override Path, it is automatically taken from the File Path/Member library (on page 25) location.
- Delete the Override Path specification from the job definition.

Pre-execution

Specifies a command to run immediately before running the job defined by the What (on page 22) parameter. The return code is ignored.

NOTE: This parameter is not relevant in z/OS environments.

Additional information	Description
Usage	Optional
Length	1-4000 characters
Variable Name	%%PRECMD

Post-execution

Specifies a command to run immediately after running the job defined by the What (on page 22) parameter. The return code is ignored.

NOTE: This parameter is not relevant in z/OS environments.

Additional information	Description
Usage	Optional
Length	1-4000 characters
Variable Name	%%POSTCMD

Run job on all hosts in group

Specifies that job submission details be broadcast to all agents within a defined Host Group. All available agents in the Host Group run an identical job, and each such job has a unique Order ID.

NOTE: This parameter is not relevant in z/OS environments.

Additional information	Description
Usage	Optional
Default	No
Format	Check box
Alternate names	▪ **Control-M/EM Utilities:** MULTIAGENT ▪ **Reporting Facility:** MULTIAGENT ▪ **Control-M/Server Utilities:** -multiagent ▪ **Control-M/EM API:** multiagent
Previously Known AS	Multi Agent

The job is processed by all the agents specified for the parameter Host (/Group) (on page 30) (in the Active jobs database, only). A host group must be specified for the **Host /Host Group** parameter. For more details, see Host (/Group) (on page 30). For example, you can run a job that detects the amount of disk space available on the computer on which the job was run. By selecting **Run job on all hosts in group**, the job checks the available disk space on every agent computer in the specified Host Group.

Variables

All variables are identified by the **%%** prefix. If **%%** is included in the value for a job processing parameter, Control-M assumes that it is referring to a variable or function.

Additional information	Description
Usage	Optional
Format	▪ Name ▪ Value
Length	▪ Name: 1-40 ▪ Value: 1-4000 z/OS: ▪ Name: 1-66 ▪ Value: 1-66
Invalid Characters	None
Alternate Names	▪ **Control-M/EM Utilities:** VARIABLE ▪ **Reporting Facility**: VARIABLE ▪ **Control-M/Server Utilities:-**variable ▪ **Control-M/EM API:** variable

A special **%%#** prefix can be used to indicate that a variable or function should not be resolved. In these cases, the actual name of the variable or function (minus the **#** sign) is output. For example, **Do Notification Variable %%#PARM1 is greater than 100 i**ssues the following message:

Variable %%PARM1 is greater than 100.

Variables are divided into the following types:

- **Job Submission variables** pass parameters to a job or set the job's working parameters. For more information, see Job submission variables (on page 232).
- **System variables** are automatically assigned values using system information available at the time of job submission (for example, **%%DATE** contains the current system date). For more information, see Control-M system variables (on page 238).
- **User-defined variables** can be defined in a number of different ways for inclusion in various job processing parameters. For more information, see User-defined variables (on page 242).
- **Variable lists**. The %%LIBMEMSYM special variable can be used to point to a file containing a list of assignments statements to be applied to a job. This variable enables you to create one or more lists of assignment statements that can be applied to many job processing definitions. For more information, see Variable lists (on page 246).

Application

Provides a logical name for sorting groups of jobs. This parameter is used to supply a common descriptive name to a set of related job groups. The jobs do not necessarily have to run at the same time.

Additional information	Description
Usage	Mandatory
z/OS	optional
Default	None
Length	None
Case Sensitive	Yes
Invalid Characters	Single quotation marks
Variable Name	A variable or expression cannot be specified as all or part of the value for this parameter. However, the value of Application can be specified using the %%APPL and %%APPLIC variables.
Alternate Names	▪ **Control-M/EM Utilities** : APPLICATION ▪ **Reporting Facility**: APPLICATION ▪ **Control-M/Server Utilities**: -application ▪ **Control-M for z/OS**:APPL ▪ **Control-M/EM API**: application

The Application parameter facilitates more convenient and orderly management of sub-applications of production jobs. The value assigned to the Application parameter can be used to determine the placement of jobs in the Control- M flow diagram. It can also be used as a criterion for building a ViewPoint.

EXAMPLE: To identify all jobs created by the accounting department:

Application: ACCT

Sub Application

Indicates the name of the Sub Application where the job belongs logically. It is a sub-category of the Application parameter. For example, the Application is Finances, and the Sub Application is Payroll.

Additional information	Description
Usage	Mandatory
Length	1-64 characters
Case Sensitive	Yes
Invalid Characters	Single quotation marks
Variable Name	None
Alternate Names	▪ **Control-M/EM Utilities:** SUB_APPLICATION ▪ **Reporting Facility:** SUB-APPLICATION NAME ▪ **Control-M/Server Utilities:** -subapp ▪ **Control-M for z/OS:** SUB APPLICATION ▪ **Control-M/EM API:** Sub Application
Previously Known As	Group

The value assigned to the Sub Application parameter determines the job's logical placement in the structure of the Control-M network ViewPoint displayed in the Control-M flow diagram. This window is described in Monitoring, in *Control-M Workload Automation*.

EXAMPLE: Accounting department Sub Application

ACCOUNTING

EXAMPLE: Sub Application of jobs run at the end of the day

END_OF_DAY

EXAMPLE: Sub Application name for associated jobs that create sales reports

SALES_REPORT

Created by

Indicates the Control-M/EM user who defined the job.

NOTE: This parameter is not relevant in z/OS environments.

Additional information	Description
Usage	Mandatory
Default	Name of the Control-M/EM user that created the job
Length	1-64 characters
Case Sensitive	Yes
Invalid Characters	Blanks; single quotation marks
Variable Name	None
Alternate Names	▪ **Control-M/EM Utilities:** CREATED_BY ▪ **Reporting Facility:** CREATED BY ▪ **Control-M/Server Utilities:** -created_by
Previously Known As	Author

The Control-M/EM user specified by this parameter must possess a valid user ID registered in the Control-M installation. Authority to perform actions is verified by Control-M/Server security exits.

This parameter is used by the Control-M/Server for verifying if the owner of the job has authorization to submit the job processing definition during the submission of jobs by the New Day Procedure. If the AuthorSecurity system parameter is set to author security mode 2 or 3 (restricted), you cannot edit the **Created by** field unless you are a Control-M/EM administrator and are online (that is, connected to a GUI server). For more information, see Control-M security and the description of the AuthorSecurity system parameter in *Control-M Workload Automation Administration*.

NOTE: Depending on the value of the AuthorSecurity system variable, this parameter may be disabled.

Doc Library/ Doc Path

For a z/OS job, Doc Library defines the name of the library where the Documentation (description) is saved. For a non-z/OS job, Doc Path defines the name of the file path where the Documentation is saved. (This is specified in the Doc Member/ Doc File (on page 42) parameter).

Additional information	Description
Usage	Optional
Length	▪ Microsoft Windows and UNIX: 1-255 characters ▪ **iSeries (AS/400)**: 1-21 characters ▪ OpenVMS: 1-60 characters ▪ z/OS: 1-44 characters
Case Sensitive	Yes
Invalid Characters	z/OS: Non-English characters
Variable Name	None
Alternate Names	▪ **Control-M/EM Utilities:** DOCLIB ▪ **Reporting Facility**: DOC LIB ▪ **Control-M/Server Utilities:**-doclib ▪ **Control-M for z/OS**: DOCLIB ▪ **Control-M/EM API:** doc_lib
Computer specific information	
UNIX	Specify the name of the directory and subdirectories (if any). One of the following symbols can be used in place of or as part of the directory or subdirectory name (these symbols are resolved at the time the job is ordered or forced): ▪ **$HOME** resolves to the home directory of the job owner. ▪ ~<username> resolves to the home directory of the specified UNIX user.

Additional information	Description
iSeries (AS/400)	You must specify one of the following: ▪ Name of an **iSeries (AS/400)** library ▪ Library / File ▪ ***LIBL** (library list) ▪ ***CURLIB** (current library)
OpenVMS	The directory can be specified as the physical path or as an OpenVMS logical name.

The library or directory specified by this parameter, and the file specified in the Doc Member/ Doc File (on page 42) parameter, do not have to exist when the job processing parameters are defined.

If you specify this parameter, you must also specify a value for the Doc Member/ Doc File (on page 42) parameter.

NOTE: To access the documentation the user must be defined and have authorization on the Control-M/Server where the documentation resides

EXAMPLE: OpenVMS

DUA0:[ACCOUNT.DOC], ACCDOC$DIR

UNIX

$HOME/ctm/account/doc

Doc Member/ Doc File

For a z/OS job, defines the name of the member where the job Documentation (description) is saved. For a non-z/OS job, the Doc File is the name of the file where the job Documentation is saved.

Additional information	Description
Usage	Optional
Length	▪ 1-64 characters ▪ z/OS: 1-8 characters
Case Sensitive	Yes
Invalid Characters	▪ Blanks ▪ z/OS: Non-English characters
Alternate Names	▪ **Control-M/EM Utilities:** DOCMEM ▪ **Reporting Facility**: DOCMEM ▪ **Control-M/Server Utilities:** -docmem ▪ **Control-M for z/OS**: DOCMEM ▪ **Control-M/EM API:** doc_member
Computer specific information	
iSeries (AS/400)	**Doc Member** indicates the name of the member in the Doc Library/ Doc Path (on page 40) file. On this computer, the **Doc Member** parameter is optional even if a value is supplied for **Doc Libray** . If a library and file name are specified in **Doc Libray** but no value is specified for **Doc Member**, the ***FILE** default member name is used (for example, the member name is the same as the file name).

The **Doc Member** parameter is normally specified together with the Doc Library/ Doc Path (on page 40) parameter.

NOTE: To access the documentation the user must be defined and have authorization on the Control-M/Server where the documentation resides.

Type

Defines whether the documentation for an OS job is in a file or URL (on page 43).

Additional information	Description
Usage	This parameter consists of two options: ▪ File ▪ URL
Format	Drop-down list
Variable Name	None

URL

Defines the URL address where the documentation is located.

Additional information	Description
Usage	This parameter can be used only if the documentation type is **URL**.
Length	Combines the length of Doc Member/ Doc File (on page 42) 64 characters and Doc Library/ Doc Path (on page 40) 255 characters

Priority

Determines the order of job processing by Control-M in the Active Jobs database.

Additional information	Description
Usage	Optional
Format	2 alphanumeric characters
Default	Blank, which is the lowest priority
Case Sensitive	No
Invalid Characters	Single quotation marks; non-English characters
Alternate Names	▪ **Control-M/EM Utilities:** PRIORITY ▪ **Reporting Facility**: PRIORITY ▪ **Control-M/Server Utilities: -**priority ▪ **Control-M for z/OS**: PRIORITY ▪ **Control-M/EM API:** priority
Computer specific information	
z/OS	A job may be so important that lower priority jobs must not be submitted until the important job has executed. Such a job is called a critical path job. If the first character of Priority in z/OS jobs is set to * (Asterisk), the job is marked as a critical path job. There is no relationship between the Critical parameter and the Priority parameter.

Active Jobs database prioritizing processing:

- Priority order is such that 9>0>Z>A. The characters are not case sensitive.
- The value for the priority parameter is a 2 character string. AA is the lowest priority. 99 is the highest. If a single character is specified, the uppercase letter A is automatically inserted as the first character. For example, priority 1 is treated as priority A1.
- The next line shows priority values from the lowest (on the left) to the highest (on the right);

  AA-A9...ZA-Z9, 0A-0Z, 01-09, 1A-19...9A-99

If a job that is waiting to be submitted because Quantitative resources are not available, has a higher priority than a job that has all the Quantitative resources available, the lower priority is submitted. However, by defining a job as critical, the user can force Control-M to reserve resources for the job, thus assuring that it s submitted as soon as possible. For more information, see Critical (on page 45).

Consult authorities at your installation for more information about your priority usage standards.

Critical

Determines whether the job is a critical-path job in Control-M, which ensures resources allocation order.

Additional information	Description
Usage	Optional
Format	Check box ▪ Selected – job is critical ▪ Cleared – job is not critical
Alternate names	▪ **Control-M/EM Utilities:** CRITICAL ▪ **Reporting Facility:** CRITICAL ▪ **Control-M/Server Utilities:** -critical ▪ **Control-M for z/OS:** (none) ▪ **Control-M/EM API:** critical
Alternate formats	
Control-M/EM Utilities	Valid values: ▪ **1** - Job is critical. ▪ **0** - Job is not critical. Default. **EXAMPLE:** <CRITICAL="1"/>
Control-M/Server Utilities	Valid values: ▪ **Y** - Job is critical. ▪ **N** - Job is not critical. Default.
Computer specific information	

Additional information	Description
For non-z/OS computers	Defining a job as critical ensures that a job that requires resources is submitted as soon as possible after all its In Conditions parameters are satisfied. As a result of a job being defined as critical, any Quantitative resources or Control resources that the job requires exclusively are reserved for the job as they become available. The preferential treatment given to a critical job is applied only after all the job's In Conditions parameters are satisfied. The Critical parameter takes precedence over the Priority parameter (a low-priority job defined as critical is given preferential treatment over a non-critical high-priority job). However, if two critical jobs are awaiting execution at the same time, the higher-priority job receives resources before the lower-priority job. Critical path priority applies to contention between Quantitative resources and between Control resources with Exclusive status. The critical path priority does not apply to contention with Control resources with Shared status. For non-z/OS computers, critical is available for jobs whose statuses are not Executing when editing details in the active Jobs database. For more information about the Priority parameter, see Priority (on page 44).
z/OS	The Critical parameter check box is replaced with a check box labeled **Emergency**. When selected, this marks the z/OS job as an Emergency job. [For z/OS jobs] The **Emergency** check box is marked if the job definition is created as a result of a Control-M/EM utility and the Critical parameter was set to '1' (see below).

EXAMPLE: Resource availability overrides critical status

If one tape drive is available, neither job is submitted. When two tape drives become available, job EBDUPDT is submitted.

Job EBDUPDT's priority level is 50, Critical is selected, and it requires two tape drives.

Item	Value
File Name	EBDUPDT
Quantitative Resources	TAPE 0002
Priority	50

Item	Value
Critical	Y

Job EBDEXEC's priority level is 04, and it requires one tape drive. It is not a Critical job.

Item	Value
File Name	EBDEXEC
Quantitative Resources	TAPE 0001
Priority	04
Critical	N

EXAMPLE: Resource availability determines job submission

If one or two tape drives are available, neither job is submitted. When three tape drives become available, job EBDBKP is submitted.

Job EBDBKP's priority level is 8A, Critical is selected, and it requires three tape drives.

Item	Value
File Name	EBDBKP
Quantitative Resources	TAPE 0003
Priority	8A
Critical	Y

Job EBDMAINT's priority level is 70, Critical is selected, and it requires one tape drive.

Item	Contents
File Name	EBDMAINT
Quantitative Resources	TAPE 0001
Priority	70
Critical	Y

Control-D Category

Defines the name of the Control-D Report Decollating Mission Category. If specified, the report decollating mission is scheduled whenever the job is scheduled under Control-M.

Additional information	Description
Usage	Optional
Length	1-20
Invalid Characters	Blanks
Alternate Names	▪ **Control-M/EM Utilities:** CATEGORY ▪ **Control-M for z/OS:** D-CAT ▪ **Reporting Facility**: CATEGORY ▪ **Control-M/Server Utilities:** -category ▪ **Control-M/EM API:** category

Run as Detached

A regular job submitted to Control-M for execution as a background process. The results of the job (the output) are analyzed by the post-processing subsystem.

Additional information	Description
Usage	Optional
Format	Check box
Variable Name	None
Alternate Names	▪ **Control-M/EM Utilities:** TASKTYPE ▪ **Reporting Facility**: TASKTYPE ▪ **Control-M/Server Utilities: -**tasktype ▪ **Control-M/EM API:** tasktype
Previously Known As	Tasktype Detached

Run as started task

Specifies that the job is invoked with the operator START command.

Additional information	Description
Usage	Optional
Format	Check box
Variable Name	None
Alternate Names	▪ **Control-M/EM Utilities:** TASKTYPE ▪ **Reporting Facility**: TASKTYPE ▪ **Control-M/Server Utilities: -**tasktype ▪ **Control-M/EM API:** tasktype
Previously Known As	Tasktype Started

Prevent NCT2

Performs data set cleanup before the original job run.

Additional information	Description
Usage	Optional

Additional information	Description	
Format	Select one of the following from the **Prevent NCT2** list box: ▪ **Blank –** Do not perform data set cleanup before the original job run. **Default.** ▪ **N** (No) – Do not perform data set cleanup before the original job run. ▪ **Y** (Yes) – Perform data set cleanup before the original job run. This value is not valid for started tasks. ▪ **L** (List) – Do not perform data set cleanup before the original job run; but generate the messages that would be required for GDG adjustment during restart. ▪ **F** (Flush) – Halt processing of the job if any data set cleanup error is detected (even if z/OS would not have stopped processing the job).	
Variable Name	None	
Alternate names	▪ **Control-M/EM Utilities:** PREVENTNCT2 ▪ **Reporting Facility:** PREVENT NCT2 ▪ **Control-M/Server Utilities:** (none) ▪ **Control-M for z/OS:** PREVENT-NCT2 ▪ **Control-M/EM API:** prevent_nct2	
Alternate Formats		
Control-M/EM Utilities	Valid values: ▪ **0** - No. Does not prevent data set cleanup. ▪ **1** - Yes. Prevents data set cleanup. **EXAMPLE:** <PREVENTNCT2="1"/>	
Control-M for z/OS	PREVENT-NCT2 is composed of two parameters:	
	PREVENT-NCT2	Valid values: ▪ **N** - No ▪ **Y** - Yes ▪ **L** - List ▪ **F** - Flush

Additional information	Description	
	DFLT	Protected field indicating the PREVENT-NCT2 default value for the Control-M for z/OS site. The default is set in parameter NCAT2 in the CTRPARM member in the IOA PARM library. A value specified in the PREVENT-NCT2 parameter overrides the site default.

If a job tries to create a data set that already exists, the job may fail with a DUPLICATE DATA SET ON VOLUME error. If a job tries to create a data set whose name is already cataloged, the job may fail with an error message that indicates a reason of NOT CATLGD for reason code 2 (the Control-M/Restart term PREVENT-NCT2 is derived from this error situation). These problems can be avoided by performing data set cleanup. During data set cleanup, Control-M/Restart does the following:

- Deletes and uncatalogs the old data sets. This prevents DUPLICATE DATSET ON VOLUME and NOT CATLGD 2 errors.
- Performs Generation Dataset (GDG) Adjustment, which is described in the *Control-M/Restart User Manual*.

Control-M/Restart automatically performs data set cleanup prior to restarts and reruns. However, it may be desirable to perform data set cleanup before the original job run, because data sets accessed by the job can have file-related errors that were generated by an entirely different job. When data set cleanup is performed as part of the original job request, it is called PREVENT-NCT2 processing. The site-defined default in parameter NCT2 in member CTRPARM determines whether data set cleanup is to be performed before the original job run. The Prevent NCT2 parameter can be used to override this default to determine what data set cleanup instructions are provided to the original job run. Possible values, and their effects, are described below:

- When **N** is specified, No special action is taken by Control-M/Restart. Data set cleanup is not performed.
- When **Y** is specified, Control-M/Restart performs data set cleanup before the original job run. It deletes and uncatalogs all data sets that can cause NCT2 and duplicate data set errors during execution, and performs GDG adjustment if necessary.
- When **L** is specified, data set cleanup is not performed for the original run, but messages that would be required for GDG adjustment during restart are generated. Without these messages, GDG adjustment might not be properly performed during restart. In addition to the GDG adjustment messages, the same messages that are generated during simulation of data set cleanup are also generated.
- When **F** is specified, If a file catalog error is detected, processing is halted (even if normal z/OS processing would not handle the problems as a fatal error) and an appropriate error message is generated.

NOTE: If you would normally specify **N** (that is, Control-M/Restart processing is not desired for the original run), but the JCL requires GDG processing, BMC Software recommends that you specify value **L** instead of value **N**.

If a value of **Y**, **L**, or **F** is specified (that is, if some kind of special NCT2 processing is desired), a ControlR step is automatically added as a first step of the submitted job. The PREVENT NCT2 parameter has no impact on restarts, because Control-M/Restart automatically performs data set cleanup prior to restarts.

Request NJE Node

Defines the node in the JES network where the job executes.

NOTE: This field is relevant only for z/OS jobs

Additional information	Description
Usage	Optional
Length	1-8 characters
Case Sensitive	No
Invalid Characters	Blanks; non-English characters
Variable Name	None
Alternate names	▪ **Control-M/EM Utilities:** (none) ▪ **Reporting Facility:** (none) ▪ **Control-M/Server Utilities:** (none) ▪ **Control-M for z/OS:** NJE NODE ▪ **Control-M/EM API:** request_nje

The **Request NJE Node** parameter is used to specify the node in the JES network on which the job is to execute.

If a value is specified for the Request NJE Node parameter, a JCL statement is generated. The precise form of the statement depends on whether Control-M is running under JES2 or JES3.

NOTE: If a value is specified for the Request NJE Node parameter, it does not override any node name specified in the job statement unless the OVERJCLM parameter in the CTMPARM library is set to **Y**

Under JES2

If Control-M is running under JES2, the Request NJE parameter generates the following JCL statement:

```
/*ROUTE XEQ node_name
```

Under JES3

If Control-M is running under JES3, the JCL statement generated by the Request NJE parameter differs slightly, taking the following form:

```
//*ROUTE XEQ node_name
```

EXAMPLE: Under JES2

The following values are entered to the job processing definition:

- `DESC`
- `OVERRIDE PATH`
- `SCHENV SYSTEM ID NJE NODE OS35`

The following statement is added to the JCL of the job:

```
/*ROUTE XEQ OS35
```

The job is executed at node **OS35**.

EXAMPLE: Under JES3

The following values are entered to the job processing definition:

- `DESC`
- `OVERRIDE PATH`
- `SCHENV SYSTEM ID NJE NODE OS35`

EXAMPLE: The following statement is added to the JCL of the job:

```
//*ROUTE XEQ OS35
```

The job is executed at node OS35.

Scheduling Environment

Indicates the JES2 workload management scheduling environment that is to be associated with the job.

NOTE: This parameter is relevant only for z/OS jobs.

Additional information	Description
Usage	Optional
Length	**1** through **16** characters
Case Sensitive	Yes
Invalid Characters	Blanks; non-English characters
Variable Name	None
Alternate names	▪ **Control-M/EM Utilities:** SCHEDULING_ENVIRONMENT ▪ **Reporting Facility:** SCHEDULE ENV ▪ **Control-M/Server Utilities:** (none) ▪ **Control-M for z/OS:** SCHENV ▪ **Control-M/EM API:** schedule_environment

The **Scheduling Environment** parameter can only be used when Control-M is running under JES2. If Control-M is running under JES3, any value specified for the **Scheduling Environment** parameter is ignored.

If a value is specified for the Scheduling Environment parameter, the JCL job statement is modified by the addition of a statement in the following form:

```
// SCHENV=schedule_environment
```

NOTE: If a value is specified for the **Scheduling Environment** parameter, it does not override any scheduling environment specified in the job statement unless the OVERJCLM parameter in the CTMPARM library is set to **Y**.

EXAMPLE: Specifying a scheduling environment

If the scheduling environment of job ACCT01 is to be SCHD2, specify the following:

- `DESC`
- `OVERRIDE PATH`
- `SCHENV SCHD2 SYSTEM ID NJE NODE`

The job statement is modified as follows:

- `//ACCT01 JOB ,PROD1,CLASS=A,MSGCLASS=X`
- `// MSGLEVEL=(1,1),`
- `// SCHENV=SCHD2`

Order Method

Defines the method for ordering the entity as one of the following:

- **Automatic (Daily):** When set to Automatic, at the same time each day (known as New Day time), each Control-M/Server runs a procedure called New Day. This procedure performs a number of tasks, including scheduling the day's jobs, and running maintenance and cleanup utilities. The New Day procedures orders the folder or folder jobs.
- **None (Manual Order):** The folder is not automatically ordered.
- **Specific User Daily:** Identifier used to assign the folder to a specific User Daily job. The User Daily name (on page 61) is ordered at a specific time of the day. For load balancing purposes, the User Daily jobs are scheduled for different times, throughout the day, other than the New Day time.

Additional information	Description
Usage	Mandatory
Format	Drop-down list
Variable Name	None

Additional information	Description
Alternate Names	▪ **Control-M/EM Utilities:** ORDER_METHOD ▪ **Reporting Facility**: ORDER METHOD ▪ **Control-M/Server Utilities: -**ordermethod ▪ **Control-M/EM API:** order method
Previously Known As	User daily

Batch Impact Manager Service Actions

Defines the automatic interventions, such as rerunning a job, displaying the critical service in BMC Service Impact Manager, or extending the service due time. The action is performed if the job finished too quickly, there is a job failure on service path, or the service finished late or too quickly.

Events are defined using the **When** parameter. Corresponding actions are defined using the **Do** parameter. Multiple **Do** parameters can be specified for the same event in the same rule.

Parameter	Description
Job failure on service path	One or more of the jobs in the critical service failed and, as a result, caused a delay in the service. **NOTE:** A service is considered OK even if one of its jobs fails, provided that another job, with an Or relationship to the failed job, runs successfully.
Service is late	The entire service is late according to BMC Batch Impact Manager calculations.
Job ran too long	One of the jobs in the critical service is late. Lateness is calculated according to the average run time and the value of the Job Runtime Tolerance parameters. **NOTE:** A service is considered on time even if one of its jobs is late, provided that the service itself is not late.
Job finished too quickly	One of the jobs in the critical service is early. This is calculated according to the average run time and the value of the Job Runtime Tolerance parameters. **NOTE:** A service is considered on time even if one of its jobs is early.

Parameter	Description
Set Variable (on page 150)	The Set Variable assigns a value to a variable for use in a rerun of the job when the On Statement/Code criteria are satisfied.
Condition (on page 152)	The Condition parameter specifies a prerequisite condition to be added or deleted to/from the Conditions table when the On Statement/Code criteria are satisfied.
Order Job (on page 154)	The Order Job parameter forces an individual job or all jobs in a folder to be placed in the Active Jobs database (regardless of each job's Scheduling criteria) when the On Statement/Code criteria are satisfied. For BIM jobs there are the following extra sub-parameters: ▪ Control-M/Server (on page 31) ▪ Folder Library (on page 63) ▪ Folder Name (on page 65) ▪ Job Name (on page 19) ▪ Date (on page 130)
Mail (on page 157)	Sends an e-mail when the specified event occurs.
Notify (on page 156)	The Notify parameter specifies a notification to be sent a specific destination when the specified On Statement/Code criteria are satisfied.
OK (End Job Ok (on page 161))	The Ok parameter assigns the completion status of OK to a job, regardless of its actual completion status. For BIM jobs there are the following extra sub-parameters: ▪ Control-M/Server (on page 31) ▪ Folder Name (on page 65) ▪ Job Name (on page 19) ▪ Date (on page 130) ▪ Problematic Job (on page 60)
Remedy (on page 162)	Opens a ticket in the Remedy Help Desk.

Parameter	Description
Rerun Job (on page 163)	Indicates if an automatic rerun should be performed when the On Statement/Code criteria are satisfied. For BIM jobs there are the following extra sub-parameters: ▪ Control-M/Server (on page 31) ▪ Folder Name (on page 65) ▪ Job Name (on page 19) ▪ Date (on page 130) ▪ Problematic Job (on page 60)
Kill Job (on page 58)	Kills a problematic job while it is still executing.
SIM (on page 59)	Sends early warning notification to, and displays the critical service in BMC Service Impact Manager.
Extend Service due time (on page 60)	Allow the job or critical service to continue running by extending the deadline by which the job or service can run and still be considered on time. Extend the period by hours and/or minutes.

Kill Job

Kills a problematic job while it is still executing.

Additional information	Description
Usage	Optional
Format	Consists of the following sub-parameters: ▪ Control-M/Server (on page 31) ▪ Folder Name (on page 65) ▪ Job Name (on page 19) ▪ Date (on page 130) ▪ Problematic Job (on page 60)

NOTE:

- If both of the following conditions are met, the Kill-job action is performed on all jobs that match the criteria: Control-M/Server, Folder Name, Job Name, and Order Date sub-parameters do not resolve to unique jobs.

- Your site supports the ordering of multiple jobs with the same name, in the same folder, in the same Control-M/Server, on the same order date.

SIM

Sends early warning notification to, and displays the critical service in BMC Service Impact Manager.

Additional information	Description
Usage	Optional
Format	Consists of the following sub-parameters: ▪ **Connect to** ▪ **Message**
Sub-parameters	
Connect to	Defines the hostname[:port] of the ProactiveNet Server/Cell to send the notification. If no port is specified, the default port 1828 is used. **EXAMPLE:** myhost.bmc.com or myhost.bmc.com:1829
Message	The message to be displayed in BMC Service Impact Manager. Mandatory. Maximum length: 211 You can use variables in the message.

Extend Service due time

Allow the job or critical service to continue running by extending the deadline by which the job or service can run and still be considered on time.

Additional information	Description
Usage	Optional
Format	Consists of the **Extend must completed by** sub-parameter.
Sub-parameters	
Extend 'Must completed by'	Amount of time to add to the service in HH:MM format. Mandatory.

Problematic Job

A job in a service that is not running on time, and, as a result, will impact the service.

Additional information	Description
Usage	Optional
Format	Checkbox
Variable	%%PROBLEMATIC_JOBS

User Daily name

Defines User Daily jobs whose sole purpose is to order jobs. Instead of directly scheduling production jobs, the New Day procedure can schedule User Daily jobs, and those User Daily jobs can schedule the production jobs. Set User Daily Name when Order Method is set to Specific User Daily.

Additional information	Description
Usage	This parameter can be used only if the Order method is **specific user daily**.
Length	1-10
Alternate Names	▪ **Control-M/EM Utilities:** USERDAILY ▪ **Reporting Facility**: USER DAILY ▪ **Control-M/Server Utilities: -**USERDAILY ▪ **Control-M/EM API:** user daily

Emergency Job

Determines whether the z/OS job is an Emergency job.

Additional information	Description
Usage	Optional
Format	Check box
Variable Name	None

System Affinity

Indicates the identity of the system in which the job must be initiated and executed (in JES2).

NOTE: This field is relevant only for z/OS jobs.

Additional information	Description
Usage	Optional
Length	1-5 alpha-numeric characters The alpha-numeric characters can be proceeded by a "/". "/" as a first character indicates NOT in JES3.
Case Sensitive	No
Invalid Characters	Non-English characters
Variable Name	None
Alternate names	▪ **Control-M/EM Utilities:** SYSTEM_AFFINITY ▪ **Reporting Facility:** SYS AFFINITY ▪ **Control-M/Server Utilities:** (none) ▪ **Control-M for z/OS:** SYSTEM ID ▪ **Control-M/EM API:** system_affinity

Indicates the identity of the processor on which the job must execute (in JES3).

The **System Affinity** parameter has different effects, depending on which release of JES is in use.

NOTE: If a value is specified for the **System Affinity** parameter, it does not override any system identity specified in the job statement unless the OVERJCLM parameter in the CTMPARM library is set to **Y**.

Under JES2

If Control-M is running under JES2, the **System Affinity** parameter is used to specify the JES2 system on which the job is to be initiated and executed. If a value is specified for the **System Affinity** parameter, the following JCL statement is generated:

```
/*JOBPARM SYSAFF=sys_id
```

Under JES3

If Control-M is running under JES3, the **System Affinity** parameter is used to specify the JES3 processor that is to execute the job. If a value is specified for the **System Affinity** parameter, the following JCL statement is generated:

```
//*MAIN SYSTEM=processor_id
```

EXAMPLE: Under JES2

The following values are entered to the job processing definition:

- `DESC`
- `OVERRIDE PATH`

- SCHENV SYSTEM ID SYS3 NJE NODE

The following statement is added to the JCL of the job:

/*JOBPARM SYSAFF=SYS3

The job is executed on the JES2 system SYS3.

Under JES3

The following values are entered to the job processing definition:

- DESC
- OVERRIDE PATH
- SCHENV SYSTEM ID PRC3 NJE NODE

The following statement is added to the JCL of the job:

//*MAIN SYSTEM=PRC3

The job is executed on processor PRC3.

Folder Type

Indicates whether the folder is simple or SMART.

Additional information	Description
Usage	Optional
Format	Check box
Variable Name	None
Previously Known As	Table Type

Folder Library

Defines the name of the library that contains the job's folder.

NOTE: This field is displayed only for z/OS jobs.

Additional information	Description
Usage	Mandatory
Length	1-44 characters
Case Sensitive	No
Invalid Characters	▪ Blanks ▪ z/OS: Non-English characters
Alternate Names	▪ **Control-M/EM Utilities:** FOLDER_DSN ▪ **Reporting Facility:** (none) ▪ **Control-M/Server Utilities:** (none) ▪ **Control-M for z/OS:** Folder Library (Folder Lib) ▪ **Control-M/EM API:** folder_library
Previously Known As	Table Library

Folder Name

Defines the name of the folder. In the Properties pane this parameter indicates the folder where the job belongs.

Additional information	Description
Usage	Mandatory, if values are specified for the **Job Name** and **Date** parameters.
Length	▪ 1-64 characters ▪ z/OS: 1-8 characters
Case Sensitive	Yes
Invalid Characters	▪ Blanks ▪ Single quotation marks ▪ z/OS: Non-English characters ▪ "$", "/", "*", "?", " " .
Variable Name	None
Alternate Names	▪ **Control-M/EM Utilities:** FOLDER_NAME ▪ **Reporting Facility:** FOLDER_ID ▪ **Control-M/Server Utilities:** FOLDER ▪ **Control-M for z/OS:** FOLDER NAME ▪ **Control-M/EM API:** folder_name
Previously Known As	Table Name

In the Properties pane, together with the Control-M Job Name (on page 19) parameter, the **Folder Name** parameter determines the position of the job in the Control-M Folder hierarchy. The **Folder Name** parameter may include folder name or folder path.

EXAMPLE: Folder name including a user-assigned serial number.

SchTbl03

EXAMPLE: Folder name including a time period.

SeptOctTbl2

From program step

Defines the first program step for a job to begin at when it is restarted.

NOTE: This parameter is relevant only for z/OS jobs.

Additional information	Description
Usage	Optional
Length	1-8
Case Sensitive	No
Invalid Characters	Blanks
Variable Name	None

To program step

Defines the last program step for a job to stop running.

NOTE: This parameter is relevant only for z/OS jobs.

Additional information	Description
Usage	Optional
Length	1-8
Case Sensitive	No
Invalid Characters	Blanks
Variable Name	None

From procedure step

Defines the first procedure step for a job to begin at when it is restarted.

NOTE: This parameter is relevant only for z/OS jobs.

Additional information	Description
Usage	Optional
Length	1-8
Case Sensitive	No
Invalid Characters	Blanks
Variable Name	None

To procedure step

Defines the last procedure step for a job to stop running.

NOTE: This parameter is relevant only for z/OS jobs.

Additional information	Description
Usage	Optional
Length	1-8
Case Sensitive	No
Invalid Characters	Blanks
Variable Name	None

Service Name

Logical name, from a user or business perspective, for the critical service. BMC recommends that the service name be unique.

NOTE: This parameter is relevant for BIM jobs only.

Additional information	Description
Usage	Mandatory
Length	1-64 characters
Case Sensitive	Yes
Invalid Characters	▪ Blanks ▪ Single quotation marks ▪ If the job runs on any version of Microsoft Windows 2000, prohibited filename characters (such as \, /, or *)
Variable Name	%%SERVICE_NAME

Service must complete

Defines the time the critical service must be completed to be considered on time. The time can be specified either by a specific hour and/or day or on a specific hour and minute after the order time.

NOTE: This parameter is relevant for BIM jobs only.

Additional information	Description
Usage	Mandatory
Format	This parameter consists of the following sub-parameters: ▪ By ▪ Number of days ▪ In number of hours ▪ Number of min after
Variable Name	%%SERVICE_DUE_TIME
Sub-parameters	
By	Specific hour, in the format **HH:MM**, according to a 24-hour clock. Valid values range include 00:01 through 99:59

Additional information	Description
Number of days	Determine on which day the critical service must complete, relative to ODATE. The value is specified in terms of the number of days after the order date of the service. 0 represents the order date, 1 represents one day after the order date, and so on. Default: 0
In number of hours	A specified number of hours.
Number of min after	A specified number of minutes.

Service Priority

Defines the priority level of this service, from a user or business perspective.

NOTE: This parameter is relevant for BIM jobs only.

Additional information	Description
Usage	Optional
Format	Drop-down list The priority can be set from 1 (highest priority) to 5 (lowest priority),and enables filtering according to priority in the Service Monitor window and the Web Client.
Default	3
Variable Name	%%SERVICE_PRIORITY

Job Run-Time Tolerance

Defines the type of deviation from the average completion time for a job in the service. If the run time falls within the tolerance set, it is considered on time, otherwise it has run too long or ended too early. The values are Percentile range or Average Run Time (in percentage or by minutes).

NOTE: This parameter is relevant for BIM jobs only.

Additional information	Description
Usage	Optional
Format	The tolerance can be defined: ▪ as a percentile range (Default) ▪ as average run time: in percentage in minutes Job Run-Time Tolerance relates only to the average completion time for the job –not the service. The tolerance value affects the Job ran too long and Job ended too quickly job-related status, but not the service is late status. Tip: For BMC Batch Impact Manager to perform its calculations correctly, BMC Batch Impact Manager and Control-M clocks should be synchronized. If the clocks cannot be synchronized, you can use the Job Run-time Tolerance parameter to compensate for this discrepancy.
Default	As percentile range, and 3 from the drop-down list.
Variable Name	None
Sub-parameters	
Percentile range (on page 70)	Defines the number of standard deviations of job run in this service. Percentile Range is the default method for specifying Job Run-Time Tolerance.
Average run time +/- (on page 71)	The percentage of the time (based on the average run time for the job) or the number of minutes that the job can be early or late, and still be considered on time.

Percentile range

Defines the number of standard deviations of job run in this service. Percentile Range is the default method for specifying Job Run-Time Tolerance.

NOTE: This parameter is relevant for BIM jobs only.

Additional information	Description
Usage	Optional
Format	Drop-down list Select one of the following values from the list, where 2 indicates more confidence in the completion time for the jobs and 4 less confidence: ▪ 2 –- 95.44% ▪ 3 –- 99.73% ▪ 4 –- 99.99%
Default	3
Variable Name	None

Average run time +/-

The percentage of the time (based on the average run time for the job) or the number of minutes that the job can be early or late, and still be considered on time.

NOTE: This parameter is relevant for BIM jobs only.

Additional information	Description
Usage	Optional
Format	▪ Numbers up to 999 ▪ Drop-down list: Percentage or Minutes
Length	▪ Percentage: 999 ▪ Minutes: 999
Default	▪ Percentage (Default): 100 ▪ Minutes: 100
Variable Name	None

Scheduling Parameters

Scheduling parameters determine when and/or how often a job is scheduled for submission

Parameter	Description
Active/Not Active between dates (on page 76)	Determines a date range (Start Date (on page 99) - End Date (on page 99)) when the job or folder can be ordered or the Rule-based Calendar (RBC) can be used. During the period that the job or folder is outside the active range (that is, inactive), it is not eligible to be ordered.
Keep Active for (on page 79)	Determines the number of extra days (beyond the original scheduling date) that the job is allowed to remain in the Active Jobs database while awaiting execution. If the job still has not run after the specified number of days, the job is removed from the Active Jobs database.
Keep all jobs in folder until folder is removed (on page 81)	Indicates that all jobs in the folder are not removed automatically from the Active Jobs database. Instead jobs wait for the folder to complete and are removed at the same time as the folder.
Keep the folder for a minimum number of days after ended not ok (on page 83)	Enables you to specify a minimum period to keep the SMART folder (and jobs) in the Active Jobs database after the folder is set to NOT OK.
Days And/Or (on page 84)	Indicates the relationship between specified Days values and Weekdays values.
Confirmation calendar (on page 85)	Indicates the name of a Control-M calendar that is used to validate scheduling dates. A shift value can be used to indicate how to handle jobs that are scheduled for a non-working day in the calendar.
Exception policy (on page 86)	Specifies what to do if a job is supposed to run on a day that is not in the confirmation calendar.
Cyclic (on page 88)	Indicates that the job must run at a designated time, interval of time.
Next expected run (on page 90)	Specifies the next expected submission date and time for the job, for rerun or cyclic jobs that use the Interval option.
All intervals are from job's (on page 91)	Indicates whether the interval between runs of a cyclic job or until the start of a rerun job is measured from the start or the end of the previous job run.

Parameter	Description
Rerun using the following interval sequence (on page 92)	Specifies a unit of time and a numerical value that defines the interval sequence for a job to rerun.
Using Calendar/Calendar (on page 96)	Specifies the name of the calendar that was defined in the Calendar Manager
Start Date (on page 99)	Sets the date when the job must be ordered.
End Date (on page 99)	Sets the date when the job must be ordered.

Parameter	Description
From Time/To Time (on page 114)	Sets time limits for submitting the job.
Ignore from time on next day (on page 118)	Enables the job to run the next day without time limitations.
Rerun Every (on page 93)	Specifies the length of time to wait between reruns of a job or between cyclic runs of a job.
Run at (on page 95)	Defines the specific times for the job to run.
Tolerance (on page 95)	Maximum delay in minutes permitted for a late submission when selecting a specific time (e.g. 5 minutes).
Maximum reruns (on page 100)	Determines the maximum number of reruns that can be performed for the job.
Minimum number of tracks (on page 101)	Minimum number of free partitioned data set tracks required by the library specified for the Partition Data Set parameter.
Order on months (on page 103)	Determines which months the job can be scheduled for processing.
Partition Data set (on page 104)	Defines the name of a partitioned data set to check for free space. If the Partition Data Set has fewer than the minimum number of required free tracks (as specified for the Minimum number of tracks parameter), the job is executed.
Relationship (on page 106)	Indicates the relationship (AND/OR) between RBCs criteria and basic scheduling criteria in the job processing definition (that is, whether either set of criteria, or both sets of criteria, must be satisfied).
Rerun Member (on page 107)	Defines the name of the JCL member to use when the job automatically reruns.
Retroactively order job that its scheduled date has passed (on page 108)	Indicates if the job should be scheduled for possible execution after its original scheduling date has passed.
SAC (on page 110)	Determines whether to adjust the logical date for a job converted from a scheduling product other than Control-M.

Parameter	Description
Rule-Based Calendar (on page 96)	Identifies a set of scheduling criteria defined for a folder. The scheduling criteria referenced by each Rule-based Calendar are defined in the outermost folder.
Schedule (on page 112)	Specifies the type of scheduling to use.
Statistics Calendar (on page 112)	Name of the Control-M periodic calendar within which statistics relating to the job are collected.
Statistics Period (on page 114)	Identifier of the actual days within the Control-M periodic calendar in relation to which statistics relating to the job are calculated.
Time Zone (on page 118)	Indicates the time zone according to which the job should be scheduled.
Must End (on page 117)	Sets the time and day when the job must finish executing. (z/OS only)

Active/Not Active between dates

Determines a date range (Start Date (on page 99) - End Date (on page 99)) when the job or folder can be ordered or the Rule-based Calendar (RBC) can be used. During the period that the job or folder is outside the active range (that is, inactive), it is not eligible to be ordered.

Additional information	Description
Usage	Optional
Format	Drop-down list. Define one or both of the following sub-parameters: ▪ Start Date (on page 99) ▪ End Date (on page 99)
Variable Name	None
Alternate names for the Active Start Date parameters	▪ **Control-M/EM Utilities:** ACTIVE_FROM_DATE ▪ **Control-M/Server Utilities:** -DATEFROM <YYYYMMDD> **EXAMPLE**: -DATEFROM 20060315 ▪ **Control-M for z/OS:** DEFINITION ACTIVE START or SCHEDULE RBC ACTIVE START ▪ **Control-M/EM API:** active_from
Alternate names for the Active End Date parameters	▪ **Control-M/EM Utilities:** ACTIVE_TO_DATE ▪ **Control-M/Server Utilities:** -DATEUNTIL <YYYYMMDD> **EXAMPLE**: -DATEUNTIL 20050315 ▪ **Control-M for z/OS:** DEFINITION ACTIVE END or SCHEDULE RBC ACTIVE END ▪ **Control-M/EM API:** active_till or active_until
Alternate formats for the Active From Date parameters	
Control-M/EM Utilities	String for date. According to site standard (*ddmmyyyy*, *mmddyyyy*, *yyyymmdd*). **EXAMPLE**: <ACTIVE_FROM="15032006"/>
Control-M for z/OS	6-integer string for date. According to site standard (*ddmmyy*, *mmddyy*, *yymmdd*).
Alternate formats for the Active To Date parameters	

Additional information	Description
Control-M/EM Utilities	String for date. According to site standard (*ddmmyyyy*, *mmddyyyy*, *yyyymmdd*). **EXAMPLE**: <ACTIVE_TILL="15032006"/>
Control-M for z/OS	6-integer string for date. According to site standard (*ddmmyy*, *mmddyy*, *yymmdd*). Default: blank.

You can use the Active parameters to define either of the following:

- An active period. The dates before this period and the dates after this period each constitute an inactive period:
  (*inactive-period1* > *defined-active-period* > *inactive-period2*)
- An inactive period. Technically you define two active periods, an earlier active period, and a later active period, and everything in-between is the inactive period:
  (*active-period1* > *defined-inactive-period* > *active-period2*)

You can use this parameter with multiple copies of a job or folder definition to create date ranges during which the job or folder definition is run with alternate values, to eliminate the necessity of making last-minute changes to a job processing definition for a fixed period of time. In all other Control-M components, ensure that the **Active Start Date** is less than or equal to the **Active End Date**.

No matter which component you use to define the Active period, you can leave either the Start date or the End date blank:

- If only a start date is specified, the job can be scheduled on or after that date.
- If only an end date is specified, the job can be scheduled on or before that date.

When you define an Inactive period:

- the end date identifies the last date of the earlier active period (*active-period1*)
- the start date identifies the first date of the later active period (*active-period2*)

Therefore, the end date must be less than the start date.

EXAMPLE: Seasonal changes

JOB_A is a job processing definition for scheduling food shipments.

- In the summer, the company ships on Mondays and Wednesdays.
- In the fall, the company ships only on Mondays.
- In winter, the company ships on Mondays, Wednesdays, and Fridays.
- In the spring, the company ships on Mondays and Wednesdays.

Four copies of the JOB_A job processing definition are created. The Days parameter must be changed seasonally to allow for variations in shipping frequency. Different Active Start Date and Active End Date parameters are used in each definition to specify the season in which the job is active. Other job processing criteria remain unchanged.

EXAMPLE: Job_A, Copy_1: Summer

- Days: 1, 3
- Active Start Date: June 1, 2005
- Active End Date: August 31, 2005

EXAMPLE: Job_A, Copy_2: Fall

- Days: 1
- Active Start Date: September 1, 2005
- Active End Date: November 30, 2005

EXAMPLE: Job_A, Copy_3: Winter

- Days: 1, 3, 6
- Active Start Date: December 1, 2005
- Active End Date: March 1, 2006

EXAMPLE: Job_A, Copy_4: Spring

- Days: 1, 3
- Active Start Date: March 2, 2006
- Active End Date: May 30, 2006

Keep Active for

Determines the number of extra days (beyond the original scheduling date) that the job is allowed to remain in the Active Jobs database while awaiting execution. If the job still has not run after the specified number of days, the job is removed from the Active Jobs database.

Additional information	Description
Usage	Optional
Format	▪ An integer from **0** through **98** (days) ▪ **99 or Forever** (no limit)
Default	0
Variable Name	None
Alternate names	▪ **Control-M/EM Utilities:** MAXWAIT ▪ **Reporting Facility:** MAXWAIT ▪ **Control-M/Server Utilities: -**maxwait ▪ **Control-M for z/OS:** MAXWAIT ▪ **Control-M/EM API:** maxwait
Parameter Values	
0	The job is deleted from the Active Jobs database if it did not execute on its scheduling date.
n **(n = 1-98)**	The job is retained in the Active Jobs database for the stated number of additional days beyond its original scheduling date (Odate) or until submitted for execution (and execution ends **OK**).
Forever	The job remains in the Active Jobs database indefinitely (or until it is manually deleted), even if it finishes executing and is completed ok. **NOTE:** This parameter is the same as specifying 99 in previous releases.
Computer specific information	
z/OS	If a non-cyclic job that was run on a Control-M installation on an z/OS computer was rerun by the operator and ended **NOTOK**, it is deleted from the Active Jobs database when the New Day procedure runs, regardless of the value specified in the **Keep ActiveKeep Active for** parameter

Additional information	Description
Previously Known As	Max Wait

The **Keep Active for** parameter of a job which belongs to a SMART folder or folder is overwritten by the **Keep Active for** value that belongs to the first **Rule-Based Calendar** that is fulfilled (the **Rule-Based Calendar** which is used by the job), if the relationship parameter of the job is defined as "**or**".

The **Keep Active for** parameter for a SMART Folder or sub-folder is taken from the first positive **Rule-Based Calendar**.

The **Keep Active for** parameter is used to handle the following types of occurrences:

For non-cyclic jobs

- A job that is scheduled for execution on a specific day is not always submitted that same day. This may be due to any number of reasons, such as a heavy production workload, a problem in one of the job's predecessors causing a process to stop, or a hardware failure. When the **Keep Active for** is assigned a value greater than zero, the job is retained in the Active Jobs database up to the specified number of days, allowing the job additional opportunities to be submitted.
- A job that ends **NOTOK** during the night can be deleted from the Active Jobs database during the execution of the New Day procedure the next morning. Using the **Keep Active for** parameter, the user can retain an overnight job that ends **NOTOK** in the Active Jobs database for a number of days, allowing the operator an opportunity to correct the problem and rerun the job without the need to re-order the job.

  NOTE: If the time zone job is set with the Control-M/Server time zone, and the job ends NOTOK, it remains one additional day in the Active Jobs database.

For cyclic jobs

- If **CYCLIC_MAXWAIT** (Control-M parameter) is set to **KEEP** (default), the **Keep Active for** parameter does not depend on whether the job ended **OK** or **NOTOK**. Cyclic jobs are only deleted from the Active Jobs database after the expiration of the **Keep Active for** parameter.
- If **CYCLIC_MAXWAIT** is **NOT_KEEP**, Cyclic jobs are removed from the Active Jobs database at the next run of the New Day procedure, unless they are executing when the New Day procedure begins. In this case, they are removed at the run of the following New Day procedure.

EXAMPLE:

Retain job until resources available

Retain a scheduled job indefinitely, until the runtime resources required for the job are available:

Keep Active forever

Retain job beyond original scheduling date

Retain a job for an extra three days beyond its original scheduling date:

- Days 02,04,06
- Keep Active for 03

Assume that the job does not run due to the absence of the required runtime resources. The job that is scheduled for day 2 of the month waits from the second through the fifth to be executed.

On the sixth of the month, Control-M "gives up," and the job that was originally scheduled for day 2 is deleted. The jobs scheduled for days 4 and 6 wait until as late as days 7 and 9 respectively to be executed.

Schedule a job for periods when the computer is inactive

Schedule the job for every working day, whether the computer is active. Allow each scheduled job three extra days to execute:

- Days Calendar WORKDAYS
- Retro Y
- Keep Active for 03

Given the following circumstances:

- The WORKDAYS calendar, specified in the Days Calendar parameter, contains the values 15, 16, 17, and 19.
- The computer was off-line from day 15 up to and including day 18.

When the computer is brought back online on day 19, the job is scheduled four times, with original scheduling dates (Odate) of the 15, 16, 17 and 19, respectively. Each scheduled job that does not execute successfully (, either not submitted or did not end OK) is handled as follows by Control-M:

- The job with Odate 15 is retained on day 19 and deleted from the Active Jobs database on day 20 even though the **Keep Active for** period of three days has already passed since each job with Retro Y is given at least one day to run.
- The job with Odate 16 is deleted on day 20 since day 19 is the last day of the **Keep Active for** period of three days.
- The job with Odate 17 is deleted on day 21 since day 20 is the last day of the **Keep Active for** period of three days.

The job with Odate 19 is deleted on day 23 since day 22 is the last day of the **Keep Active for** period of three days.

Keep all jobs in folder until folder is removed

Indicates that all jobs in the folder are not removed automatically from the Active Jobs database. Instead jobs wait for the folder to complete and are removed at the same time as the folder. Relevant for all jobs in the SMART folder including jobs that complete ok. Jobs are not automatically removed, but wait for the folder to complete.

NOTE: Even after the last job in a folder is removed, the folder may remain an extra day. This may occur when the folder is still executing, as Control-M may need to complete post processing folder activity, such as Do Shouts. The folder and all of the jobs are deleted in the next new day run.

Additional information	Description
Usage	Optional
Format	Check box Select the **Keep all jobs in folder until folder is removed** check box to indicate that all jobs in the folder are not removed automatically. Clear the check box to indicate that jobs are not kept in the folder until the folder is removed. Default: Clear
Variable Name	None
Alternate names	▪ **Control-M/EM Utilities:** REMOVEATONCE ▪ **Reporting Facility:** REMOVEATONCE ▪ **Control-M/Server Utilities:** -REMOVEATONCE ▪ **Control-M for z/OS:** REMOVEATONCE ▪ **Control-M/EM API:** removeatonce
Control-M/EM Utilities	Two possible values: ▪ No: Do not keep jobs in folder until folder is removed. Default. ▪ Yes: Keeps jobs in the folder until the folder is removed.
Control-M/Server Utilities	Two possible values: ▪ No: Do not keep jobs in folder until folder is removed. Default. ▪ Yes: Keeps jobs in the folder until the folder is removed.

NOTE: When you select this parameter, there is one **Keep Active for** value for the SMART folder and for the jobs within the folder. This value is the overrides the following values:

- **The Keep Active for** value specified in the folder entity itself (which enables you to select a minimal or default value that applies to all jobs in the folder). Th**e Keep Active for** value of the folder can still be set by the Rule Based Calendar which enables the folder to be ordered.
- **The Keep Active for** value of all the jobs that are ordered, which reside in the Active Jobs database. Jobs with a higher **Keep Active For** value that reside in the SMART folder, but were not ordered in the current order instance, are not counted.

The **Keep Active for** value of each job may also be set by the Rule Based Calendar that enable the job to be ordered.

Keep the folder for a minimum number of days after ended not ok

Enables you to specify a minimum period to keep the SMART folder (and jobs) in the Active Jobs database after the folder is set to NOT OK.

This parameter is enabled only when **Keep all jobs in folder until folder is removed** is selected.

Additional information	Description
Usage	Optional
Format	An integer from 1 through **98** (days), or **99** (**Forever**)
Default	1
Variable Name	None
Alternate names	▪ **Control-M/EM Utilities:** DAYSKEEPINNOTOK ▪ **Reporting Facility:** DAYSKEEPINNOTOK ▪ **Control-M/Server Utilities: -**DAYSKEEPINNOTOK ▪ **Control-M for z/OS:** DAYSKEEPINNOTOK ▪ **Control-M/EM API:** dayskeepinnotok
Parameter Values	
1	After the job ends not ok, the job is deleted the following day from Active Jobs, if the folder was marked not ok.
n **(n = 1-98)**	Indicates a minimum period in which the folder (and the jobs) are kept in Active Jobs database from the day that the folder is marked **not ok.**
99	The SMART Folder remains in the Active Jobs database indefinitely (or until it is manually deleted), even if it finishes executing.

NOTE: If the SMART folder is marked as **not ok,** but the status changes (for example, the job that causes the folder ended not ok, but was rerun or forced ok) the value resets. If the SMART folder was set to a status other than **not ok** there is no minimum period in which it is kept (the previous not ok is ignored). This feature enables you to set a minimum period from the folder failure, so you can investigate the cause of the problem. Default value: 1 day.

Days And/Or

Indicates the relationship between specified Days parameter values and Week days parameter values.

Additional information	Description
Usage	Optional
Format	Valid values: ▪ and ▪ or (default)
Invalid Characters	Non-English characters
Variable Name	None
Alternate names	▪ **Control-M/EM Utilities:** DAYS_AND_OR ▪ **Reporting Facility:** DAYS_AND_OR ▪ **Control-M/Server Utilities:** -cal_andor ▪ **Control-M for z/OS:** And/Or ▪ **Control-M/EM API:** and_or

If **And** is specified, both the Days/Days Calendar criteria and Week Days/Weeks Calendar criteria must be satisfied for a job to be scheduled.

NOTE: If the **And** option is selected (Month Days and Weekdays), and no criteria are specified for either the Month Days or the Week Days parameter, Control-M/EM assumes that **ALL** has been specified for the empty parameter. For example, if Monday is selected, and no days of the month are selected, the job is scheduled on all Mondays of the month.

If **Or** is specified, either the Days/Days Calendar criteria or Week Days/Weeks Calendar criteria must be satisfied for a job to be scheduled.

Confirmation calendar

Indicates the name of a Control-M calendar that is used to validate scheduling dates. A shift value can be used to indicate how to handle jobs that are scheduled for a non-working day in the calendar.

Additional information	Description
Usage	**Optional**
Length	Valid calendar name, up to 30 characters in length. z/OS: 1-8 characters
Case Sensitive	Yes
Invalid Characters	▪ Blanks ▪ z/OS: Non-English characters ▪ Computers other than z/OS: Single quotation marks
Variable Name	None
Alternate names	▪ **Control-M/EM Utilities:** CONFCAL ▪ **Reporting Facility:** CONF_CAL ▪ **Control-M/Server Utilities:** -confcal ▪ **Control-M for z/OS:** CONFCAL ▪ **Control-M/EM API:** conf_cal

The Calendar specified for Confirmation calendar must be a regular Calendar (not a periodic calendar). This calendar is used for:

- Validating scheduling dates
- Determining the scheduled work day.

Jobs to be scheduled on a given day are checked against the Confirmation calendar:

- If the day is a working day in the Confirmation calendar, the job is scheduled on that day. (This day is referred to as the original scheduling date.)
- If the day is not a working day in the Confirmation calendar, the Exception policy (on page 86) parameter is checked. Depending on the Shift value, the job may be scheduled on an earlier day, a later day, on the original scheduling date, or it may be cancelled.

  If the job's scheduling criteria also include the day to which it is shifted, it runs only once on that date (not once for regular scheduling, and once to make up for the shifted day).

  NOTE:

  - If no Confirmation calendar is specified, no value can be specified for the Exception policy (on page 86) subparameter, and this field has no effect on job scheduling.

- The **Confirmation calendar** parameter cannot be specified together with the **DATES** parameter.
- For z/OS jobs: The **Confirmation Calendar** parameter cannot be specified together with the **Partition data set and Minimum number of tracks** parameters. The Shift Value is always applied if specified.

Exception policy

Specifies what to do if a job is supposed to run on a day that is not in the confirmation calendar.

<table>
<tr><th>Additional information</th><th colspan="3">Description</th></tr>
<tr><td>Usage</td><td colspan="3">Optional</td></tr>
<tr><td>Format</td><td colspan="3">Drop-down list
▪ Do not order
▪ Order on next confirmed day
▪ Order on previous confirmed day
▪ Order Anyway
By:
-62 to 62</td></tr>
<tr><td>Variable Name</td><td colspan="3">None</td></tr>
<tr><td>Previously Known As</td><td colspan="3">Shift and ShiftNum</td></tr>
<tr><td colspan="4">Alternate formats in other Control-M components</td></tr>
<tr><td rowspan="2">Control-M/EM Utilities</td><td colspan="3">String. SHIFT and SHIFTNUM are included as related parameters (not as sub-parameters of CONFCAL).
EXAMPLE: <CONFCAL="Cal1"/></td></tr>
<tr><td>SHIFT</td><td>Valid values:
▪ IGNOREJOB
▪ NEXTDAY
▪ PREVDAY
▪ NOCONFCAL</td><td>Control-M/Server equivalent:
▪ No value
▪ >
▪ <
▪ @</td></tr>
</table>

Additional information	Description	
	SHIFTNUM	Number from **-62** to **62**.
Reporting Facility	String. SHIFT and SHIFTNUM are included as related parameters (not as sub-parameters of CONF_CAL).	
	SHIFT	Valid values: ▪ **Ignore Job** ▪ **Next Day** ▪ **Prev Day** ▪ **No Confcal**
	SHIFTNUM	String. Number from **-62** to **62**.
Control-M for z/OS	1-8 character string. SHIFT is specified as a separate parameter (not a sub-parameter of CONFCAL).	
	SHIFT	4-character value including the Shift Num value. For more information, see the *Control-M for z/OS User Manual*.

Cyclic

Indicates that the job must run at a designated time, interval of time.

Additional information	Description
Usage	Optional
Format	Check box ▪ **Selected** – job is cyclic ▪ **Clear** – job is not cyclic
Alternate names	▪ **Control-M/EM Utilities:** CYCLIC ▪ **Reporting Facility:** CYCLIC ▪ **Control-M/Server Utilities:** -cyclic ▪ **Control-M for z/OS:** Cyclic ▪ **Control-M/EM API:** cyclic
Sub-parameters	
Rerun Every (on page 93)	Fixed interval values to rerun a job: ▪ Minutes: 0-64,800 ▪ Hours: 0-1080 ▪ Days: 0-45
All intervals are from job's (on page 91)	Valid Values: ▪ End ▪ Start ▪ Target
Rerun using the following interval sequence (on page 92)	List of time intervals: ▪ Unit: Minutes, Hours, Days ▪ Amount:0-64,8000; 0-1080; 0-45
Run at (on page 95)	A list of specific times for the job to run. **NOTE:** This parameter supports time synonym.
Tolerance (on page 95)	Maximum delay in minutes permitted for a late submission when selecting a specific time (e.g. 5 minutes). Valid range: 0-999

A non-cyclic job is a job that, if its scheduling criteria are satisfied, is ordered once by Control-M on a given day (discounting reruns caused by a Rerun Job (on page 163) parameter or manual reruns).

A cyclic job is rescheduled after execution for an additional possible execution. The job executes again only when the following circumstances occur:

- The first run has completed.
- The runtime schedule and execution parameters are still satisfied.
- A specified number of minutes has elapsed since the last completion of the job or the next specified runtime has been reached.

The cyclic job runs at the time designated according to the time or time intervals selected. The Run at (on page 95) and the Tolerance (on page 95) parameters can be set to enable a job to run after the specified time. If the job being executed runs over the proceeding job's specified time, the proceeding job's execution time window is extended to the number of minutes set in the **Tolerance** field. For example, if the **Tolerance** field is set to 15 minutes, the proceeding job can still be executed 0-15 minutes after the specified time. If the tolerance time interval has passed, the proceeding job will not be performed.

NOTE: The **Run at** option is relevant for one odate only. Specific times are sorted from each new day time to the next.

Ordering a cyclic job as "non-cyclic"

The %%CYCLIC variable can be used to override the Cyclic parameter for a job, or for all jobs in a folder.

This variable is normally used when ordering a cyclic job for a single run.

NOTE: To indicate that all jobs ordered by the ctmorder utility (in Control-M/Server) should run as non-cyclic jobs, include the following statement in the command line of the ctmorder utility.

-variable %%CYCLIC N

Maximum number of days to wait for submission

The CYCLIC_MAXWAIT parameter for Control-M/Server determines when cyclic jobs, which have executed at least once, should be removed from the Active Jobs database by the New Day procedure.

Valid values are:

- **KEEP** – Jobs are removed from the Active Jobs database when Keep Active for (on page 79) days have passed regardless of whether the job ended **OK**. Default.
- **NOT_KEEP** – Jobs (non-cyclic and cyclic) are removed from the Active Jobs database at the next run of the New Day procedure. Cyclic jobs are not removed if they are executing when the New Day procedure begins. Instead, they are removed at the run of the following New Day procedure.

For more information about Control-M/Server parameters, see Control-M/Server parameters in *Control-M Administration*.

Special care should be taken when specifying a cyclic job. If not defined appropriately it may cause an endless loop in which the job is continually resubmitted for execution.

If a cyclic job is executing when the New Day procedure is run, the job is changed to non-cyclic and an appropriate message is written to the Control-M log. It is the user's responsibility to review these messages and handle them accordingly.

Terminating a cyclic job

You can prevent subsequent iterations of a cyclic job by using the Do Stop Cyclic parameter.

If a cyclic job is terminated by a Stop Cyclic Run **(on page** 164**)** parameter, the View Details screen displayed by option Z in the ctmpsm utility contains **Cyclic:T** where **T** indicates "Terminated".

For z/OS jobs

Cyclic jobs cannot contain Rerun Job (on page 163).

Next expected run

Specifies the next expected submission date and time for the job, for rerun or cyclic jobs that use the Interval option.

This parameter is relevant for z/OS only.

Additional information	Description
Usage	Optional
Format	Date calendar
Variable Name	None
Alternate Names	▪ **Control-M/EM Utilities:** NEXT_RUN ▪ **Reporting Facility**: NEXT RUN ▪ **Control-M/Server Utilities:** -nextrun ▪ **Control-M/EM API:** next run
Sub-parameters	
Date	Calendar ▪ dd-mm-yy ▪ 01-Jan-12
Time	**HH:MM**

All intervals are from job's

Indicates whether the interval between runs of a cyclic job or until the start of a rerun job is measured from the start or the end of the previous job run.

Additional information	Description
Usage	Optional
Format	List box Valid values: ▪ **Start** (Default) ▪ **End** ▪ **Target**
Invalid Characters	Non-English characters
Variable Name	None
Alternate names	▪ **Control-M/EM Utilities:** IND_CYCLIC ▪ **Reporting Facility:** IND CYCLIC ▪ **Control-M/Server Utilities:** -intervalfrom (Default value: START) ▪ **Control-M for z/OS:** (none) ▪ **Control-M/EM API:** count_cyclic_from

The interval between job runs is specified in the Interval parameter.

- When the value is **Start**, the time until the next job run is counted from the moment that the current job run begins.
- When the value is **End**, the time until the next job run is counted from the moment that the current job run is complete.
- When the value is **Target**, the interval is measured from the scheduling time of the current job run.

EXAMPLE: Calculate the interval between cyclic jobs when value is Start

Job_A is a cyclic job. The value specified for Interval is 60 seconds. The value is **Start**.

If Job_A job run takes 15 seconds, the next run of Job_A begins 45 seconds after the first run is complete.

Calculate the interval between cyclic jobs when value is End

Job_B is a cyclic job. The value specified for Interval is 60 seconds. The value is **End**.

When the Job_B job run is complete, the next run of Job_B begins 60 seconds after the first run is complete. The length of time that it takes to run Job_B does not affect the period of time between job runs.

Rerun using the following interval sequence

Specifies a unit of time and a numerical value that defines the interval sequence for a job to rerun.

Additional information	Description
Usage	Optional
Format	Drop-down list Units valid values: ▪ **Minutes** (Default) ▪ **Hours** ▪ **Days** Amount valid values: 0-64,8000; 0-1080; 0-45
Invalid Characters	Non-English characters
Variable Name	None
Alternate names	▪ **Control-M/EM Utilities:** (none) ▪ **Reporting Facility:** (none) ▪ **Control-M/Server Utilities:** The Units value is contained in the -interval parameter ▪ **Control-M for z/OS:** (none) ▪ **Control-M/EM API:** (none)
Alternate formats	
Control-M/Server Utilities	Valid values: ▪ **d** - Days ▪ **h** - Hours ▪ **m** - Minutes (Default) Format: <interval_amount><unit> **EXAMPLE:** For 24 hours: **24h**

Rerun Every

Specifies the length of time to wait between reruns of a job or between cyclic runs of a job.

Additional information	Description
Usage	Optional **NOTE:** When defining a cyclic job with a **Rerun every** parameter with the default value of **0,** if the default is not changed, the job runs continuously when submitted for execution.
Format	Valid values: ▪ **0** to **64800** (for minutes) ▪ **0** to **1080** (for hours) ▪ **0** to **45** (for days) Default: **0** The unit of measurement is determined by the Rerun using the following interval sequence (on page 92) parameter. If no value is specified, the default is **Minutes**.
Case sensitive	No
Invalid Characters	Blanks; single quotation marks
Variable Name	None
Alternate names	▪ **Control-M/EM Utilities:** INTERVAL ▪ **Reporting Facility:** INTERVAL ▪ **Control-M/Server Utilities:** -interval ▪ **Control-M for z/OS:** Rerun every ▪ **Control-M/EM API:** rerun_interval
Alternate Formats	
Control-M for z/OS	Rerun Every combines the functionality of the All intervals are from job's, and Rerun using the following interval sequence parameters.

Additional information	Description
	Valid values: ▪ 0 to 64800 (for minutes) ▪ 0 to 1080 (for hours) ▪ 0 to 45 (for days) ▪ Days– Maximum value is 45. ▪ Hours – Maximum value is 1080. ▪ Minutes – Maximum value is 64800. Default.

This field indicates if the time for the next run of the job should be calculated from the beginning or from the end of the previous run of the job.

A job can be run more than once from a given job order when

- The job's completion status was set to Rerun using a Rerun Job (on page 163) parameter
- The job is defined as Cyclic

Control-M waits at least the number of minutes specified by the **Rerun every** parameter before it attempts the next rerun of the job or before the next run of a cyclic job.

The **Rerun every** period can be calculated from either the start or the end of the previous job run, as determined by the Rerun using the following interval sequence (on page 92) parameter.

The job is re-submitted after

- The specified number of minutes have elapsed from the last rerun, or from the start or end of the last run of a cyclic job

-and-

- All submission criteria are satisfied

Run at

Defines the specific times for the job to run.

Additional information	Description
Usage	Optional
Format	Time: *hhmm*
Case sensitive	No
Invalid Characters	Blanks; single quotation marks
Variable Name	None

Tolerance

Maximum delay in minutes permitted for a late submission when selecting a specific time (e.g. 5 minutes).

Additional information	Description
Usage	Mandatory of you select the Run at (on page 95) parameter.
Format	Valid values: Minutes: **0** to **999**
Case sensitive	No
Invalid Characters	Blanks; single quotation marks
Variable Name	None

Using Calendar/Calendar

Specifies the name of the calendar that was defined in the Calendar Manager.

Additional information	Description
Usage	This parameter can be used when the **Based on Calendar** option is selected. In this case it is Mandatory. It can also be used when the **Advanced Scheduling option** is selected. In this case it is optional.
Variable Name	None
Alternate names	▪ **Control-M/EM Utilities:** CALENDAR ▪ **Reporting Facility:** CALENDAR ▪ **Control-M/Server Utilities: -**calendar ▪ **Control-M for z/OS:**CALENDAR ▪ **Control-M/EM API:** calendar

Rule-Based Calendar

Identifies a set of scheduling criteria defined for a folder. The scheduling criteria referenced by each Rule-based Calendar are defined in the outermost folder.

Additional information	Description
Usage	Optional

Additional information	Description
Format	This consists of the following parameters: ▪ Calendar Name ▪ Control-M/Server (on page 31) ▪ Scheduling options ▪ Confirmation calendar (on page 85) ▪ Exceptions ▪ Keep Active for (on page 79) ▪ Active/Not Active between dates (on page 76) ▪ Start Date (on page 99) ▪ End Date (on page 99)
Variable Name	None
Alternate names	▪ **Control-M/EM Utilities:** RBC_NAME ▪ **Reporting Facility:** (none) ▪ **Control-M/Server Utilities:** -rbc ▪ **Control-M for z/OS:** SCHEDULE RBC ▪ **Control-M/EM API:** sched_rbc
Alternate formats	
Control-M/EM Utilities	RBC_NAME parameters are contained in the RBC_NAMES parameter. The RBC_NAME value is a string. RBC_NAMES cannot have a value. **EXAMPLE:** *<RBC_NAMES RBC_NAME="T1" RBC_NAME="T2"/>*
Subparameter	
Calendar Name	Specifies the name of the name of the calendar. ▪ 1-19 characters ▪ Case sensitive ▪ Invalid characters include: Blanks, ! at the beginning of a calendar name Computers other than z/OS: Single quotation marks Z/OS: * (asterisk), ? (question mark), non-english characters

Each folder contains one or more sets of basic scheduling criteria that can be applied to job processing definitions of jobs in the folder. Each set of basic scheduling criteria is assigned a unique name, specified in the RBC option, which is used for referencing that set of criteria.

To apply a set of scheduling criteria in a folder to a job processing definition, specify the RBC name of the desired criteria in the RBC field.

If multiple RBC names are specified in the job processing definition, the RBCs are checked sequentially (according to the order in which they are defined for the folder) during job scheduling to determine if the criteria are satisfied. The first set of RBC criteria that is satisfied is applied to the job. Subsequent Rule-Based Calendars specified for the job are not checked.

If you select the **Use Parent Rule-based Calendar** checkbox, all Rule-Based Calendars in the folder are applied to the job.

Each job processing definition can have its own basic scheduling criteria defined, independent of the RBC criteria in the folder.

Scheduling jobs in a folder

Jobs in a folder are eligible for scheduling on a particular day only if at least one RBC in the folder is satisfied.

If a folder is eligible for scheduling on a particular day, a job in the folder is scheduled in either of the following cases:

- The value of the Relationship parameter is **OR**. The basic scheduling criteria of the job or a specified RBC (or both) are satisfied.
- The value of the Relationship parameter is **AND**. Both the basic scheduling criteria of the job and a specified RBC are satisfied.

Example: Two sets of Rule-Based Calendars

Folder **ACCOUNTS** for group **ACCOUNTS_GROUP** contains two sets of scheduling criteria Rules-Based Calendars.

One set is identified by the **ALL_DAYS** RBC, and the other set is identified by the **SUNDAYS** RBC.

The following information is specified for **ALL_DAYS** RBC:

Week DaysALL

The following information is specified for **SUNDAYS** RBC:

Week Days1

For jobs that should run on any day, specify:

Schedule RBCALL-DAYS

For jobs that should only run on Sundays, specify:

Schedule RBCSUNDAYS

Start Date

Sets the date when the job must be ordered.

Additional information	Description
Usage	Optional
Format	Calendar ▪ dd-mm-yy ▪ 01-Jan-12
Variable Name	None

End Date

Sets the date when the job must be ordered.

Additional information	Description
Usage	Optional
Format	Calendar ▪ dd-mm-yy ▪ 01-Jan-12
Variable Name	None

Maximum reruns

Determines the maximum number of reruns that can be performed for the job.

Additional information	Description
Usage	Optional
Length	An integer from **0** through **99**. z/OS: There is a difference between cyclic jobs and regular jobs: ▪ Cyclic jobs – an integer from **0** through **9999** ▪ Regular jobs – an integer from **0** through **255**.
Default	0
Variable Name	None
Alternate names	▪ **Control-M/EM Utilities:** MAXRERUN ▪ **Reporting Facility:** MAX RERUN ▪ **Control-M/Server Utilities:** -maxrerun ▪ **Control-M for z/OS:** MAXRERUN ▪ **Control-M/EM API:** rerun_max
Computer specific information	
z/OS	When a job is first run, the **Maximum reruns** field in the Active Jobs database, that is, in the Zoom screen, contains the same value as the **Maximum reruns** parameter in the job scheduling definition. However, in the Active Jobs Database Maximum reruns works as a "reverse-counter" of automatic reruns. Each time the job is automatically rerun, the value is decreased by one until the field contains a value of zero.

When the job's completion status is set to Rerun using the Rerun Job (on page 163) parameter, Control-M checks the number of reruns specified in the **Maximum reruns** parameter. If the number in the **Maximum reruns** parameter is greater than the number of reruns that have already been performed for the job, a rerun (automatic rerun) process is performed for the job.

When a job is assigned **Rerun** status, the job is not rerun if either

- Maximum = 0
- The number of reruns specified in this parameter has already been performed

Control-M waits at least the number of minutes specified by the Interval parameter before it attempts the next rerun of the job.

The job is re-submitted after

- The specified number of minutes has elapsed from the last rerun

-and-

- all submission criteria are satisfied

A rerun counter is displayed on the Monitoring domain of the job properties pane, indicating how many times the job has been rerun from the current job order.

The automatic rerun process works as follows:

- Control-M determines that automatic rerun is possible only if the job ENDS **NOTOK** and a specified Rerun Job (on page 163) statement is activated during post-processing. If the monitor determines that automatic rerun is possible, it sets the status of the job to ENDED NOTOK – RERUN NEEDED.
- The monitor then checks the value of **Maximum reruns** in the Active Jobs database. If the value is zero, automatic rerun is not possible and the job is not submitted for rerun. If the value is greater than zero, rerun is possible and the monitor submits the job for rerun when all runtime criteria are satisfied.
- The JCL for the rerun job is taken from the member specified in the RERUNMEM parameter. If no RERUNMEM value is specified, the JCL for the rerun is taken from the regular JCL member of the job that is specified in the MEMNAME parameter.

Maximum reruns applies only to automatic reruns. The **Maximum reruns** counter is not affected by reruns performed manually using the Rerun option in the Control-M for z/OS Active Jobs database screen.

If a job is defined as cyclic, the **Maximum reruns** parameter can be used to specify the number of iterations. This number excludes the initial run of the job.

Minimum number of tracks

Minimum number of free partitioned data set tracks required by the library specified for the Partition Data Set parameter.

NOTE: This parameter is available for z/OS jobs only.

Additional information	Description
Usage	Mandatory, if a value is specified for the **Partition Data set** parameter.
Format	A positive 3-digit number (leading zeros are required)
Variable Name	None

Additional information	Description
Alternate names	▪ **Control-M/EM Utilities:** MIN ▪ **Reporting Facility:** MIN PDS TRAC ▪ **Control-M/Server Utilities:** (none) ▪ **Control-M for z/OS:** MINIMUM ▪ **Control-M/EM API:** min_pds_tracks

The **Partition Data Set** parameter specifies a library, and the **Minimum number of tracks** parameter specifies the minimum number of free tracks required by that library.

The **Minimum number of tracks** and **Partition Data set** parameters are always used together and are never used with other Basic Scheduling parameters

When to use

The **Minimum number of tracks** and **Partition Data set** parameters are intended for use in jobs and started tasks that compress, clean, and/or enlarge libraries. They are also specified for tasks that issue a warning message to the IOA Log file if the minimum number of free tracks is not available.

If **Minimum number of tracks** and **Partition Data Set** parameters are defined for a job, the scheduling of the job is not related to or depends on any date criteria. Instead, the job is scheduled if the actual number of free tracks available in the specified library is below the specified minimum at time of daily job ordering. The job or started task can then compress, clean, or enlarge the library (or issue the appropriate warning).

NOTE: Minimum number of tracks does not work with PDSE-type libraries because they always appear to be 100 percent full. Minimum number of tracks only checks current extents.

Order on months

Determines which months the job can be scheduled for processing.

Additional information	Description
Usage	Optional
Format	A check box is displayed corresponding to each month of the year. When selected, the job is scheduled for that month. Default: The job is scheduled for all months.
Invalid Characters	Blanks; single quotation marks; non-English characters
Variable Name	None
Alternate names	
Control-M/EM Utilities	Indicate the specific month: ▪ **JAN** ▪ **FEB** ▪ **MAR** ▪ **APR** ▪ **MAY** ▪ **JUN** ▪ **JUL** ▪ **AUG** ▪ **SEP** ▪ **OCT** ▪ **NOV** ▪ **DEC**
Reporting Facility	Specify the specific month: ▪ **MONTH_1** (January) ▪ **MONTH_2** (February) and so on... ▪ **MONTH_12** (December)

Additional information	Description
Control-M/Server Utilities	-month
Control-M for z/OS	MONTHS
Control-M/EM API	Indicate the specific month: ▪ **JAN** ▪ **FEB** ▪ **MAR** ▪ **APR** ▪ **MAY** ▪ **JUN** ▪ **JUL** ▪ **AUG** ▪ **SEP** ▪ **OCT** ▪ **NOV** ▪ **DEC**
Alternate formats	
Control-M/EM Utilities	Indicate for each month. ▪ **0** (Do not schedule for this month. Default) ▪ **1** (Schedule for this month.) **EXAMPLE:** <JAN="1"/>

The **Months** parameter cannot be used together with the **Dates** parameter.

Partition Data set

Defines the name of a partitioned data set to check for free space. If the Partition Data Set has fewer than the minimum number of required free tracks (as specified for the Minimum number of tracks parameter), the job is executed.

NOTE: This parameter is available for z/OS jobs only.

The **Partition Data set** and **Minimum number of tracks** parameters must be specified together.

Additional information	Description
Usage	Mandatory, when a value is specified for the **Minimum number of tracks** parameter
Length	1-44 characters
Case Sensitive	No
Invalid Characters	Blanks; non-English characters
Variable Name	None
Alternate names	▪ **Control-M/EM Utilities:** PDSNAME ▪ **Reporting Facility:** PDS Name ▪ **Control-M/Server Utilities:** (none) ▪ **Control-M for z/OS:** PDS ▪ **Control-M/EM API:** pds

The data set must be cataloged, and it must be a partitioned data set.

Parameters **Minimum number of tracks** and **Partition Data set** are always used together and are never used with other Scheduling parameters.

The **Partition Data Set** parameter identifies a library. The **Minimum number of tracks** parameter specifies the minimum number of free tracks required by that library.

These parameters are intended for use (that is, definition) in jobs or started tasks that compress, clean and/or enlarge libraries, or which issue a warning message to the IOA Log file.

If the **Minimum number of tracks** and **Partition Data Set** parameters are defined for a job, the scheduling of the job is not related to or depends on any date criteria. Instead, the job is scheduled if the actual number of free tracks available in the specified library is below the specified minimum when the New Day procedure is run.

The job or started task can then compress, clean, or enlarge the library (or issue the appropriate warning).

NOTE: This parameter is not supported for PDSE-type libraries because they always appear to be 100 percent full.

Relationship

Indicates the relationship (AND/OR) between RBCs criteria and basic scheduling criteria in the job processing definition (that is, whether either set of criteria, or both sets of criteria, must be satisfied).

Additional information	Description
Usage	For jobs that are in folders.
Format	Option buttons. When **or** is selected, if either set of criteria (a specified RBC or the job's basic scheduling criteria) is satisfied, the job is scheduled. Default. When **and** is selected, both a specified RBC and the job's basic scheduling criteria must be satisfied for the job to be scheduled.
Invalid Characters	Non-English characters
Variable Name	None
Alternate names	▪ **Control-M/EM Utilities:** RBC_RELATIONSHIP ▪ **Reporting Facility:** RBC_RELATIONSHIP ▪ **Control-M/Server Utilities:** -relationship ▪ **Control-M for z/OS:** RELATIONSHIP ▪ **Control-M for z/OS Utilities:** RELATION ▪ **Control-M/EM API:** rbc_relationship

For jobs in folders, two types of basic scheduling criteria can be specified:

- Rule-Based Calendars. Pointers to sets of scheduling criteria defined in the folder.
- Basic Scheduling Criteria. Defined in each job processing definition.

In some cases, both sets of criteria must be satisfied for the job to be scheduled. In other cases, it is sufficient if either set of criteria is satisfied. The AND-OR option enables you to specify the required combination:

- When either set of criteria is sufficient choose **OR**.
- When both sets of criteria are required, choose **AND**.

  NOTE:

  - If an AND relationship is specified and no Rule-Based Calendars are defined in the job, the job is not scheduled.
  - This parameter is relevant only to the job processing definitions of jobs that are in folders.

Rerun Member

Defines the name of the JCL member to use when the job automatically reruns.

This field is relevant only for z/OS jobs.

Additional information	Description
Usage	Optional
Length	1-8 characters
Case Sensitive	Yes
Invalid Characters	Blanks; non-English characters
Variable Name	None
Alternate names	▪ **Control-M/EM Utilities:** IRERUNMEM ▪ **Reporting Facility:** RERUN MEM ▪ **Control-M/Server Utilities:** (none) ▪ **Control-M for z/OS:** RERUNMEM ▪ **Control-M/EM API:** rerun_member

Although the **Rerun Membe**r parameter can be used to specify the name of a JCL member to use for automatic rerun, note the following points:

- The Order Job (on page 154) parameter provides a more flexible alternative to the Rerun Member parameter.
- Control-M/Restart users can use the Set Restart Options (on page 183) parameter to restart the failed job instead of using the **Rerun Member** parameter to rerun the job.

The automatic rerun process works as follows:

- Control-M determines that automatic rerun is possible only if the job ends **NOTOK** and a specified Rerun Job (on page 163) statement is activated during post-processing. If Control-M determines that automatic rerun is possible, it sets the job's status to ended **NOTOK** – RERUN NEEDED.
- Control-M then checks the value of Maximum reruns (on page 100) in the Active Jobs database. If the value is zero (or no maximum reruns value was specified), automatic rerun is not possible and the job is not submitted for rerun. If the value is greater than zero, rerun is possible and the monitor submits the job for rerun when all runtime criteria are satisfied. Runtime criteria include not only the Runtime Scheduling parameters, but also the Rerun using the following interval sequence (on page 92) parameter, which specifies the minimum allowable interval between runs of the same job.
- The JCL for the rerun job is taken from the member specified in the **Rerun Member** parameter. If no Rerun Member value is specified, the JCL for the rerun is taken from the job's regular JCL member specified in the File name/Member name (on page 28) parameter.

Rerun Member has additional characteristics:

- The member name can be the same as, or different from, the job name.
- The member specified in the **Rerun Member** parameter must be in the library specified in the File Path/Member library (on page 25) parameter.
- The **Rerun Member** parameter overrides the **Member name** parameter value in the JCL, and the **Member name** parameter value becomes irrelevant for reruns.

Availability

- This parameter is relevant only for z/OS jobs.
- The **Set Restart Options** and **Rerun Member** parameters cannot be specified together.
- The **Rerun Member** parameter cannot be specified for cyclic jobs and cyclic started tasks.

Retroactively order job that its scheduled date has passed

Indicates if the job should be scheduled for possible execution after its original scheduling date has passed.

Additional information	Description
Usage	Optional
Format	Check box Select the **Retroactively order job that its scheduled date has passed** check box to indicate that the job should be scheduled retroactively. Clear the check box to indicate that the job should not be scheduled retroactively. Default: Clear
Invalid Characters	Non-English characters
Variable Name	None
Alternate names	▪ **Control-M/EM Utilities:** RETRO ▪ **Reporting Facility:** RETRO ▪ **Control-M/Server Utilities:** -retro ▪ **Control-M for z/OS:** Retroactively order job that its scheduled date has passed ▪ **Control-M/EM API:** retro
Alternate formats	

Additional information	Description
Control-M/EM Utilities	RETRO has two possible values: ▪ **0** (No) – Do not allow scheduling of the job after its original scheduling date has passed. Default. ▪ **1** (Yes) – Allow scheduling of the job after its original scheduling date has passed. **EXAMPLE**: <RETRO="1"/>
Control-M/Server Utilities	RETRO has two possible values: ▪ N (No) – Do not allow scheduling of the job after its original scheduling date has passed. Default. ▪ **Y** (Yes) – Allow scheduling of the job after its original scheduling date has passed.

The **Retroactively order job that its scheduled date has passed** parameter is used to control situations where the computer has not been working for a day or more (for example, due to a weekend, a holiday or a hardware failure). The **Retroactively order job that its scheduled date has passed** parameter indicates to the New Day procedure or User Daily if the job should be retroactively scheduled for days the computer (or Control-M) was inactive.

- If **Retroactively order job that its scheduled date has passed** is specified, Control-M/Server places job orders in the Active Jobs database for all days that the job should have been scheduled. The scheduling dates for which job orders are issued range from the day following the last running date of the New Day procedure or User Daily to the current working date, provided that those dates satisfy criteria specified in one of the scheduling parameters (Days, Days Calendar, and so on). Each job order placed in the Active Jobs database uses a date in the range as its original scheduling date.
- If **Retroactively order job that its scheduled date has passed** is not specified, the job is scheduled only for the current working date (provided that this date satisfies the job's scheduling criteria).
- If **Retroactively order job that its scheduled date has passed** is specified for a User Daily Job, only the last (most recent) order of the User Daily Job actually orders jobs. It also performs Retro ordering for all included jobs. As a general rule, there is no need to specify Retro in the job processing definition of a User Daily job.

  You can use the ctmrpln Control-M/Server utility to determine when selected jobs in a folder are scheduled to run.

  Retroactively order job that its scheduled date has passed cannot be used with the Minimum number of tracks and Partition data Set parameters.

No SMART folder support

The **Retroactively order job that its scheduled date has passed** parameter is not supported for SMART folders. If this parameter is selected for a SMART folder, it is ignored.

The **Retroactively order job that its scheduled date has passed** parameter can be selected for individual jobs in a SMART folders

EXAMPLE: Schedule a job on a specific day of the month

Schedule the job on specific days in the month. If the day has passed, do not schedule the job.

Days 15, 16, 18, 19, 20

Retroactively order job that its scheduled date has passed No

Assume the computer was down from the 16th to (and including) the 18th. The 15th is the last date on which the job was scheduled for execution. Today is the 19th. Therefore, the job is only scheduled for execution on the 19th.

EXAMPLE: Schedule a job for every workday

Schedule the job for every workday, whether the computer is active:

Days Calendar WORKDAYS

Retroactively order job that its scheduled date has passed Yes

Given the following circumstances

the Work Days calendar contains the dates 15, 16, 18, and 19

the computer was down from the 16th up to (and including) the 18th

the 15th was the last date that the job was scheduled for execution

today is the 19th

The job is scheduled three times with the original scheduling dates: 16, 18, and 19 (that is, three job orders are added to the Active Jobs database).

SAC

Determines whether to adjust the logical date for a job converted from a scheduling product other than Control-M.

This parameter is relevant only for jobs running in Control-M for z/OS.

Additional information	Description
Usage	Optional
Format	The following options are available: ▪ **Blank** – No adjustment is made. The SMART folder and all the jobs in it are scheduled according to the regular criteria. This is the default. ▪ **P** (Previous) – The operation of this value depends on whether it is in a folder or a job scheduling definition. If this value is set for a SMART folder, the SMART folder is scheduled both on the day indicated by the regular scheduling criteria and on the day preceding that day. If this value is set for a job, the job is scheduled on the day preceding that indicated by the regular scheduling criteria. ▪ **N** (Next) – The operation of this value depends on whether it is in a folder or a job scheduling definition. If this value is set for a SMART folder, the SMART folder is scheduled both on the day indicated by the regular scheduling criteria and on the day following that day. If this value is set for a job, the job is scheduled on the day following that indicated by the regular scheduling criteria. **+** (Plus) – [For folder scheduling definitions only] The scheduling of the SMART folder is adjusted to the next logical date. **-** (Minus) – [For folder scheduling definitions only] The scheduling of the SMART folder is adjusted to the previous logical date.
Variable Name	None
Alternate names	▪ **Control-M/EM Utilities:** PREV_DAY ▪ **Reporting Facility:** Prev Day ▪ **Control-M/Server Utilities:** (none) ▪ **Control-M for z/OS:** SAC ▪ **Control-M/EM API:** sac

Use this parameter if one of the following is true:

- You have migrated to Control-M/EM from another job scheduling product that did not enable you to select the New Day time.

- You have selected a New Day time that is different from the time at which the logical date changed under your old product.

In such circumstances, jobs may be scheduled at a time that falls between the time at which the logical date changed under your former product and the Control-M New Day time. This may cause jobs to be scheduled on the wrong date. The SAC parameter enables the logical date to be automatically adjusted in the case of such jobs, so that they run on the required date.

EXAMPLE: In the MIXEDTBL folder, the JOBA, JOBB, and JOBC jobs in the folder have the SAC parameter set to **P** and the JOBX, JOBY, and JOBZ jobs have the SAC parameter left blank. In such a case, the folder must be scheduled on both the regular scheduling day for the folder and on the previous day, so that both types of jobs can run.

Set the SAC parameter of the MIXEDTBL folder to **P**.

EXAMPLE: In the ONEVAL folder, all the jobs have the SAC parameter set to **P**. In such a case, the folder must be scheduled on the day preceding the regular scheduling day only.

Set the SAC parameter of the ONVAL folder to **-**.

Schedule

Specifies the type of scheduling to use.

Additional information	Description
Usage	Mandatory
Format	Drop-down list
Variable Name	None
Alternate Names	▪ **Control-M/EM Utilities:** SCHEDULE ▪ **Reporting Facility**: SCHEDULE ▪ **Control-M/Server Utilities:** -schedule ▪ **Control-M/EM API:** schedule

Statistics Calendar

Name of the Control-M periodic calendar within which statistics relating to the job are collected.

NOTE: This parameter is relevant only for jobs running in Control-M for z/OS .

Additional information	Description
Usage	Optional
Format	A valid period calendar name consisting of from 1 through 8 alphanumeric characters.
Case Sensitive	Yes
Invalid Characters	▪ Blanks ▪ Non-English characters
Variable Name	None
Alternate names	▪ **Control-M/EM Utilities:** STAT_CAL ▪ **Reporting Facility:** Stat Calendar ▪ **Control-M/Server Utilities:** (none) ▪ **Control-M for z/OS:** STAT CAL ▪ **Control-M/EM API:** statistic_cal

As part of the post-processing for each job, Control-M for z/OS determines the elapsed run time of the job. All accumulated information regarding job execution, including the elapsed run time, is written to the IOA Log file.

Periodically, a statistics utility may be used to scan and analyze the IOA Log file. This utility gathers information about the start time of each job, its elapsed run time, CPU utilization time, and so on. The utility places this information in the Statistics file, where averages of these values can be maintained for each job.

If the **Statistics Calendar** parameter is not used to specify a periodic calendar, the statistics relating to a job are based on all run times of the job.

The **Statistics Calendar** parameter identifies the periodic calendar that is used to gather average runtime statistics for the job. The information displayed in the Statistics Period field identifies the specific days within that calendar that are used as the basis of the runtime statistics.

By using the **Statistics Calendar** parameter together with the information displayed in the **Statistics Period** field, you can obtain more precise statistical information about the running of the job, as shown in the following example:

EXAMPLE: Assume that a job runs daily, weekly, and monthly, and that the **Statistics Calendar** parameter identifies a periodic calendar that contains a number of months each specified in a manner similar to the following:

```
-----S----------S----------S----------S----------S----------S---
                1 2 3 4 5 6 7 8 9 + 1 2 3 4 5 6 7 8 9 + 1 2 3 4 5 6 7 8 9 +
 09             D D W D    D D D D W D    D D D D W D    D D D D W D    D D D M
```

In this example, the job runs daily in Period D, weekly in Period W, and monthly in Period M.

If the job runs on the 3rd of the month, its statistics are collected for Period W. If it runs on the 6th of the month, its statistics are collected for Period D, and so on.

Statistics Period

Identifier of the actual days within the Control-M periodic calendar in relation to which statistics relating to the job are calculated.

NOTE: This parameter is relevant only for jobs running in Control-M for z/OS

Additional information	Description
Usage	Optional
Format	1 alphabetic display only
Variable Name	None

This parameter provides more precise statistical information relating to z/OS jobs.

For full information, see Statistics Calendar (on page 112).

From Time/To Time

Sets time limits for submitting the job.

Additional information	Description
Usage	Optional
Format	If not left blank, the **From** and **To** sub-parameters must contain valid times expressed in 24-hour format (**HH:MM**). By default, valid times are from **00:00** through **23:59**. However, if Time Synonym (in File/Options>Job Properties) is checked, valid times are from New Day time until New Day Time plus 23:59. For example, if New Day time is 8:00 A.M., the range of valid times is from 08:00 until 31:59. Alternatively, if **From** contains a valid time value, **To** can contain the **>** character. For more information, see below. The **From** value can be specified without a **To** value. The opposite is also true.
Variable Name	None

Additional information	Description
Alternate names	**Control-M/EM Utilities:** ▪ TIMEFROM ▪ TIMETO **Reporting Facility:** ▪ FROM TIME ▪ TO TIME **Control-M/Server Utilities:** ▪ -timefrom ▪ -timeuntil **Control-M for z/OS:** ▪ TIME FROM ▪ TIME UNTIL **Control-M/EM API:** ▪ time_from ▪ time_until
Sub-parameters	
Time	**HH:MM** For z/OS jobs, includes the day
Any time on next day	Optional sub-parameter, valid only for Control-M for z/OS that enables you to set a range of days after the original scheduling date of the job during which the job can be submitted. From + num days defines the beginning of the range and Until + num days defines the end of the range, where the variable num is a number of days after the Odate (original scheduling date) of the job. Valid values for num are any number from 1 through 120.
End of day	Optional sub-parameter that enables you to set the job to be submitted by the end of day.
Allow submission past next new day	Optional sub-parameter that enables you to set the the job to be submitted after its original scheduling date.

EXAMPLE: Submit the job between 9 A.M. and 6 P.M.

Time From 09:00 + days Until 18:00 + days

Submit the job at any time after 6 P.M.

Time From 18:00 + days Until > + days

In this example, the job can be submitted at any time after 6 P.M. on the day of submission. If the job has not yet been submitted when the New Day time at the site arrives, it is submitted as soon as possible, even at a later date, until the period set by the Keep Active for parameter expires.

The New Day time at the site is 6:00 A.M. Submit the job after the New Day time, but no later than 7:30 A.M.

Time From + days Until 07:30 + days

In this example, the job can only be submitted between 6:00 A.M. and 7:30 A.M. If the job is still on the Active Jobs database on the following day, it will again be submitted between 6:00 A.M. and 7:30 A.M. on that day.

The New Day time at the site is 6:00 A.M. Submit the job at any time between midnight and the New Day time.

Time From 00:00 + days Until + days

In this example, the job can only be submitted between midnight and 6:00 A.M.

The New Day time at the site is 1:00 A.M. Submit the job at any time between 11:00 P.M. and 6:00 A.M.

Time From 23:00 + days Until 06:00 + days

In this example, the job is eligible for submission between 11:00 P.M. and 1:00 A.M., and between 1:00 A.M. and 6:00 A.M.

The New Day time at the site is 1:00 A.M. and the Odate is September 6. Submit the job at any time between 11:00 P.M. on September 9, and 6:00 A.M. on the sixth day after the Odate of the job

Time From 23:00 + 003 days Until 06:00 + 006 days

The New Day time at the site is 1:00 A.M. Submit the job at any time between 11:00 P.M. on the third day after the Odate of the job, and the New Day time on the sixth day after the Odate of the job

Time From 23:00 + 003 days Until + 006 days

In this example, the job is eligible for submission between 11:00 P.M. on the third day after the Odate of the job and 1:00 A.M. on the sixth day after the Odate of the job.

The New Day time at the site is 1:00 A.M. Submit the job at any time between 11:00 P.M. on the third day after the Odate of the job, and 1:00 A.M. on the sixth day after the Odate of the job

Time From + 003 days Until 06:00 + 006 days

In this example, the job is eligible for submission between 1:00 A.M. on the third day after the Odate of the job and 6:00 A.M. on the sixth day after the Odate of the job.

The New Day time at the site is 1:00 A.M. Submit the job at any time between 1:00 A.M. on the third day after the Odate of the job, and 1:00 A.M. on the sixth day after the Odate of the job

Time From + 003 days Until + 006 days

In this example, the job is eligible for submission at any time between 1:00 A.M. on the third day after the Odate of the job and 1 A.M. at the end of the sixth working day after the Odate of the job.

The New Day time at the site is 8:00 A.M., and the Time synonym option in File/Options>Job Properties is checked. Submit the job at any time from one hour after the start of the New Day until one hour before the next New Day.

Time From 09:00 + days Until 31:00 + days

In this example, the job is eligible for submission between 9:00 A.M. until 7:00 A.M. the next morning. Because the Time Synonym option is checked, 7:00 A.M. the next morning is indicated as the sum of (24:00 + 7:00) = 31:00.

Must End

Sets the time and day when the job must finish executing. (z/OS only)

Additional information	Description
Usage	Optional
Format	Valid times expressed in 24-hour format (**hh:mm**). By default, valid times are from **00:00** through **23:59**.
Variable Name	None
Sub-parameters	
Time	**HH:MM**
+ num days	Optional sub-parameter, valid only for Control-M for z/OS that enables you to set a range of days after the original scheduling date of the job during which the job can be submitted. From + num days defines the beginning of the range and Until + num days defines the end of the range, where the variable num is a number of days after the Odate (original scheduling date) of the job. Valid values for num are any number from 1 through 120.

Ignore from time on next day

Enables the job to run the next day without time limitations.

Additional information	Description
Usage	Optional
Format	Check box
Variable Name	None
Alternate Names	▪ **Control-M/EM Utilities:** TIMETO ▪ **Reporting Facility**: TO TIME ▪ **Control-M/Server Utilities:** timeuntil ▪ **Control-M/EM API:** time_until

Time Zone

Indicates the time zone according to which the job should be scheduled.

Additional information	**Description**		
Usage	Optional		
Format	Drop-down list: Three–character value The following values are supplied with Control-M:		
	HNL	Honolulu	GMT-10.00
	HAW	Hawaii	GMT-10:00
		NOTE: HAW is maintained for backward compatibility purposes	
	ANC	Anchorage Standard Time	GMT-09:00
	PST	Pacific Standard Time	GMT-08:00
	MST	Mountain Standard Time	GMT-07:00
	CST	Central Standard Time	GMT-06:00

Additional information	Description		
	EST	Eastern Standard Time	GMT-05:00
	ATL	Atlantic Standard Time	GMT-04.00
	RIO	Rio de Janeiro	GMT-03.00
	GMT	Greenwich Mean Time	GMT+00:00
	WET	Western European Time	GMT+01:00
	CET	Central European Time	GMT+02:00
	EET	Eastern European Time	GMT+03:00
	DXB	Abu Dhabi	GMT+04.00
	KHI	Karachi	GMT+05.00
	DAC	Dacca	GMT+06.00
	BKK	Bangkok	GMT+07.00
	HKG	Hong Kong	GMT+08:00
	TYO	Tokyo	GMT+09.00
	TOK	Tokyo	GMT+09:00
		NOTE: TOK is maintained for backward compatibility purposes	
	SYD	Sydney	GMT+10.00
	MEL	Melbourne	GMT+10:00
	NOU	Noumea	GMT+11.00
	AKL	Auckland	GMT+12.00
	If necessary, these default values can be modified, and new values can be defined. For more information, see the description of time zone support in System configuration of *Control-M Workload Automation Administration*.		

Additional information	Description
Default	If no value is specified for this parameter, the job runs with the time zone of the Control-M/Server that ordered the job.
Invalid Characters	Non-English characters
Alternate names	▪ **Control-M/EM Utilities:** TIMEZOME ▪ **Reporting Facility:** (none) ▪ **Control-M/Server Utilities:** -timezone ▪ **Control-M for z/OS:** TIMEZONE ▪ **Control-M/EM API:** time_zone

Consider the following:

- Newly defined jobs with specified time zones must be saved at least 48 hours before their intended execution dates (in order to ensure that they are ordered automatically by the appropriate New Day Procedure or User Daily).

  If they must run "today" they should be ordered manually (for example, using the ctmorder utility).

- Specified Odates are calculated according to the working date (not the actual date). This means that if a job is defined as working on the 5th of the month at 3:00 A.M., and the working day begins at 5:00 A.M., then the job is actually run at 3:00 A.M. on the morning of the 6th (which is still part of the working day of the 5th).

- In addition to time zones, you can also order a job that is intended for execution on a future date. For more information, see the odate and odate_option parameters in any of the following Control-M/Server utilities:

  - ctmudly
  - ctmudchk
  - ctmorder
  - ctmcreate

For more information about these utilities, see *Control-M Utilities*.

BMC Software recommends that you do not combine jobs that have time zone specifications with jobs that do not specify a time zone in the same folder.

When a job is considered for ordering by the New Day procedure, it is ordered if its scheduling date occurs within the next 48 hours. When a job is ordered by a User Daily job, it is ordered only if its scheduling criteria are satisfied for the current working date. For this reason, BMC Software recommends that you arrange the jobs for each time zone in a separate folder. For more information, see the description of time zone support in System configuration of *Control-M Workload Automation Administration*.

Prerequisites parameters

Prerequisite parameters are related to the prerequisites of the job and the processes that accompany prerequisites

Parameter	Description
Requires user confirmation (on page 122)	Specifies whether user confirmation is required before the job is submitted for execution.
Control Resources (on page 123)	(From Forecast only) Indicates the resources required by the job during execution and the type of control (shared or exclusive) the job requires over each resource. The **Control Resources** parameter is used to control parallel execution of jobs.
In Condition (on page 125)	(From Forecast only) Specifies prerequisite conditions that must be satisfied before the job is submitted for execution. The **In Conditions** parameter makes the submission of the job dependent on the existence of one or more prerequisite conditions.
Quantitative Resources (on page 133)	(From Forecast only) Indicates the name and quantity of Quantitative resources required by the job.
Condition Name (on page 130)	Name of the In/Out Condition.
Date (on page 130)	The date of the condition.
Condition Relationships (on page 132)	Specifies the logical relationship between In Conditions.
Condition Brackets (on page 132)	Parentheses are used to isolate groups of In Conditions in a longer list of conditions to indicate logical relationships that are resolved before the entire condition statement is resolved. For more information, see In Condition (on page 125).
Adjust Condition (on page 136)	Determines whether to ignore prerequisite conditions normally set by predecessor jobs if the relevant predecessor jobs are not scheduled.

Requires user confirmation

Specifies whether user confirmation is required before the job is submitted for execution.

Additional information	Description
Usage	Optional
Format	Select the **Requires user confirmation** check box to indicate that confirmation is required. The job's submission criteria are not evaluated until Control-M receives manual confirmation from the user (using the Confirm/Restart option in the Control-M window). Clear the check box to indicate that confirmation is not required. The job's submission criteria are evaluated when the job is placed in the Active Jobs database, without the need for user confirmation.
Invalid Characters	Non-English characters
Variable Name	None
Alternate names	▪ **Control-M/EM Utilities:** CONFIRM ▪ **Reporting Facility:** (none) ▪ **Control-M/Server Utilities:** -confirm ▪ **Control-M for z/OS:** CONFIRM ▪ **Control-M/EM API:** confirm_flag
Alternate formats	
Control-M/EM Utilities	Valid values: ▪ **0** - No confirmation required. Default ▪ **1** - Confirmation required. **EXAMPLE:** <CONFIRM="1"/>
Control-M/Server Utilities	Valid values: ▪ **N** - No confirmation required. Default ▪ **Y** - Confirmation required.
Control-M for z/OS	Valid values: ▪ **N** - No confirmation required. Default ▪ **Y** - Confirmation required.

If **Requires user confirmation** is selected, a status of WAIT USER is assigned to the job when it is placed in the Active Jobs database. When you confirm the job, it is submitted after the remaining submission criteria are satisfied.

Control Resources

(From Forecast only) Indicates the resources required by the job during execution and the type of control (shared or exclusive) the job requires over each resource. The **Control Resources** parameter is used to control parallel execution of jobs.

Additional information	Description	
Usage	Optional **NOTE:** A maximum of 99 Control Resources can be specified for a job.	
Length	1-64 characters z/OS: 1-20 characters	
Case Sensitive	Yes However, if the Uppercase Only check box was selected in the Add Control-M Definition window, you cannot use lowercase characters.	
Invalid Characters	▪ Blanks ▪ Computers other than z/OS: Single quotation marks	
Variable Support	None	
Alternate names	▪ **Control-M/EM Utilities:** CONTROL ▪ **Reporting Facility:** RESOURCE ▪ **Control-M/Server Utilities:** -control ▪ **Control-M for z/OS:** CONTROL ▪ **Control-M/EM API:** control_resource	
Alternate formats		
Reporting Facility	The **Control Resources** parameter is composed of the sub-parameters below.	
	RESOURCE	Name of the resource. String.

Additional information	Description	
	TYPE	Type of Control resource: ▪ **E** - Exclusively owned by a job ▪ **S** - Shared by any number of jobs
Sub-parameters		
Resource Name	Defines the name of the Control resource.	
	Length	From 1 through 64 characters
	Case Sensitive	Yes However, if the Uppercase Only check box was selected in the Add Control-M Definition window, you cannot use lowercase characters.
	Invalid Characters	Blanks; single quotation marks.
Type	▪ Shared ▪ Exclusive	
On Fail (z/OS)	▪ Release ▪ Keep	

If a job requires exclusive use of a Control resource, only that job can use the Control resource. If another job requests the same resource in exclusive or shared state, this second job is not submitted by Control-M until the resource is released by the first job.

If a job requires a Control resource in shared state, that job can run in parallel with other jobs that request the same resource in shared state.

EXAMPLE: Scheduling a job using an exclusive resource

Job R1 (which reorganizes a disk) cannot run in parallel with job B1 (which backs up the same disk) or job B2 (which reads data from the disk). Jobs B1 and B2 do not interfere with each other and can be executed in parallel.

Job R1 is defined as follows:

File name = R1

- Control Resources: VS01 = E

Job B1 is defined as follows:

- File name: B1
- Control Resources: VS01 = S

Job B2 is defined as follows:

- File name: B2
- Control Resources = VS01 = S

Jobs B1 and B2 can run simultaneously since they both use resource VS01 in shared mode. Job R1 requests VS01 for exclusive use and therefore cannot run in parallel with either job B1 or B2.

If either job B1 or B2 is running, Control-M does not submit job R1.

If job R1 is running Control-M does not submit job B1 or B2.

Control-M submits job B1 if job B2 is running, and vice-versa.

In Condition

(From Forecast only) Specifies prerequisite conditions that must be satisfied before the job is submitted for execution. The **In Conditions** parameter makes the submission of the job dependent on the existence of one or more prerequisite conditions.

NOTE: A maximum of 99 prerequisite conditions can be specified for the In Condition parameter.

Additional information	Description
Usage	Optional
Format	The **In Condition** parameter is composed of the Condition Name (on page 130), Date (on page 130), and Condition Relationships (on page 132) sub-parameters.
Variable Name	Yes. Variable system variables (but not other types of variables) can be specified as the entire value for this parameter.
Alternate names	▪ **Control-M/EM Utilities:** INCOND ▪ **Reporting Facility:** CONDITION ▪ **Control-M/Server Utilities:** -incond ▪ **Control-M for z/OS:** IN ▪ **Control-M/EM API:** in_condition
Alternate formats	
Control-M/EM Utilities	INCOND is composed of the NAME. ODATE, AND_OR, and OP sub-parameters. The OP value is a right or left parenthesis. Used to isolate portions of the In condition expression.
Control-M/EM Utilities continued	<INCOND NAME="Cond1" ODATE="ODAT" AND_OR="AND" OP=")">

Additional information	Description
Reporting Facility	Condition name
Control-M for z/OS	Each specified prerequisite condition consists of the mandatory cond_name and dateref sub-parameters.

Logical relationships between conditions

The logical relationship (And/Or) used when two or more prerequisite conditions are specified determines whether all or only some of the prerequisite conditions must exist in order for the job to be submitted.

- Expressions in parentheses are resolved first.
- Pairs of parentheses cannot be nested.
- AND operations are resolved prior to the resolution of OR operations.
- AND conditions do not need to be grouped together before OR conditions.

Time stamp

NOTE: Not relevant in z/OS environments.

Inserting the **@HHMMSS** code into an In condition name includes a time stamp (in hours, minutes, and seconds) that is resolved to the time that the job is entered in the Active Jobs database.

Future and past dates

You can use the Days Offset field to set a future or past date for an In condition. Valid values for this field are a + (plus) or - (minus) sign, followed by a number from 0 through 999. The value in the field represents the number of days in the future, or in the past, relative to the actual order date.

EXAMPLE: Schedule a job that depends on the successful completion of another job

Schedule JOB_B to execute after the successful completion of JOB_A (regardless on what day JOB_A was run):

- Job Name JOB_A
- Out Conditions JOB_A_OK Date ODAT
- Job Name JOB_B
- In Conditions JOB_A_OK Date ****

If JOB_A completes with a termination status of OK, prerequisite condition JOB_A_OK is created (with JOB_A's original scheduling date). JOB_B, that is waiting for prerequisite condition JOB_A_OK, is submitted for execution as soon as the prerequisite condition is created.

Daily jobs

JOB_A and JOB_B are daily jobs. JOB_B should be submitted only after the successful completion of JOB_A. JOB_B must not be mistakenly submitted based on the successful completion of JOB_A from a previous day.

- Job Name JOB_A
- Out Conditions JOB_A_OK Date ODAT

- Job Name JOB_B
- In Conditions JOB_A_OK Date ODAT

Assuming that the scheduling date is May 5:

If JOB_A completes with a termination status of OK, prerequisite condition JOB_A_OK is created (with the date 0505). JOB_B, that is waiting for prerequisite condition JOB_A_OK with date 0505, is submitted for execution as soon as the prerequisite condition is created.

The In Conditions parameter of JOB_B is not satisfied by the JOB_A_OK condition with a date of 0504 (created the previous day).

Schedule a job that depends on the completion of multiple jobs

Schedule the job that produces the salary statistics report for top management after the set of jobs that calculates the salaries ends OK:

- File Name EBDRPT1A
- Days 01,15
- In Conditions SALARY-OK

When the set of jobs that calculates the salaries ends OK, it creates prerequisite condition SALARY-OK.

The report is produced twice a month, for the 1st and for the 15th. The report for the 15th is produced only if its prerequisite condition SALARY-OK exists (signifying that the 15th's salary job ended OK). The existence of the prerequisite condition for the 1st – SALARY-OK does not enable the submission of the report for the 15th.

The report for the 1st does not necessarily run on the 1st of the month. Suppose the salary jobs only finish executing on the 3rd; only then is the prerequisite condition SALARY-OK for the 1st created. This is because the prerequisite condition is always associated with a scheduling date, and not with the actual running date. Therefore, a date reference should be added to the condition:

- File Name EBDRPT1A
- Days 01,15
- In Conditions SALARY-OK Date ODAT
- Job scheduling that depends on previous job runs and a generic date reference

Consider a similar example: a monthly total report must be produced based on data from the last two runs. Also, the job must run when communication channels to a remote site are active:

- Days 01,15
- Retro Y
- Keep Active for 06
- In Conditions SALARY-OK Date: ODAT And/Or: A
- SALARY-OK Date: PREV And/Or: A
- COMM-ACTIVE Date: **** And/Or: A

The job is submitted only if the jobs for the 1st and the 15th have finished. Prerequisite condition COMM-ACTIVE is based on a "generic" date reference that exists only when communication is active.

The communication process itself can be monitored by Control-M. When communication is not active, Control-M deletes prerequisite condition COMM-ACTIVE, preventing abends of jobs that depend on active communication.

Maintain order of job runs

Assume a group of jobs runs every day of the week, except Saturday and Sunday. It is very important that some of the jobs scheduled for the various days of the week do not run in parallel. The order of these jobs must be maintained even in cases of delays:

- Week Days 2,3,4,5,6
- Retro Y
- Keep Active for 8
- In Conditions DEPOSITS Date: PREV

The job is submitted only if the prerequisite condition DEPOSITS of the previous scheduling date exists. The prerequisite condition DEPOSITS is created only after the group of jobs called DEPOSITS finishes.

Job scheduling that depends on site conditions

Suppose a Database master data set exists, and that it is accessed by many programs. Unfortunately, the contents of the Database are often destroyed or damaged because of bugs in old programs. When, and if, it is discovered that the contents of the Database are corrupted, submission of all the jobs that access the file must be prevented:

- Dates 0201, 0403, 1101
- Retro Y
- Keep Active for 06
- In Conditions MASTER-FILE-OK Date: ****

It has been discovered that the Database contents are indeed "bad." At this point, prerequisite condition MASTER-FILE-OK must be deleted, either manually from Control-M/EM, or automatically by the job or process that made the discovery. When the Database is restored to its normal state, the prerequisite condition is added again and execution proceeds as planned.

Job scheduling when a failure has occurred

A detached process, LOGCLOSE, is to begin whenever a certain communication process crashes because of a boot operation, or as a result of an operator's cancelation:

- Retro Y
- Keep Active for 0
- In Conditions COMM-CANCELED Date: ****

The detached process LOGCLOSE starts executing whenever prerequisite condition COMM-CANCELED is active. The prerequisite condition is added by Control-M when the communication process is canceled by an operator or after a computer crash.

Date reference

The following example provides a further explanation of the concept of the scheduling date reference:

- File Name EBDRPT6D
- Days 01,15,20
- Months 1-N 2-N 3-N 4-N 5-N 6-N 7-Y 8-N 9-Y 10-N 11-N 12-N
- In Conditions EBD-REPORTS-READY Date: ****

Today is the 15th of September. The date reference values resolved in this job are written in mmdd date format:

- ODAT 0915
- PREV 0901
- **** Any date reference

Condition including terms isolated with parentheses

ctmcreate -tasktype command -cmdline ls -jobname cond_600 -incond cond_21 ODAT AND -incond '('cond_22 ODAT OR -incond cond_23')' ODAT AND -incond cond_24 ODAT OR

Out condition name including a time stamp

ctmdefine -tasktype command -cmdline date -application app -sub_application grp -folder time_cond -days ALL -month ALL Y -outcond a@HHMMSS ODAT ADD -jobname counter -memname counter

The resulting Out condition statement:

*CONDITION a101028 0113 ADDED

Today is the 15th of September. The date reference values resolved in this job are written in mmdd date format:

- ODAT 0915
- PREV 0901
- **** Any date reference

Condition Name

Name of the In/Out Condition.

Additional information	Description
Usage	Mandatory
Length	▪ Condition names for Control-M versions prior to 6.0.01 can be from 1 through 20 characters. ▪ Condition names for Control-M version 6.0.0*x* can be from 1 through 39 characters. ▪ Condition names for Control-M version 6.1.0.*x* can be from 1 through 255 characters. (z/OS: Not more than 39 characters)
Case sensitive	Yes However, if the **Uppercase Only** check box was selected in the **Add Control-M Definition** window, you cannot use lowercase characters.
Invalid Characters	Computers other than z/OS: Blanks; single quotation marks; ")" and "(" (parentheses); "|" (pipe) z/OS: Blanks; ")" and "(" (parentheses); "|" (pipe)
Variable Name	None

Date

The date of the condition.

Additional information	Description	
Usage	Mandatory	
Format	Drop-down list	
Options	**Any Date**	Any scheduling date. The Condition parameter is satisfied if any prerequisite condition with the same name exists, regardless of its associated date.

Additional information	Description	
	No Date	Condition is not date-dependent. This value is valid only for Control-M/Server version 6.0.01 and later or for Control-M for z/OS.
	Order Date	Variable that is automatically replaced by the job's original scheduling date (that is, the date on which the job was ordered). Beginning with version 6.1.03 of Control-M/EM and Control-M/Server, ODAT can also be specified when modifying details of a job in the Active Jobs database.
	Previous Order Date	Variable that is automatically replaced by the job's previous scheduling date. **PREV** cannot be specified when modifying details of a job in the Active Jobs database.
	Next Order Date	Variable that is automatically replaced by the job's next scheduling date. **NEXT** cannot be specified when modifying details of a job in the Active Jobs database.
	Specific Date	A 4-digit date reference in the **mmdd** or **ddmm** format, depending on the site standard.
	Offset	+ or - followed by a number from 0 through 999, indicating the number of days in the future (+), or in the past (-), relative to the actual order date.

Condition Relationships

Specifies the logical relationship between In Conditions.

Additional information	Description	
Usage	Mandatory	
Format	Drop-down list	
Options	**AND Between All**	Specifies that all In Conditions have an AND relationship
	OR Between All	Specifies that all In Conditions have an OR relationship
	Set Manually	Enables you to set the Condition Relationship criteria manually

Condition Brackets

Parentheses are used to isolate groups of In Conditions in a longer list of conditions to indicate logical relationships that are resolved before the entire condition statement is resolved. For more information, see In Condition (on page 125).

NOTE: Pairs of parentheses cannot be nested.

Additional information	Description	
Usage	Optional	
Format	Opening Bracket drop-down list Closing Bracket drop-down list	
Options	*blank*	Indicates no special relationship between the current and following condition.
	(	Beginning of a comparison between successive conditions.

Additional information	Description	
	)	End of a comparison between successive conditions.

Quantitative Resources

(From Forecast only) Indicates the name and quantity of Quantitative resources required by the job.

Additional information	Description
Usage	Optional.
Format	Each Quantitative resource is specified using the mandatory sub-parameters described below.
Case Sensitive	Yes. However, if the Uppercase Only check box was selected in the Add Control-M Definition window, you cannot use lowercase characters.
Invalid Characters	▪ Blanks ▪ Computers other than z/OS: Single quotation marks
Variable Name	Yes. Variable system variables (but not other types of variables) can be specified as the entire value for this parameter.
Alternate names	▪ **Control-M/EM Utilities:** QUANTITATIVE ▪ **Reporting Facility:** RESOURCE ▪ **Control-M/Server Utilities:** -quantitative ▪ **Control-M for z/OS:** RESOURCE ▪ **Control-M/EM API:** quantitative_resource
Sub-parameters	

<table>
<tr><th>Additional information</th><th colspan="2">Description</th></tr>
<tr><td>Name</td><td colspan="2">Name of the Quantitative resource.

The following special suffixes can be appended to the specified resource name:

EXAMPLE: $ represents any single character.

TAPE$ can represent TAPE1 or TAPE2.

However, if the job requires two TAPE$ units, it can only use two TAPE1 units or two TAPE2 units, not one of each. (The $ can only assume value 1 or 2 for the job; it cannot represent both.)

▪ @ identifies a Quantitative resource used for load balancing.</td></tr>
<tr><td></td><td>Length</td><td>From 1 through 64 characters

z/OS: 1 through 20 characters</td></tr>
<tr><td></td><td>Case Sensitive</td><td>Yes.

However, if the Uppercase Only check box was selected in the Add Control-M Definition window, you cannot use lowercase characters.</td></tr>
<tr><td></td><td>Invalid Characters</td><td>Blanks; single quotation marks.</td></tr>
<tr><td>Quantity</td><td colspan="2">Amount of the resource that is required. Valid values for this field are from 1 through 9999.</td></tr>
</table>

When a Quantitative resource is specified for a job, Control-M determines whether a sufficient quantity of the specified resource is available before submitting the job. When the job is submitted, the specified quantity of resource is assigned to that job and is not available to other jobs. When the job finishes executing, the resource is made available to other jobs.

NOTE: A maximum of 99 Quantitative resources can be specified for a job.

The **Quantitative Resources** parameter is used to control the use of Quantitative resources in the installation (for example, tape drives, CPU utilization).

For load balancing, Quantitative resources are used to specify the resources that must be available on the agent computer selected by Control-M to execute the job.

Load balancing is available for certain computers with Control-M version 2.2x or later.

EXAMPLE: Specifying two tape drives

A Control-M installation has 10 tape drives available for production. A job that requires the use of two tape drives is defined with a Quantitative Resources parameter specifying the number of tape drives required:

Quantitative Resources TAPE 2

Control-M determines whether two tape drives are available. If the drives are available, and all other submission criteria for the job have been satisfied, the tape drives are allocated to the job, and the job is submitted for execution. The total number of free tape drives is now eight. When the job finishes executing, the two tape drives are returned to the pool of available resources.

EXAMPLE: Quantitative resource requirement that cannot be fulfilled

Given the following situation:

- Several jobs that require tape drives are currently executing.
- Only one tape drive is currently available.

Job A requires two tape drives and contains the following parameter (all other submission criteria for the job have been satisfied):

Quantitative Resources TAPE 2

As long as two tape drives are not available, Job A is not submitted for execution.

If another tape drive is released by a different job, or if an authorized user increases the number of existing tape drives so that two are available, Control-M submits the job for execution.

EXAMPLE: Defining a quantitative resource with a mask character

A Control-M installation is defined as having the following Quantitative resources: three units of TAPE1 and three units of TAPE2. A job requiring three tape drives contains the following parameter:

Quantitative Resources TAPE$ 3

The job is submitted for execution when three units of the same type are available (that is, either three units of TAPE1 or three units of TAPE2) since the mask character $ can only represent a single value for a given job.

EXAMPLE: Defining multiple quantitative resources with mask characters

A job requires two tape drives and a printer:

Quantitative Resources TAPE$ 2 PRINT$ 1

Two units of TAPE1 and one unit of PRINT2 are available. However, the job is not submitted until a unit of PRINT1 becomes available since the mask character $ can only represent a single value for a given job. In this instance, $ represents the number 1.

EXAMPLE: Load balancing

A job to be submitted by the load-balancing mechanism requires 10 units of the Quantitative resource CPU:

Quantitative Resources CPU@ 10

The job is submitted to an agent computer possessing at least 10 available units of the specified resource.

Adjust Condition

Determines whether to ignore prerequisite conditions normally set by predecessor jobs if the relevant predecessor jobs are not scheduled.

This parameter is relevant only for SMART Folders.

Additional information	Description
Usage	Optional
Format	Valid values are: ▪ Yes – Ignore relevant prerequisite conditions. ▪ No – Do not ignore relevant prerequisite conditions. ▪ Dummy ▪ [Control-M for z/OS as of version 6.2.*xx* only]. Order as a PSEUDO job any job with scheduling criteria that are not satisfied on the current ODATE, with the File Path/Member library (on page 25) parameter of the job set to DUMMY. For more information, see ADJUST CONDITIONS in the job production parameters chapter in the *Control-M for z/OS User Guide*. ▪ [Control-M/Server] The CTM_GROUP_ADJUST_DUMMY configuration parameter controls the creation of dummy jobs that run in place of unscheduled prerequisite jobs. This parameter is relevant only when Adjust Condition is set to **Y**. **NOTE:** The Control-M/Server CTM_ADJUST_COND_SCOPE configuration parameter controls whether jobs in the SMART Folder must ignore conditions set by jobs in the active jobs database that are not scheduled, or to ignore conditions set by jobs in the SMART Folder that are not scheduled. In the latter case, the missing conditions are removed from the job at order time. This parameter is relevant only when Adjust Condition is set to **Y**.
Invalid Characters	Non-English characters
Variable Name	None

Additional information	Description
Alternate names	▪ **Control-M/EM Utilities:** ADJUST_COND ▪ **Reporting Facility:** ADJUST CONDITION ▪ **Control-M/Server Utilities:** -adjust_cond ▪ **Control-M for z/OS:** Adjust Condition ▪ **Control-M/EM API:** adjust_condition
Alternate formats in other Control-M components	
Control-M/EM Utilities	String. **Example:** <ADJUST_COND="Y"/>
Control-M/Server Utilities	▪ **Y** – ignore the relevant prerequisite conditions ▪ **N** – utilize prerequisite conditions normally set by predecessor jobs. Default.
Control-M for z/OS	▪ Blank – do not consider the parameter. Default. ▪ **N** – ignore the relevant prerequisite conditions ▪ **Y** – utilize prerequisite conditions normally set by predecessor jobs

The Adjust Condition parameter is applied to all jobs in a folder. It defines job dependencies in the folder as being either conditional or absolute.

Conditional and absolute dependencies

- **Absolute job dependency**
  Each job in a folder runs only after all the predecessor jobs in the folder have run. All prerequisites In Condition (on page 125) for a job must exist in the Conditions/Resources table before the job can run.

  Select **No** in the **Adjust Condition** list in the Control-M Properties pane.

- **Conditional job dependency**
  Each job in the folder waits for its predecessor jobs to run only if the predecessor jobs in the folder are scheduled. Prerequisite condition requirements are ignored and the successor job runs if a predecessor job is not scheduled. The other runtime scheduling criteria for the job must be satisfied.

  Select **Yes** in the **Adjust Condition** list in the Control-M Properties pane.

Actions parameters

Actions parameters are used to describe:

- Actions to be performed if the job is not submitted.
- Actions to be performed after the job has finished executing.
- Handling of the job's log (Output).
- Notification messages to various users.
- Rerun conditions.

Based on the job's execution status codes, Control-M automatically assigns a completion status:

Status	Description
OK	Job ended **OK**. The job finished executing with a successful operating system completion status. This status can be overridden with the End Job Ok (on page 161) parameter (as a result of an On Statement/Code evaluation).
NOTOK	Job did not end **OK**. The job finished executing with an unsuccessful operating system completion status or due to a submission failure (for example, queue does not exist). This status can be overridden with the End Job Ok (on page 161) parameter (as a result of an On Statement/Code evaluation).

Steps parameters can also be defined for SMART folders.

- Actions to be performed if a SMART folders finished **OK** are performed only if all the jobs in the SMART folder finished **OK**.
- Actions to be performed if a SMART folders finished **NOTOK** are performed only if one or more jobs in the SMART folders finished **NOTOK**.

Post-processing of a SMART folder occurs when the last job in the SMART folder ends. A SMART folder may become active again after the post-processing has been completed, if one or more jobs are rerun, or if a new job is added to the SMART folder and submitted. In this case, when the last job in the SMART folder ends again, the completion status of the SMART folder is rechecked and the appropriate post-processing actions are performed.

For more information about SMART folders, see Job definition in *Control-M Workload Automation*.

Parameter	Description
Auto-Archive (on page 139)	Determines whether SYSDATA (job output) must be archived.
CTB Step (on page 142)	Adds Control-M/Analyzer steps as the first and/or last step of the job's execution.
On/Do Actions (on page 146)	Automatic interventions, such as rerunning a job or ordering a remedial job, if a job ends with a particular error code.
Code values (on page 175)	Code values can be condition codes, user abend codes, system abend codes, various end codes and statuses, and certain keywords. They can also be prefaced by certain qualifiers.
Out Condition (on page 200)	(From Forecast only) Specifies prerequisite conditions to be added or deleted after the job completes with a completion status of **OK**.
Action (on page 202)	Indicates how the job's log file (Output) should be handled after the job ends with a status of **OK**.
Retention Days (on page 210)	Determines the number of days to retain the job in the History Jobs file. For z/OS jobs only.
Retention Generations (on page 211)	Maximum number of generations of the job to keep in the History Jobs file. For z/OS jobs, only.
Specific Range Name (on page 173)	Specifies a range of steps in the steps of an On PGMST statement.
Pre-Notifications (on page 212)	The Pre-Notifications parameter gives you the ability to create a message that is sent to one or more specified destinations when certain conditions are encountered.
Final Notifications (on page 221)	Sends a notification if a job terminates with a completion status of OK or NOTOK.

Auto-Archive

Determines whether SYSDATA (job output) must be archived.

NOTE: The parameter is displayed only if Control-R is selected.

Additional information	Description
Usage	Optional
Format	Check box. When this check box is selected, job output is archived. When this check box is cleared, job output is not archived. In this case, Control-R cannot restart the job, and SYSDATA viewing under Control-M for z/OS is not possible. When the Auto-Archive check box is selected, the sub-parameters described below can also be specified.
Variable Support	None
Alternate names	▪ **Control-M/EM Utilities:** AUTOARCH ▪ **Reporting Facility:** AUTO ARCHIVE ▪ **Control-M/Server Utilities:** (none) ▪ **Control-M for z/OS Utilities:** AUTOARC ▪ **Control-M for z/OS:** AUTO-ARCHIVE ▪ **Control-M/EM API:** ▪ auto_archive ▪ sys_db ▪ arch_max_days ▪ arch_max_runs
Computer specific information	The Auto-Archive parameter is relevant only for jobs to be run in z/OS environments or OpenVMS computers.
Sub-parameters	
SYS DB	A check box If this check box is selected, a single data set is used for archiving the SYSDATA of all jobs until it is full. When the first archive data set is full, another SYSDATA data set is allocated and used. BMC Software recommends this method. If this check box is cleared, a separate data set is created for the SYSDATA of each job run.

Additional information	Description
Max Days...	A number from 00 through 99. Indicates the maximum number of days to retain the SYSDATA archive data set for jobs that ended **NOTOK**.
Max Runs...	Indicates the maximum number of job runs to retains the SYSDATA archive data set for jobs that ended **NOTOK**. Valid values: ▪ a number from 00 through 99 ▪ blank (there is no maximum number of job runs)

The **Auto-Archive** parameter enables you to decide whether to archive job output (SYSDATA). SYSDATA refers to all information in the job log, the expanded JCL (job script), and to the output messages of the job.

While archiving SYSDATA is normally desirable, it might not be desirable for cyclic jobs, started tasks, or frequently repeated jobs that do not require restart.

BMC Software recommends that you select the SYSDB check box. BMC Software does not recommend creating a separate data set for each run because:

- Creating many data sets consumes a large amount of space in the disk VTOC.
- Each data set is allocated on a track basis. If the SYSDATA does not completely fill the track, large amounts of disk space may be wasted.

When archiving SYSDATA, BMC Software recommends that value **99** not be specified for the Keep Active for (on page 79) parameter for cyclic jobs or started tasks. Otherwise, these jobs, which are never automatically deleted from the Active Jobs database, can cause the disk to fill up with unnecessary archived SYSDATA.

NOTE: Specified parameters take effect only during execution of the New Day procedure (CONTDAY) or the Control-M CTMCAJF utility. Therefore, it is possible to find more generations of the same job than the current value of Maximum reruns (on page 100).

Whenever a job is deleted from the Active Jobs database and does not reside in the History file, its SYSDATA is deleted regardless of Maximum reruns (on page 100) and Keep Active for (on page 79).

The Keep Active for (on page 79) and Maximum reruns (on page 100) parameters define retention criteria for the archived SYSDATA of jobs that ended **NOTOK**. Defaults for these parameters are defined using Control-M/Restart installation parameters. **Keep Active for** and **Maximum reruns** values in a job definition are used to override the Control-M/Restart defaults. If both parameters are specified, retention is limited by the parameter that is satisfied first.

CTB Step

Adds Control-M/Analyzer steps as the first and/or last step of the job's execution.

Additional information	Description	
Usage	Optional	
Availability	This parameter is accessed from the Control-M for z/OS interface only.	
Format	CTB Step is composed of the **AT**, **NAME**, **TYPE**, and **ARGUMENTS** sub-parameters, described below.	
Invalid Characters	Non-English characters	
Variable Support	None	
Alternate names	▪ **Control-M/EM Utilities:** (none) ▪ **Reporting Facility:** (none) ▪ **Control-M/Server Utilities:** (none) ▪ **Control-M for z/OS:** (none) ▪ **Control-M/EM API:** ctb_step	
Sub-parameters		
AT	Indicates where to place the Control-M/Analyzer step in the job. Mandatory. Valid values: ▪ **S (Start)** – The indicated Control-M/Analyzer step must become the first step of the job. ▪ **E** (End) – The indicated Control-M/Analyzer step must become the last step of the job.	
NAME	Name of the Control-M/Analyzer entity. Must be a valid name of a Control-M/Analyzer rule or mission. Mandatory.	
	Length	1 through 8 characters
	Case Sensitive	Yes
	Invalid Characters	Blanks; non-English characters

Additional information	Description	
TYPE	Type of Control-M/Analyzer entity. Mandatory. Valid values: ▪ **R** (Rule) – Entity is a Control-M/Analyzer rule. ▪ **M** (Mission) – Entity is a Control-M/Analyzer mission.	
ARGUMENTS	Arguments to be passed to the Control-M/Analyzer step. Optional.	
	Length	0 through 60 characters
	Case Sensitive	Yes
	Invalid Characters	Non-English characters

A maximum of two CTB STEP statements (that is, one START statement and one END statement) can be specified.

Multiple arguments must be separated by a comma without a space because they are automatically passed to the Control-M/Analyzer step as a PARM=<*arguments*> parameter in the step's JCL.

Control-M uses the status returned by Control-M/Analyzer as it would use the return status of any job step.

- If Control-M/Analyzer returns a status of **OK** or **TOLER** (within accepted tolerances), Control-M considers the step as having ended **OK**.
- If Control-M/Analyzer returns a status of **NOTOK** or **ABEND**, Control-M considers the job step as having ended **NOTOK**.

NOTE: The Auto-Archive parameter is relevant only for jobs to be run in z/OS environments or OpenVMS computers. The parameter is displayed only if Control-R is selected.

EXAMPLE: Check results and set a condition

After successfully performing salary calculations, job SACALC01 invokes rule CHKCALC to ensure that the results are reasonable, and then sets OUT condition SALARY-OK.

JOB: SACALC01 LIB CTM.PROD.SCHEDULE FOLDER: SALARY

COMMAND ===> SCROLL===> CRSR

+--+

MEMNAME SACALC01 MEMLIB GENERAL

RUN AS SYS1 TASKTYPE JOB PREVENT-NCT2 DFLT N

APPL SAL SUB APPLICATION SALARY

DESC SALARY CALCULATIONS

OVERRIDE PATH

SCHENV SYSTEM ID NJE NODE

```
SET VAR
CTB STEP AT END NAME CHKCALC TYPE RULE
ARGUMENTS %%ODATE
CTB STEP AT NAME TYPE
DOCMEM SACALC01 DOCLIB CTM.PROD.DOC
================================================================
==================
DAYS 01,15 DCAL
AND/OR
WDAYS WCAL
MONTHS 1- Y 2- Y 3- Y 4- Y 5- Y 6- Y 7- Y 8- Y 9- Y 10- Y 11- Y 12- Y
DATES
CONFCAL SHIFT RETRO Y KEEP ACTIVE FOR 00 D-CAT
MINIMUM PDS
DEFINITION ACTIVE FROM UNTIL
================================================================
==================
IN
CONTROL
RESOURCE
PIPE
TIME: FROM UNTIL PRIORITY DUE OUT SAC CONFIRM
TIME ZONE:
================================================================
==================
OUT SALARY-OK ODAT +
COMMANDS: EDIT, DOC, PLAN, JOBSTAT 11.17.00
```

Step range list

Specifies a range of steps in the steps of an On program step statement.

This parameter is only relevant for z/OS.

Additional information	Description
Usage	Optional
Format	Consists of the Name, From, and To sub-parameters described below.
Invalid Characters	None English characters.
Variable Support	None
Alternate names	▪ **Control-M/EM Utilities:** (none) ▪ **Reporting Facility:** (none) ▪ **Control-M/Server Utilities:** (none) ▪ **Control-M for z/OS:** (none) ▪ **Control-M/EM API:** ctb_step
Sub-parameters	
Name	Name for the range. 1through 7 characters. Only trailing blanks are allowed in this field.

From	First program step or program step, procedure step in the range.
	Program step is the step name in the EXEC statement that identifies the program to be executed:
	//pgmstep EXEC PGM= pgmname
	Procedure step is the step name in the EXEC statement that invokes the procedure:
	// procstep EXEC procname
	Program step values and procedure step values can each be from 1 through 8 characters in length, and must not contain blanks.
	Non-English characters not allowed.

To	Last program step or program step, procedure step in the range. The To sub-parameter is optional. If blank, its value defaults to the last step in the job. Non-English characters not allowed.

Whenever a Step Range statement is specified, it eliminates the need to define separate On program step, On procedure step and On Codes statements and accompanying Do actions for each step in the range. The defined Step Range Name can be used (without redefining the range) in subsequent On program step, On procedure step, and On Codes statements, by specifying the Step Range Name, preceded by an asterisk (*), in the On program step field.

Any number of step ranges can be specified. After entering a Step Range parameter, another Step Range parameter line is automatically displayed.

Using all runs of a job including restarts

When processing On blocks, Control-M can incorporate the results of all previous runs and restarts, filtering them for jobs restarted with the Control-M for z/OS RESTART, RECAPTURE CONDITION or ABEND CODES parameters. Control-M/Restart searches previous runs to determine which steps must be considered part of the restarted job.

For example, if one step finished successfully during its original run and another step finished successfully after a restart, the On block check for the successful finish for both steps produces a TRUE result and the On statement is satisfied.

Activation of this facility requires that the Control-M for z/OS ALLRUNS parameter in the CTRPARM member be set to YES. When activated, this facility may apply to any specified step, step range, or to the +EVERY step value.

On/Do Actions

Automatic interventions, such as rerunning a job or ordering a remedial job, if a job ends with a particular error code.

Control-M processes On/Do Actions combinations in the following order:

1. On/Do Action combinations related to completion codes, for example:

   ON "*" "COMPSTAT>0"

2. On Statement/Code combinations related to Output, for example:

   ON "*cp aaa bbb*" "*not found*"

3. On Statement/Code combinations based on the Job ended OK/not Ok state of the job, for example:

   ON "*" "NOTOK"

On Parameters

Parameter	Description
Job ended	Determines whether the accompanying DO statements are performed if the job ended.
Job ended OK	Determines whether the accompanying DO statements are performed if the job ended OK.
Job ended not OK	Determines whether the accompanying DO statements are performed if the job ended not OK.
OS completion status (on page 188)	Determines whether the accompanying DO statements are performed if the OS completion status is met. ▪ =, <, >, !, Even, Odd ▪ Completion code
Job's number of executions (on page 191)	Determines whether the accompanying DO statements are performed if the job's number of executions is met. ▪ =, <, >, !, Even, Odd ▪ Number of executions
Job's number of reruns (on page 189)	Determines whether the accompanying DO statements are performed if the job's number of reruns due to DO RERUN, is met. For more information, see Rerun Job (on page 163). ▪ =, <, >, !, Even, Odd ▪ Number of reruns
Output cannot be found	Determines whether the accompanying DO statements are performed if the Output cannot be found.
Job has not been submitted yet	Determines whether the accompanying DO statements are performed if the job is submitted to the Control-M/Agent to run, but returns as not submitted.
Job's number of failures (on page 192)	Determines whether the accompanying DO statements are performed if the job's number of failures is satisfied.

Parameter	Description
Specific statement output (on page 194)	The On parameters for a specific statement output requires the following: ▪ Statement: An asterisk must be specified in this field. The statement set is compared to the On statements defined for the job. If the Code parameter is specified as the * mask character, it can match even a null error message record successfully. If there is a match between the On statements and a statement set from the job's log, the corresponding Do action statements are executed. Otherwise, Control-M proceeds to the next statement set. ▪ Code: Defines the exit code of DOS .bat scripts and REXX .cmd scripts that are returned to Control-M/Server, as described in Utilization of exit codes (on page 195). On Statement/Code combinations related to completion codes, for example: ON "*" "COMPSTAT>0" On Statement/Code combinations related to Output, for example: ON "*cp aaa bbb*" "*not found*" On Statement/Code combinations based on the OK/NOTOK state of the job, for example: ON "*" "NOTOK"

Do Actions

Parameter	Description
Set Variable (on page 150)	The Set Variable assigns a value to a variable for use in a rerun of the job when the On Statement/Code criteria are satisfied.
Add/Remove Condition (on page 152)	The Add/Remove Condition parameter specifies a prerequisite condition to be added or deleted when the On Statement/Code criteria are satisfied.
Order Job (on page 154)	The Order Job parameter forces an individual job or all jobs in a folder to be placed in the Active Jobs file (regardless of each job's Scheduling criteria) when the On Statement/Code criteria are satisfied.
Notify (on page 156)	The Notify parameter specifies a notification to be sent a specific destination when the specified On Statement/Code criteria are satisfied.

Parameter	Description
End Job Not Ok (on page 160)	The End Job Not OK parameter assigns a completion status of **NOTOK** to a job when the On Statement/Code criteria are satisfied, regardless of the job's actual completion status.
End Job Ok (on page 161)	The End Job Ok parameter assigns the completion status of **OK** to a job, regardless of its actual completion status.
Remedy (on page 162)	Opens a ticket in the Remedy Help Desk.
Rerun Job (on page 163)	Indicates if an automatic rerun should be performed when the On Statement/Code criteria are satisfied.
Control-M Analyzer Rule (on page 180)	Invokes a Control-M/Analyzer rule to be executed.
Mail (on page 157)	A mail message that is sent to the indicated e-mail addresses when the condition specified by the On Statement/Code parameter is satisfied.
Set Restart Options (on page 183)	Job steps to be executed during restart of a job.
Stop Cyclic Run (on page 164)	When specified, this parameter prevents subsequent iterations of the current cyclic job.
Handle Output (on page 164)	Indicates how the job's output should be handled when the On Statement and Code criteria are satisfied.

Set Variable

The Set Variable assigns a value to a variable for use in a rerun of the job when the On Statement/Code criteria are satisfied.

Additional information	Description
Usage	Optional
Format	**Name** and **Value** text boxes Enter the variable name in the **Name** text box, and the expression in the **Value** text box. Application-specific job parameters may not be specified in variable values. The names of application-specific job parameters are prefixed by two percent signs, the application's abbreviation and a hyphen (%%SAPR3- for SAP, %%OAP- for Oracle, and so on).
Length	Computers other than z/OS: The variable name can be up to 40 characters long (including the **% %** prefix). The Variable expression Value can be up to 214 characters long. z/OS: The total length of **Set Variable**, including the %% prefix, variable-name, value, and the = sign, must not exceed 55 characters.
Case Sensitive	Yes
Invalid Characters	Computers other than z/OS: The following characters cannot be included as part of variable-name in a User-defined variable: < > [ ] { } ( ) = ; ‘ ~ | : ? . + - * / & ^ # @ ! , " '. In value the only invalid characters are embedded blanks. Leading blanks are valid. z/OS: Blanks
Variable Support	Yes. A variable or expression can be specified as all or part of the value for this parameter.

Additional information	Description	
Alternate names	▪ **Control-M/EM Utilities:** DOVARIABLE ▪ **Reporting Facility:** (none) ▪ **Control-M/Server Utilities:** -dovariable ▪ **Control-M for z/OS:** (none) ▪ **Control-M/EM API:** do_variable	
Alternate formats		
Control-M/EM Utilities	The value for the DOVARIABLE parameter is contained in the EXP sub-parameter. **EXAMPLE:** <DOVARIABLE EXP="%%PARM1=%%TIME"/>	
	EXP	String containing both the variable and the expression, presented as an equation.
Sub-parameters		
Name	Name of the variable. The **%%** prefix is entered automatically. **1**-**38** character string (following the **%%** prefix).	
Type	From the Variable Type drop-down list, select a variable type as described in Adding a variable.	
Value	Variable expression. **1**-**214** character string. Application-specific job parameters may not be specified in variable values. The names of application-specific job parameters are prefixed by two percent signs, the application's abbreviation and a hyphen (%%SAPR3- for SAP, %%OAP- for Oracle, and so on).	

Variables are resolved (replaced) at the time a job is submitted.

The **Set Variable** parameter can be used to assign a value to a variable for use in a rerun of the job. When a job is rerun, statements specified in Set Variable are evaluated after statements specified in Variable Assignment.

Unless it is assigned a value in the Variable Assignment parameter, a variable that is assigned a value in a **Set Variable** parameter does not have any value during the first submission of the job.

Add/Remove Condition

The Add/Remove Condition parameter specifies a prerequisite condition to be added or deleted when the On Statement/Code criteria are satisfied.

Additional information	Description	
Usage	Optional	
Format	Each **Add/Remove Condition** parameter consists of three sub-parameters, described below.	
Variable Support	None	
Alternate names	▪ **Control-M/EM Utilities:** DOCOND ▪ **Reporting Facility:** (none) ▪ **Control-M/Server Utilities:** -docond ▪ **Control-M for z/OS:** DO COND ▪ **Control-M/EM API:** do_cond	
Alternate formats		
Control-M/EM Utilities	DOCOND contains the following sub-parameters:	
	NAME	Name of the condition. String.
	ODATE	Date (four characters long) for the prerequisite condition. String. Default: **ODAT**.
	SIGN	Valid values: ▪ **ADD** (Default) ▪ **DEL**
Sub-parameters		
Name	Descriptive name of a prerequisite condition.	
	Length	▪ 1 through 255 characters ▪ z/OS: 1 through 39 characters ▪ Control-M for z/OS versions prior to 6.0.00: 1 through 20 characters

Additional information	Description	
	Case Sensitive	Yes. However, if the Uppercase Only check box was selected in the Add Control-M Definition window, you cannot use lowercase characters.
	Invalid Characters	Computers other than z/OS: Blanks; single quotation marks; ")" and "(" (parentheses); "|" (pipe) z/OS: Blanks; ")" and "(" (parentheses); "|" (pipe)
Date	Date (four characters long) for the prerequisite condition. ▪ **Specific date** (**mmdd** or **ddmm** format). ▪ **ODAT**. Automatically replaced by the job's original scheduling date when the job is ordered. ▪ **PREV** Automatically replaced by the job's previous scheduling date when the job is ordered (or for a forced job, ODAT-1). ▪ **NEXT** Automatically replaced by the job's next scheduling date when the job is ordered (or for a forced job, ODAT+1). ▪ ******** (or **$$$$**) Any condition date. For deleting a prerequisite condition, only. When specified, all prerequisite conditions with the specified condition name are deleted, regardless of their dates. ▪ **STAT** Condition is not date-dependent. Note that this value is valid only for Control-M/Server version 6.0.01 and later. **NOTE: PREV** and **NEXT** cannot be specified when modifying details of a job in the Active Jobs database. A date reference is required for each condition. However, beginning with version 6.1.03 of Control-M/Server, ODAT can be specified when modifying details of a job in the Active Jobs database.	
Sign	Indicates if the condition should be added (created) or deleted. ▪ **+** Adds (creates) the condition ▪ **–** Deletes the condition If the value of ODAT is $$$$ or ****, + cannot be selected.	

The **Add/Remove Condition** parameter is optional. However, each Condition parameter specified must have a Date reference and a Sign specification.

When the criteria specified in the On Statement/Condition parameter are satisfied, the designated prerequisite condition(s) are added or deleted from the Control-M Conditions List.

If parameters Add/remove Condition and Out Conditions perform opposing actions on the same prerequisite condition and date, the **Add/Remove Condition** parameter overrides the **Out Conditions** parameter.

For more information on the **Out Conditions** parameter, see Out Condition (on page 200)

NOTE: A maximum of 99 prerequisite conditions can be specified for the **Add/Remove Condition** parameter.

Order Job

The Order Job parameter forces an individual job or all jobs in a folder to be placed in the Active Jobs file (regardless of each job's Scheduling criteria) when the On Statement/Code criteria are satisfied. This can be done on a local Control-M/Server or on a remote Control-M/Server.

Order Job causes the specified job to be placed immediately in the Active Jobs database. The job is submitted for execution as soon as all its submission criteria are satisfied.

Additional information	Description
Usage	Optional
Format	Select **Order Job** from the Do list box. The **Control-M**, **Folder**, **Job Name**, and **Date** text boxes are displayed. These sub-parameters are described below.
Variable Support	A variable or expression can be specified as all or part of the value for this parameter. This is only relevant to local variables.
Alternate names	▪ **Control-M/EM Utilities:** FORCEJOB ▪ **Reporting Facility:** (none) ▪ **Control-M/Server Utilities:** -forcejob ▪ **Control-M for z/OS:** FORCE JOB ▪ **Control-M/EM API:** force_job
Alternate formats	
Control-M/EM Utilities	FORCEJOB consists of the DSN, FOLDER_NAME, NAME, and ODATE sub-parameters. The value of DSN is the Folder Library name, which must be specified for z/OS jobs. **EXAMPLE:** <FORCEJOB DSN="z/OS.FLD" FOLDER_NAME="fld1" JOB="Job3" ODATE="DATE"/>
Control-M for z/OS	ORDER JOB consists of the FOLDER, NAME, and LIBRARY sub-parameters. The value of DSN is the Folder Library name, which must be specified for z/OS jobs.
Sub-parameters	

Additional information	Description	
Control-M	List of remote Control-M machines. **NOTE:** Only Control-M/EM versions 9 and above are supported.	
Library	Auto-populated for Control-M for z/OS	
Folder	Name of the job folder	
	Length	1 through 20 characters
	Invalid Characters	Blanks; single quotation marks OS/900: Non-English characters
Job Name	Job name. If this field is blank, all jobs in the specified folder are forced.	
	Length	▪ Computers other than z/OS: 1 through 64 characters. ▪ z/OS: 1 through 8 characters.
	Invalid Characters	Blanks (embedded); single quotation marks
Date	Value to be used as the original scheduling date for the job. Valid values are:	
	ODAT	Resolves to the original scheduling date of the job resulting in the execution of this parameter (default). **NOTE:** As of Control-M/Server version 6.1.03, ODAT can also be specified when modifying details of a job in the Active Jobs database.
	date	A date reference, 4 or 6 characters long (either **mmdd**, **ddmm**, **yymmdd**, or **yyddmm** format, depending on the site standard). z/OS: Only the yymmdd or yyddmm format can be used

NOTES:

- When Order Job forces a job that belongs to a SMART Folder, it treats it as a regular job and forces it alone (that is, without a SMART Folder)

- When Ordering a job on a remote Control-M/Server, the action is asynchronous. Both the action itself and variables are passed on to the remote Control-M server. If communication is down between the local and remote servers, the request will be sent once the remote server is up again. A check will be made every 48 hours
- If multiple On/Do actions have been defined on a remote Control-M/Server, the order in which they will be forced on the remote Control-M server may not reflect the order of the On/Do actions. So On/Do action #2 may be forced before action #1.

Notify

The Notify parameter specifies a notification to be sent a specific destination when the specified On Statement/Code criteria are satisfied.

Additional information	Description
Usage	Optional
Format	Drop-down list Notify has the following options: ▪ Mail (on page 157) ▪ Remedy (on page 162) ▪ Alerts Window ▪ Control-M Log File ▪ User Console ▪ ECS
Invalid Characters	z/OS, except for Message sub-parameter: Non-English characters
Variable Support	Yes. A variable or expression can be specified as all or part of the value for this parameter.
Alternate names	▪ **Control-M/EM Utilities:** NOTIFY ▪ **Reporting Facility:** (none) ▪ **Control-M/Server Utilities:** -notify ▪ **Control-M for z/OS:** NOTIFY ▪ **Control-M/EM API:** notify

Mail

A mail message that is sent to the indicated e-mail addresses when the condition specified by the On Statement/Code parameter is satisfied.

Additional information	Description	
To	E-mail addresses for recipients of the mail message. Multiple addresses can be specified, separated by semicolons ";". Mandatory. **NOTE:** Only physical addresses or variables can be specified. Logical addresses (as specified for the Notify (on page 156) parameter) are not supported for Mail.	
	Length	Computers other than z/OS: 1 through 255 characters z/OS: 1 through 255 characters
	Case Sensitive	Yes
	Invalid Characters	Computers other than z/OS: Blanks; single quotation marks z/OS: Blanks; non-English characters
	Variable Support	Yes. A variable or expression can be specified as all or part of the value for this parameter.
CC	E-mail addresses that should be CCed for the mail message. Multiple addresses can be specified, separated by semicolons (";"). Optional. **NOTE:** On some UNIX computers, all recipients are shown in the **To** field because the **CC** field is not supported.	
	Length	Computers other than z/OS: 1 through 255 characters z/OS: 1 through 255 characters
	Case Sensitive	Yes
	Invalid Characters	Computers other than z/OS: Blanks; single quotation marks z/OS: Blanks; non-English characters

Additional information	Description	
	Variable Support	No. A variable or expression cannot be specified as all or part of the value for this parameter.
Subject	Subject line for the message.	
	Length	Computers other than z/OS: 1 through 99 characters z/OS: 1 through 70 characters
	Case Sensitive	Yes
	Invalid Characters	Computers other than z/OS: Single quotation marks; do not use "-" as the first character. z/OS: Non-English characters
	Variable Support	No. A variable or expression cannot be specified as all or part of the value for this parameter.
message	Text of the mail message.	
	Length	Computers other than z/OS: 1 through 4000 characters z/OS: 1 through 255 lines, each containing 1 through 70 characters However, do not exceed 4000 characters.
	Case Sensitive	Yes
	Invalid Characters	Computers other than z/OS: Single quotation marks z/OS: None
	Variable Support	Yes. A variable or expression can be specified as all or part of the value for this parameter.

Additional information	Description	
Urgency	Select from the following buttons, which indicate a level of urgency for the message: ▪ **Regular** Indicates that the message should have a regular level of urgency. ▪ **Urgent** Indicates messages with a high priority. Urgent message are sent with a special indication so that the recipient of the message is aware of the urgency. ▪ **Very Urgent** Indicates that the message should have the highest level of urgency. For Control-M/Server utilities, only.	
Attach Output	Specifies at the job level whether the output should be sent as an email attachment.	
	Format	List
	Values	▪ **Yes**: Send the job's output as an attachment ▪ **No**: Do not send the job's output as an attachment ▪ **Default**: Refers to the ADD OUTPUT TO EMAIL value in the config.file to determine whether to send the job's output as an attachment.

End Job Not Ok

The End Job Not OK parameter assigns a completion status of **NOTOK** to a job when the On Statement/Code criteria are satisfied, regardless of the job's actual completion status.

Additional information	Description
Usage	Optional
Format	Select **End Job Not Ok** from the Do list box. No additional information is required.
Variable Support	None
Alternate names	▪ **Control-M/EM Utilities:** NOTOK ▪ **Reporting Facility:** (none) ▪ **Control-M/Server Utilities:** -donotok ▪ **Control-M for z/OS:** DO NOTOK ▪ **Control-M/EM API:** do
Alternate formats	
Control-M/EM Utilities	**NOTOK** is a valid value for the DO parameter. It is not a parameter itself.
Control-M for z/OS	**NOTOK** is a valid value of the DO parameter. It is not itself a parameter.

For most jobs:

- If **End Job Ok** and **End Job Not Ok** are both specified, and are implemented, the last statement to be implemented determines the status assigned to the job.
- For information about the **End Job Ok** parameter, see End Job Ok (on page 161).

For z/OS jobs:

- End Job Not Ok overrides the completion status of a job and changes it to NOTOK.
- Do NOTOK cannot be specified together with Do Rerun or Do OK for the same codes-event.

When:

End Job Ok, End Job Not Ok, and/or Rerun Job are specified for different codes-events in a job processing definition.

-and-

More than one of the codes-events is executed, including the codes-event specifying End Job Ok.

Then:

End Job Ok is overridden by End Job Not Ok and/or Rerun Job, regardless of the order in which the codes-events are executed.

End Job Ok

The End Job Ok parameter assigns the completion status of **OK** to a job, regardless of its actual completion status.

Additional information	Description
Usage	Optional
Format	Select **End Job Ok** from the Do list box. No additional information is required.
Variable Support	No. A variable or expression cannot be specified as all or part of the value for this parameter.
Alternate names	▪ **Control-M/EM Utilities:** OK ▪ **Reporting Facility:** (none) ▪ **Control-M/Server Utilities:** -dook ▪ **Control-M for z/OS:** DO OK ▪ **Control-M/EM API:** do

For most jobs:

If End Job Ok and End Job Not ok are both specified, and are implemented, the last statement to be implemented determines the status assigned to the job.

For z/OS jobs:

- End Job Not Ok overrides the completion status of a job and changes it to NOTOK.
- End Job Not Ok cannot be specified together with rerun Job or End Job Ok for the same codes-event.

When:

End Job Ok, End Job Not Ok and/or Rerun Job are specified for different codes-events in a job processing definition.

-and-

More than one of the codes-events is executed, including the codes-event specifying End Job Ok.

Then:

End Job OK is overridden by Do Job Not Ok and/or Rerun Job, regardless of the order in which the codes-events are executed.

Remedy

Opens a ticket in the Remedy Help Desk.

Additional information	Description
Usage	Optional
Format	Consists of sub-parameters, described below.
Invalid Characters	The following special characters are not permitted in the Remedy parameter or its sub-parameters. ▪ Single quotation mark (‘) ▪ Double quotation marks (") ▪ Equals (=) ▪ Less than (<) ▪ Greater than (>) ▪ Ampersand (&) ▪ Backslash (\) ▪ Non-English characters
Variable Support	Yes
Sub-parameters	
Urgency	The urgency level of the ticket that will be opened in Remedy. Mandatory. Valid values are: ▪ L = Low (Default) ▪ M = Medium ▪ H = High ▪ U= Urgent ▪ C = Clear
Summary	A brief summary is displayed in Remedy. By default, a summary of the problem appears using variables. For more information on the field’s characteristics, refer to Remedy documentation.
Description	A detailed description is displayed in Remedy. By default, a description of the problem appears using variables. For more information on the field’s characteristics, refer to Remedy documentation.

The action parameter Remedy is applicable on Control-M/Server and on jobs within the batch service.

Rerun Job

Indicates if an automatic rerun should be performed when the On Statement/Code criteria are satisfied.

Additional information	Description
Usage	Optional
Alternate names	▪ **Control-M/EM Utilities:** RERUN ▪ **Reporting Facility:** (none) ▪ **Control-M/Server Utilities:** -rerunjob ▪ **Control-M for z/OS:** RERUN JOB ▪ **Control-M/EM API:** do

Rerun Job specifies that the job should be rerun according to the criteria specified in the Maximum reruns (on page 100) and Rerun Every (on page 93) parameters. These parameters must be specified before a Rerun Job action can be defined.

NOTE: Rerun Job cannot be used to rerun SMART Folders.

For most jobs:

If Rerun Job is specified, and End Job Ok or End Job Not Ok is specified, and implemented, the last statement to be implemented determines the status assigned to the job.

For z/OS jobs:

When a Rerun Job action is specified for a job, the job's completion status is set to NOTOK, even if it was previously specified as OK.

- Cyclic jobs cannot contain a Rerun Job parameter.
- Rerun Job cannot be specified together with End Job Ok or End Job Not Ok for the same codes-event.

  When:

  End Job Ok, End Job Not Ok, and/or Rerun Job are specified for different codes-events in a job processing definition.

  -and-

  More than one of the codes-events is executed, including the codes-event specifying End Job Ok.

  Then:

  End Job Ok is overridden by End Job Not Ok and/or Rerun Job, regardless of the order in which the codes-events are executed.

Stop Cyclic Run

When specified, this parameter prevents subsequent iterations of the current cyclic job.

Additional information	Description
Usage	Optional
Alternate names	▪ **Control-M/EM Utilities:** SPCYC ▪ **Reporting Facility:** (none) ▪ **Control-M/Server Utilities:** -stopcyclicrun ▪ **Control-M for z/OS:** STOPCYCLRUN ▪ **Control-M/EM API:** do

Cyclic jobs normally run a regular intervals for as long as specified scheduling criteria are satisfied.

If the related On Statement/Code criteria are satisfied, Stop Cyclic Run prevents future iterations of the current cyclic job.

If the current job is not a cyclic job, this parameter has no effect on job processing.

If a cyclic job is terminated by a **Stop Cyclic Run** parameter, the View Details screen displayed by option Z in the ctmpsm utility contains **Cyclic:T** where **T** indicates "Terminated".

Handle Output

Indicates how the job's output should be handled when the On Statement and Code criteria are satisfied.

Additional information	Description
Usage	Optional
Format	Select **Handle Output** from the Do list box and specify the relevant sub-parameters. The following sub-parameters are available for this parameter:

Additional information	Description	
	Option	Indicates what to do with the output data. The options that can be selected in this list box are: Computers other than z/OS: ▪ Copy ▪ Delete ▪ Move ▪ Print z/OS: ▪ None ▪ Change jobs class ▪ Delete output ▪ Copy output ▪ Move output
	File	Name of the file.
	The format for this parameter is the same as specified for Output Handling. For more information about this parameter, see Action (on page 202)	

Additional information	Description	
Variable Support	A variable or expression can be specified as all or part of the value for this parameter.	
Alternate names	▪ **Control-M/EM Utilities:** HANDLEOUTPUT ▪ **Reporting Facility:** (none) ▪ **Control-M/Server Utilities:** -handleoutput ▪ **Control-M for z/OS:** HANDLE OUTPUT ▪ **Control-M/EM API:** handle_output	
Alternate formats		
Control-M/EM Utilities	HANDLEOUTPUT is composed of the parameters below. **EXAMPLE:** <HANDLEOUTPUT OPTION="ChangeClass" PAR="F" FROM="C"/>	
	OPTION	Indicates what to do with the Output data. Mandatory. Valid values: ▪ Release ▪ Delete ▪ Copy ▪ Move **NOTE:** Copy and Move are not used with z/OS.
	PAR	Certain OPTION values require that you supply additional information (such as **Release, NewDest**). The PAR parameter holds that information as a string.
	FROM	Limits the Output handling operation to only Outputs from the specified class.

Additional information	Description	
Control-M for z/OS	Handle Output is composed of the parameters below.	
	OPT	Output option code. Mandatory. Valid values: ▪ **C** – Change the class of the job output. ▪ **D** – Delete (purge) the job output. ▪ **F** – Copy the job output to file. ▪ **N** – Change destination of job output. ▪ **R** – Release the job output.
	data	Relevant Output data. Mandatory and valid only if the specified OPT value is C, **F**, or **N**. Valid values depend on the OPT value, as follows: ▪ **F** – File name. String comprised of from 1 through 44 characters. All characters are valid except blanks. ▪ **C** – New class (1 character). Any character is valid except blank, but an asterisk (*****) indicates the original MSGCLASS of the job. ▪ **N** – New destination (1 through 8 characters). All characters are valid except blanks.
	FRM	FROM class. Optional. Limits the Output handling operation to only Outputss from the specified class.

If no Output Handling is specified (or the job does not end **OK**), and no Handle Output statement is activated, the job's log is placed in the default location specified by Control-M until the New Day procedure performs its cleanup.

NOTE: The default destination of the Job log is determined by a Control-M system parameter on each computer. For more information, see your Control-M administrator.

The Automatic Log Copy Control-M system parameter is not affected in any way by the function of Handle Output.

Any program step

Indicates that the DO statements must be performed if the specified codes are found in any program step.

However, if **Any program step** is specified with the codes **Job Ended OK**, **Job Ended Not OK**, **Any type of execution error**, **Job out put was lost**, **Job was canceled during executing**, **Job failed due to security requirements**, **Job not submitted**, or ***Unknown error occurred**, the On criteria are satisfied only if the entire job ends with the specified code criteria.

If Any program step is specified with the **ORDER** code, no other codes can be specified in the same On block, and the **Procedure** field must be left blank.

Additional information	Description
Usage	Optional
Format	The **Any program step** option consists of the following sub-parameters: ▪ Procedure ▪ Return statuses
Variable Support	None
Sub-parameters	
Procedure	Defines the procedure name. 1 through 8 characters in length, and must not contain blanks. Non-English characters not allowed. When left blank, any program step names are checked regardless of whether they are directly from the job or from a called procedure. Default. The On statement is satisfied if the criteria are satisfied from any procedure directly from the job.
Return statuses	See Code values (on page 175)

Every program step

Indicates that the value is used without being accompanied by limiting step values when the code criteria is satisfied for every step. The code criteria is satisfied for every step in the job without exception.

Additional information	Description
Usage	Optional
Format	The **Every program step** option has the following sub-parameters: ▪ Procedure ▪ Return statuses
Variable Support	None
Sub-parameters	
Procedure	Defines the procedure name. 1 through 8 characters in length, and must not contain blanks. Non-English characters not allowed. **None**: Any program step names are checked regardless of whether they are directly from the job or from a called procedure. Default. The On statement is satisfied if the criteria are satisfied from any procedure directly from the job. **Every**: The code criteria is satisfied for every step in the job without exception.
Return Statuses	See Code values (on page 175).

Specific Step Name

Name of a specific procedure/program step. If a specific procedure/program step is specified, only program/procedure steps from the invoked procedure/program are checked to see if they satisfy the code criteria. Program/procedure steps directly from the job are not checked.

Additional information	Description
Usage	Optional
Format	The **Specific Step Name** option consists of the following sub-parameters: ▪ Procedure ▪ Program ▪ Return statuses
Variable Support	None
Sub-parameters	
Procedure	Defines the procedure name. 1 through 8 characters in length, and must not contain blanks. Non-English characters not allowed.
Program	Defines the program name. 1 through 8 characters in length, and must not contain blanks. Non-English characters not allowed.
Return statuses	See Code values (on page 175).

JOBRC

Assign a completion code for the entire job based on the completion codes of its steps.

Additional information	Description
Usage	Optional
Format	The **JOBRC** option consists of the **Return Statuses** sub-parameter.
Variable Support	None
Sub-parameters	
Return statuses	See Code values (on page 175).

The parameter has the following options:

- The highest return code from the steps (the default) – this is the same as how Control-M assigned the job's completion code prior to z/OS 1.13. If the job's execution fails because of an ABEND, the job completion code is set to the last ABEND code.
- The return code of the last step - in this case the job's completion code is based on the completion code of its last executed step.
- The return code of a specific step - in this case the job's completion code is based on the completion code of a specific step identified by its name.

Control-M extracts the JOBRC, which is calculated by JES2 or JES3 based on the JOBRC parameter specified in the JCL JOB statement, and use it for the job analysis during post processing.

An ON PGMSTEP +JOBRC statement can specify any DO action to be processed when the extracted JOBRC fits the codes specified in the statement. JOBRC is extracted in all z/OS releases (in releases earlier than z/OS 1.13, it is set equal to MAXRC) so ON PGMSTEP +JOBRC can be handled in any z/OS release.

In some special situations Control-M does not extract the JOBRC. Either Control-M is not able to extract JOBRC (when JOBRC not supplied by z/OS or JES), or it is assumed that users do not require that Control-M extract JOBRC (when the results based on JOBRC would be different than what users actually need). Some special situations are described in the following examples:

- The job execution was terminated by the 'Cancel' operator command.
- The job execution was terminated prematurely according to the COND=parameter specified in the JOB JCL statement.
- The job execution is the restart run performed by Control-M/Restart and the
- ALLRUNS Control-M/Restart installation parameter, which specifies whether Control-M considers during post processing all the previous runs of a job, is activated.

If JOBRC is not extracted, the ON PGMSTEP +JOBRC statement is ignored.

An ON PGMSTEP +JOBRC statement cannot be connected by AND/OR Boolean relationships with other ON PGMSTEP statements.

Since JOBRC is not used for evaluating the OK/NOTOK status of job when DO OK/NOTOK actions are defined in the ON PGMSTEP statements of step(s), DO OK/NOTOK actions in ON PGMSTEP +JOBRC cannot be defined together with DO OK/NOTOK in the ON PGMSTEP statements of step(s).

Output pattern

Indicates that the DO statements must be performed if the specified pattern is found in the output.

Additional information	Description
Usage	Optional
Format	The **Output pattern** consists of the following sub-parameters: ▪ Pattern ▪ From Col ▪ To Col
Variable Support	None
Sub-parameters	
Pattern	A string in the output. 1-40 characters
From Col	The first column to look for the pattern. 3 digit number.
To Col	The last column to look for the pattern. 3 digit number.

Specific Range Name

Specifies a range of steps in the steps of an On PGMST statement.

Additional information	Description	
Usage	Optional	
Format	Consists of the **Program** and **Return Status** sub-parameters below.	
Alternate names	▪ **Control-M/EM Utilities:** STEP_RANGE ▪ **Reporting Facility:** Composed of the FROM PGMST, FROM PROCS TO PGMSTEP, and TO PROCSTEP parameters. ▪ **Control-M/Server Utilities:** (none) ▪ **Control-M for z/OS:** STEP RANGE ▪ **Control-M/EM API:** step_range	
Alternate formats		
Control-M/EM Utilities	STEP_RANGE is composed of the following parameters:	
	NAME	1-7 character string.
	FPGMS	First program step in the range. 1-8 character string.
	FPROCS	First process step in the range. 1-8 character string.
	TPGMS	Last program step in the range. 1-8 character string.
	TPROCS	Last process step in the range. 1-8 character string.
Sub-parameters		
Program	Defines the program name.	
Return Status	See Code values (on page 175).	

Whenever a Specific Range Name statement is specified, it eliminates the need to define separate On PGMST, On PROCST, and On Codes statements and accompanying Do actions for each step in the range. The defined Specific Range Name can be used (without redefining the range) in subsequent On PGMST, On PROCST, and On Codes statements, by specifying the Specific Range Name, preceded by an asterisk (*****), in the On PGMST field.

Any number of step ranges can be specified. After entering a Specific Range Name parameter, another Specific Range Name parameter line is automatically displayed.

Using all runs of a job including restarts

When processing On blocks, Control-M can incorporate the results of all previous runs and restarts, filtering them for jobs restarted with the Control-M for z/OS RESTART, RECAPTURE CONDITION or ABEND CODES parameters. Control-M/Restart searches previous runs to determine which steps must be considered part of the restarted job.

For example, if one step finished successfully during its original run and another step finished successfully after a restart, the On block check for the successful finish for both steps produces a **TRUE** result and the On statement is satisfied.

Activation of this facility requires that the Control-M for z/OS ALLRUNS parameter in the CTRPARM member be set to **YES**. When activated, this facility may apply to any specified step, step range, or to the **+EVERY** step value.

NOTE: For z/OS jobs only.

EXAMPLE: Control-M Properties Pane

Define program steps STEP20 through STEP29A as step range DF2. If any of these steps produce any system or user abend (except user abend U2030), rerun the job and sends a message to TSO-P43.

EXAMPLE: Control-M for z/OS interface

Define program steps STEP20 through STEP29A as step range DF2. If any of these steps produce any system or user abend (except user abend U2030), rerun the job and sends a message to TSO-P43.

JOB: PRDKPL01 LIB CTM.PROD.SCHEDULE FOLDER: PRODKPL

COMMAND ===> SCROLL===> CRSR

+----------+

==========

OUT

AUTO-ARCHIVE Y SYSDB Y KEEP ACTIVE FOR MAXIMUM RERUNS

RETENTION: # OF DAYS TO KEEP 030 # OF GENERATIONS TO KEEP

OUPUT OP (C,D,F,N,R) FROM

MAXIMIM RERUNS RERUNMEM INTERVAL FROM

STEP RANGE DF2 FR (PGM.PROC) STEP20 . TO STEP29A .

STEP RANGE FR (PGM.PROC) . TO .

ON PGMST *DF2 PROCST CODES S**** U**** NU2030 A/O

```
RERUN JOB
NOTIFY TO TSO-P43 URGENCY R
= JOB PRDKPL03 ABENDED, THE JOB IS RERUN
DO
ON PGMST PROCST CODES A/O
DO
NOTIFY WHEN TO URGN
MS
======= >>>>>>>>>>>>>>>>>>> END OF SCHEDULING PARAMETERS
<<<<<<<<<<<<<<<< =====
COMMANDS: EDIT, DOC, PLAN, JOBSTAT 11.17.00
```

Code values

Code values can be condition codes, user abend codes, system abend codes, various end codes and statuses, and certain keywords. They can also be prefaced by certain qualifiers. All of these are described below.

A maximum of 245 codes can be specified for any On step statement, as follows:

- Each line of an On statement contains fields for specification of up to four codes.
- Whenever a fourth code on a line is specified, and **Enter** is pressed, a new line within the same On statement is opened, allowing specification of up to another four codes.

NOTE: If a End Job Ok (on page 161) statement is specified in the job scheduling definition, it is ignored for steps for which any of the following codes apply: Job was canceled during executing, Job not submitted, Maximum reruns number reached, unknown error occurred.

Value	Description
Condition return code	Step condition code. 4-digit value.
System ABEND code	Step system abend code. 3-character hex value.
User ABEND code	Step user abend code. a 4-digit value.

Value	Description
Job was queued for execution	Any step that executes, including steps with JCL errors and steps returned with an ABEND code. For reasons of backward compatibility, the Job was queued for execution code does not include steps with the **FLUSH** code or **SNRUN** (described below). The Job was queued for execution code does, however, include jobs not submitted and jobs whose output was lost if Any program step (on page 168) is specified.
Job was Set to Ok	This code applies when a Job is **Set to OK** from the Control-M Monitoring screen (Screen 3). To specify a code of **FORCE**, all of the following must apply: ▪ No other code can be specified in the same statement. ▪ The PGMST value must be Any program step (on page 168). ▪ No PROCST value can be specified. ▪ No other On statements can appear in the On block. Valid DO statements for the FORCE code are: ▪ Notify ▪ Add/Remove Condition ▪ Order Job ▪ Set Variable ▪ Notify
Job Output was lost	Job Output was lost. This value can be specified only with the Any program step **(on page** 168**)** step value.
Job was canceled during executing	Job was canceled during execution or re-execution. This value can be specified only with the Any program step **(on page** 168**)** step value.
Job failed due to JCL error	Job failed due to JCL error.
Job failed due to security requirements	Job failed due to security requirements (only under ACF2). This value can be specified only with the Any program step **(on page** 168**)** step value.
Job not submitted	Job not submitted. Submission of a job or initiation of a started task failed for any reason. This value can be specified only with the Any program step **(on page** 168**)** step value.
Job Ended OK	A program step finished executing **OK** (by default with a condition code of **C0004** or less). This value can be specified only with the Any

Value	Description
	program step **(on page** 168**)** step value. **NOTE:** Through parameter MAXCCOK in member CTMPARM in the IOA PARM library, the default condition code can be set to **C0000**. If a job is **ORDERED OK**, the DO statements following an any program step.. CODES OK statement are processed only if the FRCOKOPT parameter in the CTMPARM member in the IOA PARM library is set to **Y**.
Job Ended Not OK	A program step (or the job) finished executing **NOTOK**. This code covers all types of failures, including non-execution errors (for example, job not run, JCL error, job not submitted), and (by default) any condition code greater than **C0004**. This value can only be specified with the Any program step **(on page** 168**)** step value. You can use the MAXCCOK parameter in the CTMPARM member in the IOA PARM library to set the default condition code to **C0000**.
Any type of execution error	Any type of execution error. It is the same as **NOTOK**, but is triggered only if the job has actually started executing. This value can only be specified as the Any program step (on page 168) step value.
File allocation problem (NCT2)	A NOT CATLGD 2 or NOT RECATLGD 2 event occurred in the job step. The default result of this event is a **NOTOK** status for the step. A message containing the data set name is written to the IOA Log file. If you do not want to be alerted to NOT RECATLGD 2 events, see your INCONTROL administrator.
Job terminated by CMEM	Job terminated by CMEM due to an NCT2 event.
Maximum reruns number reached	Rerun (recovery) is needed, but no more reruns are available. **REC** is followed by a zero (**0**), not a letter O.
unknown error occurred	An unknown error occurred, usually as a result of a computer crash during job execution. This value can only be specified with the Any program step (on page 168) step value.
Job was queued for execution	Job was queued for re-execution.
Step not executed (FLUSH)	A JCL COND or JCL IF/THEN/ELSE statement caused a step to not run. This code is described in more detail in below.
Step not executed	A step did not run.

Step not executed (FLUSH)

The FLUSH code generally applies when a step does not run but no error is indicated. This code is assigned when:

- A JCL COND or JCL IF/THEN/ELSE statement caused the step not to run. Control-M detects code FLUSH steps by the IEF272I message (Step was not executed).
- If a job was restarted by Control-M/Restart, and Control-M is to consider all job runs during post-processing (ALLRUNS is set to YES in the CTRPARM member), a step is defined as FLUSH if both the following statements are true:
  - Either the step did not previously run, or Control-M/Restart did not recapture a completion or abend code from a previous run
  - Either of the following statements is also true:
    - It was not executed during the RESTART run because of a JCL COND or JCL IF/THEN/ELSE statement.
    - It was not executed due to a RESTART decision (the CTR103I message).
    - Because a code of FLUSH does not indicate that an error occurred during job execution, assignment of this status does not cause a job status of **NOTOK**.
    - If a JCL statement other than the COND or IF/THEN/ELSE statement caused the step not to run, it is not defined as a FLUSH step.
    - If the failure of a step causes subsequent steps not to be executed, these subsequent steps are not defined as FLUSH steps.

For reasons of backward compatibility (that is, to ensure that the application of the **Job was queued for execution** code remains unchanged), the **Job was queued for execution** code does not include FLUSH steps.

Step not executed

A step is defined as code step not executed if it did not run. This code includes:

- Any step with a code of FLUSH.
- Any step that does not appear in the job.
- Instances where a step does not run because of a JCL error in a prior step (the step with the JCL error does not have a status of SNRUN)
- If a job was restarted by Control-M/Restart, and Control-M is to consider all job runs during post-processing (the ALLRUNS parameter is set to YES in the CTRPARM member), a step is defined as step not executed if both the following statements are true:
  - Either the step did not previously run, or Control-M/Restart did not recapture a completion or abend code from a previous run.
  - The step was not executed during the RESTART run.

Step not executed cannot be specified together with Any program step (on page 168). (Because Step not executed includes steps that do not exist in a job, and any program step includes all step names even if they do not exist in a job, specifying both in the same job would cause a condition that Step not executed could not process.).

A status of Step not executed does not indicate that an error occurred during a job execution, nor does it cause a job status of **NOTOK**. It only indicates that it did not run.

For backward compatibility (that is, to ensure that the application of the **Job was queued for execution** code remains unchanged), the **Job was queued for execution** code does not include Step not executed steps.

Code qualifiers and relationships

Any character in a condition code, system abend code or user abend code may be replaced by an asterisk (*****). An asterisk means "any value" for the character it replaces. For example, if **S*13** is specified, the code criteria for the step is satisfied by codes **S013**, **S613**, **S913**, and so on.

The additional qualifiers in the following table can be used in specific circumstances.

Qualifier	Description
>	Greater than. Valid as a qualifier for condition codes and user abend codes.
<	Less than. Valid as a qualifier for condition codes and user abend codes.
N	Specifies not to perform the accompanying DO statements if the specified code exists in the step. Valid as a qualifier for condition codes, user abend codes and system abend codes.

The **N** qualifier indicates that the DO statements must not be performed if the specified condition exists. It does not indicate that the DO statements must be performed if the specified condition does not exist.

The relationship between multiple codes in an On statement is OR (that is, the appearance of any of the codes in the specified step satisfies the On criteria), except for range specifications (for example, >10 <40).

However, code criteria qualified by N take precedence over all other code criteria. If a code that is specified with an N qualifier is generated by the specified step, accompanying DO actions are not performed even if other On code criteria are satisfied.

If Step1 ends with a condition code of C0004 and Step 5 ends with system abend code S0C4, perform the indicated notification.

On Statement PGMST= Step1 PROCST= Codes=C0004 AND

On Statement PGMST= Step5 PROCST= Codes=S0C4

Notify To=emuser Urgn=Regular Msg= Backup operations ...

Control-M Analyzer Rule

Invokes a Control-M/Analyzer rule to be executed.

Additional information	Description	
Usage	Optional	
Variable Support	None	
Alternate names	▪ **Control-M/EM Utilities:** DOCTBRULE ▪ **Reporting Facility:** (none) ▪ **Control-M/Server Utilities:** (none) ▪ **Control-M for z/OS:** DO CTBRULE ▪ **Control-M/EM API:** do_ctbrule	
Alternate formats		
Control-M/EM Utilities	DOCTBRULE contains the following sub-parameters.	
	NAME	Name of the Control-M/Analyzer rule that is to be executed.
	PAR	Arguments
Sub-parameters		
name	Text box. Name of the Control-M/Analyzer rule that is to be executed. The Control-M/Analyzer rule contains all balancing specifications to be performed. Mandatory.	
	Length	1 through 8 characters
	Case Sensitive	Yes
	Invalid Characters	Blanks; non-English characters

Additional information	Description	
ARG	Text box. Arguments that are passed to the Control-M/Analyzer rule. Separate multiple arguments by commas. Optional.	
	Length	1 through 45 characters
	Case Sensitive	Yes
	Invalid Characters	Blanks; non-English characters

When Control-M Analyzer is specified, balancing is performed by the Control-M/Analyzer Runtime environment according to the specified rule definition and using the specified arguments. The Control-M/Analyzer Runtime environment is invoked once for each Control-M Analyzer statement in the job scheduling definition.

If Control-M Analyzer is specified under ON PGMST ANYSTEP, the Control-M/Analyzer Runtime environment is invoked only once.

When Control-M calls a Control-M/Analyzer rule, Control-M/Analyzer System variable SYSOPT contains the value **CTMWORK**. This variable can then be tested within the Control-M/Analyzer rule definition to determine if Control-M invoked the Control-M/Analyzer Runtime environment.

When the Control-M/Analyzer Runtime environment is invoked by Control-M, that is, Control-M/Analyzer System variable SYSOPT is set to **CTMWORK**, Control-M/Analyzer can analyze and balance SYSDATA. For more information about invoking Control-M/Analyzer rules from Control-M job scheduling definitions, see the discussion of the interface to Control-M in the *Control-M/Analyzer User* Guide.

This parameter is available only for Control-M for z/OS jobs at sites using Control-M/Analyzer.

EXAMPLE: Execute a Control-M/Analyzer rule when a job ends OK

If the job ends OK, execute Control-M/Analyzer balancing rule GOVTBAL.

```
JOB: GOVTREPT LIB CTM.PROD.SCHEDULE FOLDER: BACKUP
COMMAND ===> SCROLL===> CRSR
+-------------------------------------------------------------------------+
TIME: FROM UNTIL PRIORITY DUE OUT SAC CONFIRM
TIME ZONE:
=========================================================================
OUT FINANCE-GOVTREPT-OK ODAT +
AUTO-ARCHIVE Y SYSDB Y KEEP ACTIVE FOR MAXIMUM RERUNS
```

```
RETENTION: # OF DAYS TO KEEP 030 # OF GENERATIONS TO KEEP
Output OP (C,D,F,N,R) FROM
MAXIMUM RERUNS RERUNMEM INTERVAL FROM
STEP RANGE FR (PGM.PROC) . TO .
ON PGMST ANYSTEP PROCST CODES OK A/O
DO CTBRULE = GOVTBAL ARG DOREPORT,10,%%ODATE
DO
ON PGMST PROCST CODES A/O
DO
NOTIFY WHEN NOTOK TO TSO-M44 URGN R
MS JOB GOVTREPT ENDED "NOT OK"
NOTIFY WHEN TO URGN
MS
======= >>>>>>>>>>>>>>>>>> END OF SCHEDULING PARAMETERS
<<<<<<<<<<<<<< ======
COMMANDS: EDIT, DOC, PLAN, JOBSTAT 11.17.00
```

Set Restart Options

Job steps to be executed during restart of a job.

Additional information	Description	
Usage	Optional	
Format	Select **Set Restart Options** from the **Do** list box. **From** and **To** text boxes and a **Confirm** check box are displayed. Enter the required information. Select the check box, if required. These sub-parameters are described below.	
Variable Support	None	
Alternate names	▪ **Control-M/EM Utilities:** DOIFRERUN ▪ **Reporting Facility:** (none) ▪ **Control-M/Server Utilities:** (none) ▪ **Control-M for z/OS:** DO IFRERUN ▪ **Control-M/EM API:** do_ifrerun	
Alternate formats		
Control-M/EM Utilities	DOIFRERUN is composed of the following sub-parameters:	
	CONFIRM	Valid values: ▪ **0** (No confirmation. Default) ▪ **1** (Confirm)
	FPGMS	First program step in the range. 1-8 character string.
	FPROCS	First process step in the range. 1-8 character string.
	TPGMS	Last program step in the range. 1-8 character string.
	TPROCS	Last process step in the range. 1-8 character string.
Control-M for z/OS	Set Restart Options is composed of the following sub-parameters:	
	FROM	First program or process step in the range. 1-8 character string.

Additional information	Description	
	TO	Last program or process step in the range. 1-8 character string.
	CONFIRM	Valid values: ▪ **N** (No confirmation. Default) ▪ **Y** (Confirm)

Additional information	Description
Sub-parameters	

From	Step at which the job must be restarted. Mandatory.
	Valid values:
	▪ **pgmstep** – Program step within the job stream.
	▪ **pgmstep.procstep** – Program step within the called procedure.
	▪ **$FIRST** – First step of the job.
	▪ **$ABEND** – Step of the job that ended **NOTOK** due to system abend, user abend, condition code C2000 (PL/1 abend) or JFAIL (job failed on JCL error). **$ABEND** is a subset of **$EXERR** (below).
	▪ **$FIRST.$ABEND** – First step of the abended procedure.
	▪ **$FIRST.$CLEANUP** – This reserved keyword instructs Control-M to run a Control-M/Restart data set cleanup for the job. Data set cleanup is performed from the first step of the job. The job itself is not restarted.
	▪ **$EXERR** – Job step that ended with any error, including an abend, or that ended with a condition code that is redefined using the On and DO statements as **ENDED NOTOK**.
	NOTE: For both From and To steps, **pgmstep** is the name of the step (EXEC statement) that executes the program from which to begin or end the restart:
	// pgmstep EXEC PGM= program
	procstep is the name of the step (EXEC statement) that invokes the procedure from which the above **pgmstep** program is executed:
	// procstep EXEC procedure
	pgmstep and **procstep** values can each be **from 1** through **8** characters, and must not contain blanks.
	When specifying a **procstep** when the procedures are nested, the innermost **procstep** in which the program is included must be specified.

Additional information	Description

To	Step at which the restarted job must terminate. Optional.
	Valid values are:
	NOTE: Non-English characters are invalid for this sub-parameter.
	▪ **pgmstep** – Program step within the job stream
	▪ **pgmstep. procstep** – Program step within the called procedure.
	If not specified, the restarted job terminates at the last job step that would normally be executed.
	NOTE: For both From and To steps, **pgmstep** is the name of the step (EXEC statement) that executes the program from which to begin or end the restart:
	// pgmstep EXEC PGM= program
	procstep is the name of the step (EXEC statement) that invokes the procedure from which the above **pgmstep** program is executed:
	// procstep EXEC procedure
	pgmstep and **procstep** values can each be from **1** through **8** characters, and must not contain blanks.
	When specifying a **procstep** when the procedures are nested, the innermost **procstep** in which the program is included must be specified.

Confirm	Specifies whether a manual confirmation is required before the job is restarted. ▪ When the check box in the Properties pane is clear, no confirmation is required. The job restart can be automatically submitted (by the Do Rerun parameter) without a manual confirmation. Default. ▪ When the check box in the Properties pane is selected, confirmation is required. The job restart is not submitted unless the job is confirmed manually from the Control-M/EM flow diagram. Non-English characters are invalid for this sub-parameter.

When a Set Restart Options statement is specified, the rerun is performed by the Control-M/Restart facility using the specified restart sub-parameters.

- When Set Restart Options is specified with a Confirm parameter value of **No** (**Confirm** check box is clear):
- If a Set Restart Options statement follows, the job is automatically submitted for rerun.
- If a Set Restart Options statement does not follow, the job is not automatically rerun. Instead, the job remains displayed with its error status in Control-M Workload Automation.

  In this case, to submit the job for rerun or restart, rerun the job from Control-M Workload Automation. The Rerun (with Restart) Confirmation window is displayed. Request the restart or rerun from the window.
- When Set Restart Options is specified with a Confirm parameter value of Yes (the **Confirm** check box is selected), the job appears in the Control-M Workload Automation with a WAIT CONFIRMATION (WITH RESTART) status and is not restarted unless confirmed. Confirm the job to restart it.

When a job is submitted for restart, if **$FIRST** is specified in the From sub-parameter, a **$FIRST** step specification is passed "as is" to the Control-M/Restart step. If **$ABEND** or **$EXERR** is specified, the specified **$ABEND** or **$EXERR** value is first resolved to the appropriate step by the Control-M monitor and then passed to the Control-M/Restart step.

If **$FIRST.$ABEND** is specified, the Control-M monitor determines which procedure abended and then passes the **$FIRST** step specification for that procedure to the CONTROLR step. For information regarding the Control-M/Restart step, refer to the *Control-M/Restart User Manual*.

The Maximum reruns (on page 100) parameter determines the maximum number of times the restart or rerun can be performed.

- This parameter is available only if Control-M/Restart is installed on a Control-M for z/OS site.
- The Set Restart Options and Rerun Member parameters cannot be specified together.

This parameter is available only if Control-M/Restart is installed on a Control-M for z/OS site.

OS completion status

Determines whether the accompanying DO statements are performed if the OS completion status is satisfied.

Additional information	Description
Usage	Optional
Format	Select **OS completion status** from the **When** drop-down list, and fill in the sub-parameters below.
Variable Name	None
Alternate names	▪ **Control-M/EM Utilities:**ON ▪ **Control-M/Server Utilities:**-on ▪ **e-Trigger:**on_do_statement ▪ **Control-M/EM API:** on_statement
Sub-parameters	
Operator	▪ = (equal) ▪ < (less than) ▪ > (greater than) ▪ ! (not equal) ▪ EVEN (the field is considered satisfied by an even return code ▪ ODD (the field is considered satisfied by an odd return code
Completion Code	The number of the code returned by the operating system.

Additional information	Description
Alternate formats	
Control-M/EM Utilities	ON is composed of the STMT, CODE, PGMS, PROCS, and AND_OR sub-parameters. STMT is a **1**-**132** character string, used only when you are specifying an On statement value.
Control-M/EM API	on_statement is composed of the following sub-parameters: ▪ and_or ▪ code ▪ procedure_step ▪ program_step ▪ statement

Job's number of reruns

Determines whether the accompanying DO statements are performed if the job's number of reruns is satisfied.

Additional information	Description
Usage	Optional
Format	Select **Job's number of reruns** from the **When** drop-down list, and fill in the sub-parameters below.
Variable Name	None
Alternate names	▪ **Control-M/EM Utilities:** ON ▪ **Control-M/Server Utilities:** -on ▪ **e-Trigger:** on_do_statement ▪ **Control-M/EM API:** on_statement
Sub-parameters	

Additional information	Description
Operator	▪ = (equal) ▪ < (less than) ▪ > (greater than) ▪ ! (not equal) ▪ EVEN (the field is considered satisfied by an even return code ▪ ODD (the field is considered satisfied by an odd return code
Number of reruns	The number of reruns for the action to take place
Alternate formats	
Control-M/EM Utilities	ON is composed of the STMT, CODE, PGMS, PROCS, and AND_OR sub-parameters. STMT is a **1**-**132** character string, used only when you are specifying an On statement value.
Control-M/EM API	on_statement is composed of the following sub-parameters: ▪ and_or ▪ code ▪ procedure_step ▪ program_step ▪ statement

Job's number of executions

Determines whether the accompanying DO statements are performed if the job's number of executions is satisfied.

Additional information	Description
Usage	Optional
Format	Select **Job's number of executions** from the **When** drop-down list, and fill in the sub-parameters below.
Variable Name	None.
Alternate names	▪ **Control-M/EM Utilities:** ON ▪ **Control-M/Server Utilities:** -on ▪ **e-Trigger:** on_do_statement ▪ **Control-M/EM API:** on_statement
Sub-parameters	
Operator	▪ = (equal) ▪ < (less than) ▪ > (greater than) ▪ ! (not equal) ▪ EVEN (the field is considered satisfied by an even return code ▪ ODD (the field is considered satisfied by an odd return code
Number of executions	The number of executions for the action to take place.

Additional information	Description
Alternate formats	
Control-M/EM Utilities	ON is composed of the STMT, CODE, PGMS, PROCS, and AND_OR sub-parameters. STMT is a **1-132** character string, used only when you are specifying an On statement value.
Control-M/EM API	on_statement is composed of the following sub-parameters: ▪ and_or ▪ code ▪ procedure_step ▪ program_step ▪ statement

Job's number of failures

Determines whether the accompanying DO statements are performed if the job's number of failures is satisfied.

Additional information	Description
Usage	Optional
Format	Select **Job's number of failures** from the **When** drop-down list, and fill in the sub-parameters below.
Variable Name	None
Alternate names	▪ **Control-M/EM Utilities:**ON ▪ **Control-M/Server Utilities:**-on ▪ **e-Trigger:**on_do_statement ▪ **Control-M/EM API:**on_statement
Sub-parameters	
Operator	= (equal)

Additional information	Description
Number of failures	The number of failures fro the action to take place.
Alternate formats	
Control-M/EM Utilities	ON is composed of the STMT, CODE, PGMS, PROCS, and AND_OR sub-parameters. STMT is a **1**-**132** character string, used only when you are specifying an On statement value.
Control-M/EM API	on_statement is composed of the following sub-parameters: ▪ and_or ▪ code ▪ procedure_step ▪ program_step ▪ statement

Specific statement output

Determines whether the accompanying DO statements are performed if a specific statement in the output is satisfied.

Additional information	Description
Usage	Optional
Format	Select **Specific statement output** from the **When** drop-down list, and fill in the sub-parameters below.
Variable Name	None
Alternate names	▪ **Control-M/EM Utilities:**ON ▪ **Control-M/Server Utilities:**-on ▪ **e-Trigger:**on_do_statement ▪ **Control-M/EM API:**on_statement
Sub-parameters	
Statement	A character string, from 1 through 132 characters in length, containing a statement from the job script file The specified string can be a portion of the statement. Statement character strings can each contain mask characters. Valid mask characters are: ▪ ***** – represents any number of characters (including no characters) ▪ **$** – represents any single character ▪ **?** – represents any single character
Code	A character string, from 1 through 255 characters in length, to be compared to the operating system's response to the specified statement. Code character strings can each contain mask characters. Valid mask characters are: ▪ ***** – represents any number of characters (including no characters) ▪ **$** – represents any single character ▪ **?** – represents any single character

Additional information	Description
Alternate formats	
Control-M/EM Utilities	ON is composed of the STMT, CODE, PGMS, PROCS, and AND_OR sub-parameters. STMT is a **1-132** character string, used only when you are specifying an On statement value.
Control-M/EM API	on_statement is composed of the following sub-parameters: ▪ and_or ▪ code ▪ procedure_step ▪ program_step ▪ statement

NOTE: Chinese, Japanese and Korean character sets: The **?** and **$** wildcards are not supported. The ***** wildcard is supported only as a standalone value or as a suffix to another value. It cannot be embedded in, or act as a prefix to another value.

Utilization of exit codes

Both DOS **.bat** scripts and REXX **.cmd** scripts can return an exit code to Control-M/Server upon completion. The_exit script utility is used by **.bat** scripts. For more information about script utilities, see Definition, ordering, and monitoring in *Control-M Workload Automation Utilities.*

Control-M/Server can distinguish between exit codes by using the following expression in the Code sub-parameter of the On Statement/Code job processing parameter:

```
COMPSTAT=<value>
```

EXAMPLE: Using exit codes

In this example, a REXX script exits with an exit code of 5, as displayed below:

```
exit 5
```

This condition can be detected by defining the following On Statement/Code parameter:

```
Stmt: *
Code: COMPSTAT=5
```

EXAMPLE: Job ends with a status of NOTOK

Using the following commands, any completion code other than 2 causes the job to end with a status of **NOTOK**.

```
On
Statement *
```

```
Code  COMPSTAT!2
Do NOTOK
```

EXAMPLE: Job ends with a status of OK

Using the following commands, any even completion code causes the job to end with a status of **OK**.

```
On
Statement *
Code COMPSTAT EQ EVEN
Do OK
```

Example: Create a condition on NOTOK

Using the following commands, a completion status of **NOTOK** causes Control-M to create a condition.

```
On
Statement *
Code NOTOK
Do Cond PRKZ_NOTOK   ODAT    +
```

EXAMPLE: Output from an OpenVMS job

Assume that the following messages are issued from an OpenVMS job:

```
$ SET NOON
$ MOU/SYS/OVER=ID MUA0:
$ COPY JWINFO_2507.DAT mua0:ACCOUNT.DAT
%COPY-E-OPENIN, error opening
          J$DSK:[USR1]JWINFO2507.DAT; as input
-RMS-E-FNF, file not found
$SH
%DCL-W-INSFPRM, missing command parameters – supply
          all required parameters
USER1        Job terminated at 10-AUG.-1997 09:01:42.07
Accounting information:
Buffered I/O count:43Peak working set size:375
Direct I/O count:24Peak page file size:2485
Page faults:495Mounted volumes:0
Charged CPU time:0 00:00:00.61Elapsed time:0 00:00:02.63
```

The following On statement would be triggered by the preceding messages:

```
On
```

```
Statement COPY JWINFO_$$$$.*
Code %COPY-E-OPENIN, error opening J$DSK:[USR1]*
Do NOTOK
```

EXAMPLE: Output from a UNIX Job

Assume that the following messages are issued from a UNIX job:

```
+ date
Mon Mar 20 10:30:58 IST 2006
+ pwd
/export/users/ctmagent
+ ls
BMCINSTALL
ctm
installed-versions.txt
lsagent.sh
```

The following On statement would be triggered by the preceding messages:

```
ON
Statement ls
Code lsagent*
Do NOTOK
```

EXAMPLE: Rerunning a job (for Microsoft Windows)

Assume that the following is the Output from a batch job:

```
copy job411.dat  tempt.dat
File not found - JOB411.DAT
           0 file(s) copied
```

The following On statement would be triggered by the preceding job output:

```
On
Statement copy job411.dat
Code File not found
Do Rerun
```

EXAMPLE: Assign NOTOK status for an iSeries (AS/400) job

Assume that the following is the Output from an iSeries (AS/400) job:

```
*NOnE  Request 21/11/05 16:41:07  QWTSCSBJ  QSYS     02FF  TEST
CMTST   0009
Message . . . . :  -CALL PGM(CMTST)
```

```
CPD0170 Diagnostic 30  21/11/05 16:41:07  QCLCLCPR  QSYS    02FF
TEST   CMTST   009
Message . . . . : Program CMTST in library *LIBL not found.
Cause   . . . . . : The Program specified on the CALL command
cannot be found.
```

The following On statement would be triggered by the preceding job output:

```
On
Statement *CALL PGM (CMTST)*
Code *Program CMTST in library *LIBL not found.*
Do NOTOK
```

Issue a shout

In this example a shout is issued to the administrator if the word "error" is displayed anywhere in the Output.

```
On
Statement stmt= * code= *error*
Do Shout To= Adman Urgn= High Msg= Potential error in Job %%jobid
```

EXAMPLE: Using completion codes other than 2

A completion code other than 2 causes the job to end NOTOK.

```
On
Statement stmt= * code= COMPSTAT !2
Do NOTOK
```

EXAMPLE: Using even completion codes causes the job to end OK

Any even completion code causes the job to end OK.

```
On
Statement stmt= * code= COMPSTAT EQ Even
Do OK
```

EXAMPLE: Trigger a second job after the first job ends NOTOK

If a job ends NOTOK, add a condition that triggers a job that now becomes necessary.

```
On
Statement stmt= * code= NOTOK
Do Condition Name=JobX_NotOK Date= odat sign= +
```

EXAMPLE: Set the status of a job to NOTOK under certain conditions

Set the status of a job to NOTOK if the Output indicates that the password directory is not found, In such a case, the Output of the UNIX job contains text similar to the following:

```
cp  /etc/passwd /tmp
```

```
cp  /etc/passwdx /tmp
cp: /etc/passwdx: No such file or directory
cp  /etc/passwd /usr/kvm/xyzzy
cp: /usr/kvm/xyzzy: Permission denied
exit 12
On
Statement stmt= cp/etc/passwdx/tmp code= cp*no*file
Do NOTOK
```

EXAMPLE: Rerun a job under certain conditions

Rerun the job if the Output indicates that a particular file to be copied was not found:

```
On
Statement stmt= copy job411.dat code= File not found
Do Rerun
```

Out Condition

(From Forecast only) Specifies prerequisite conditions to be added or deleted after the job completes with a completion status of **OK**.

Additional information	Description
Usage	Optional
Format	The **Out Condition** parameter is composed of the **Name**, **Date**, and **Sign** sub-parameters. These sub-parameters are described below.
Variable Support	Yes. Variable system variables (but not other types of variables) can be specified as the entire value for this parameter.
Alternate names	▪ **Control-M/EM Utilities:** OUTCOND ▪ **Reporting Facility:** CONDITION ▪ **Control-M/Server Utilities:** -outcond ▪ **Control-M for z/OS:** OUT ▪ **Control-M/EM API:** out_condition
Add/Remove the condition	Indicates whether the specified condition is to be added (created) or deleted. ▪ **+** Adds (creates) the prerequisite condition. Default. ▪ **–** Deletes the prerequisite condition. If the value of ODAT is $$$$ or ****, + cannot be selected.

The **In Conditions** parameter makes the submission of the job dependent on the existence of one or more prerequisite conditions.

NOTE: A maximum of 99 prerequisite conditions can be specified for the Out Conditions parameter.

If the job completion status is **OK**, then, according to the option specified, the prerequisite conditions are added to or deleted from the Conditions/Resources table. The Out Conditions parameter is performed before the Add/Remove Condition (on page 152) parameter. Therefore, the **Out Conditions** parameter can be overridden by those of the **Add/Remove Condition** parameter. For examples, see Add/Remove Condition (on page 152).

Prerequisite conditions

A prerequisite condition is a user-defined entity whose existence can be tested to determine whether a job should be submitted for execution.

You can specify that a prerequisite condition be added (that is, created) or deleted as part of the post-processing treatment of a job (using the **Out Condition** parameter and the **Add/Remove Condition** parameter).

Prerequisite conditions are used to define and implement job execution dependencies. A job containing an In Condition definition is not submitted for execution unless the specified In conditions exist. This permits job-to-job dependencies or job dependencies based on successful completion of a manual task (such as a file creation or a restore operation).

When created, each prerequisite condition is associated with a specific date. A date is also specified when testing for the existence of a prerequisite condition. Thus, the submission of a job can be made contingent upon the existence of a prerequisite condition created on a specific date. This enables you to specify jobs, for example, that depend on conditions created on the same day and that ignore conditions created on previous days.

A prerequisite condition can represent any user-specified situation. The following represent the kind of self-explanatory conditions for which a job may need to test:

- JOB-EJGH12-FINISHED
- SALARY-INPUT-READY
- CHECKS-PUNCHED
- WEEKEND
- COMM-ACTIVE

Time stamp

NOTE: Not relevant in z/OS environments.

Inserting the **@HHMMSS** code into an Out condition name includes a time stamp (in hours, minutes, and seconds) that is resolved to the time that the job is entered in the Active Jobs database.

Future and past dates

You can use the Days Offset field to set a future or past date for an Out condition. Valid values for this field are a + (plus) or - (minus) sign, followed by a number from 0 through 999. The value in the field represents the number of days in the future, or in the past, relative to the actual order date.

For examples of prerequisite conditions, see In Condition (on page 125).

Action

Indicates how the job's log file (Output) should be handled after the job ends with a status of **OK**.

Additional information	Description	
Usage	Optional	
Format	Drop-down list. Valid values: ▪ Change job class (z/OS only) ▪ Delete ▪ Copy ▪ Move ▪ Print For most of these options, a second field is displayed.	
	The format of the second field is as follows:	
	Length	**Computers other than z/OS:** Up to **255** characters. **z/OS:** Up to 44 characters, as follows: ▪ File Name (Copy): up to 44 characters ▪ New Class Name (Change job class): 1 character ▪ New Destination (Move): up to 8 characters
	Case sensitive	Yes
	Invalid Characters	Blanks
	Variable Support	Yes. A variable or expression can be specified as all or part of the value for this parameter.

<table>
<tr><th>Additional information</th><th colspan="2">Description</th></tr>
<tr><td>Alternate names</td><td colspan="2">▪ Control-M/EM Utilities: NOTIFICATION
▪ Reporting Facility:
Output Handling is composed of two sub parameters
▪ SYSOPT Equivalent to the Output handling text box.
▪ PARM Equivalent to the variously-named text boxes that are displayed when specific Output Handling options are chosen.
▪ Control-M/Server Utilities: -OUTPUT
▪ Control-M for z/OS: OUTPUT
▪ Control-M/EM API: Output handling is composed of three sub-parameters:
▪ output_from_class Equivalent to From Class
▪ output_option Equivalent to Option.
▪ output_parameter Equivalent to Parameter.</td></tr>
<tr><td colspan="3">Alternate Formats</td></tr>
<tr><td>Reporting Facility</td><td colspan="2">The values for SYSOPT and PARM are strings.</td></tr>
<tr><td rowspan="2">Control-M/Server Utilities</td><td colspan="2">Two values are supplied for -OUTPUT:</td></tr>
<tr><td><option></td><td>▪ RELEASE
▪ DELETE
▪ COPY
▪ MOVE</td></tr>
</table>

Additional information	Description	
	<parameter>	The appropriate value, depending on the **<option>** value specified. String.
Control-M for z/OS	Output is composed of the following sub-parameters:	
	OP	Type of Output handling to perform. Mandatory. Valid values are: ▪ **C** – Change the class of the job output. [z/OS, only.] ▪ **D** – Delete the job output. ▪ **F** – Copy the job output to file. ▪ **N** – Change the destination of the job output. ▪ **R** – Release the job output.
	data	Relevant Output data. Mandatory and valid only if the specified OP value is **F**, **C**, or **N**. Valid values depend on the OP value: ▪ **F** – File name. String comprised of from 1 through 44 characters. All characters are valid except blanks. ▪ **C** – New class (1 character). Any character is valid except blank, but an asterisk (*****) indicates the original MSGCLASS of the job. ▪ **N** – New destination (1 through 8 characters). All characters are valid except blanks.
	FRM	1-character value that identifies the class of job output to process. Selected when **C** is the value of OPT. [z/OS, only.] Optional. If a **FRM** class is not specified, all Output classes are treated as a single, whole unit.

If no Output handling is specified (or the job does not end **OK**), and no Handle Output (on page 164) statement (in the On Statement/Code parameters) is activated, the job's log file is placed in the location determined by Control-M until it removed by the New Day procedure.

NOTE: The Control-M **Automatic Log Copy** system parameter is not affected in any way by specified Output Handling

Some Output Handling options (those that are selected from the Output Handling list) require you to supply additional input. When such an option is selected from the Actions Tab, an additional text box is be displayed. The label for this text box is different for each Output Handling option that is selected. When an additional text box is displayed, enter the appropriate information in it.

Output Options

Value	Description
Delete	Deletes the log file.
Copy	Copies the log file. Outputs should not be copied to a Control-M internal directory or subdirectory (for example, a subdirectory underneath the Output directory). Specify the log file name and full path to which the log file must be copied in the **File Name** text box. The following can occur: **Log file name** **Full path** **Remark** Specified Specified The log file is copied as specified in the **File Name** text box. Specified Not specified The log file is copied to the job owner's home directory using the file name specified. For jobs executed on remote hosts, the log file is copied to the agent home directory through which the job was submitted. However, the file is not copied if WMI protocol is used. Not specified Specified The log file is copied to a default file name determined according to the considerations below, to the full path specified. Not specified Not specified The log file is copied to the job owner's home directory using the default file name determined according to the considerations below. For jobs executed on remote hosts, the log file is copied to the agent home directory through which the job was submitted.
	File name considerations

Value	Description
	For Microsoft Windows: **<Mem_Name>|<Job_Name>_<order_num>_<rerun_num>.DAT** For OpenVMS: **<Mem Name>.TXT** For UNIX: **<Mem_Name>|<Job_Name>.LOG<order_num>.<rerun_num>** For OS/2, the full path name must be specified.
	Special notes for iSeries (AS/400) computers: This second field is mandatory. It indicates the database file to which the job log is copied. The file must be specified in one of the following formats: **library/file** ***LIBL/file** **file** If the specified file does not exist, it is created. The job log is placed in the file as member **CM<AS/400_Job Number>** (the first line of the member contains details that identify the job). If the specified file already exists, and its record length is 132 or more, the job log is placed in the file as a new member identified as above. If the specified file already exists, and its record length is less than 132, the job log is truncated and placed in the file as a new member identified as above.

Value	Description
Print	Releases the log file for printing on the specified printer/output queue. A second field is displayed for the printer queue to which the log file should be released. The default printer/output queue and the Output directory are determined by Control-M system parameters on each computer. For more information, see your Control-M administrator. Special notes for iSeries (AS/400) computers: If an output queue is not specified, the output of the job is sent to the queue specified by the Control-M **Default Output Queue** system parameter. The name of the output queue must be specified in one of the following formats: **library/outq** ***LIBL/outq** **outq** If **outq** (without library) or ***LIBL/outq** is specified, the specified outq is taken from the ***LIBL** (Library list) of Control-M. If the specified outq is not found in the Library list, the Release option is not executed. If **library/outq** is specified, the output queue is taken from the specified library. If the specified **outq** is not found in the specified library, the Release option is not executed.
Move	Moves the log file and deletes it from the Control-M Output directory. The **New Destination** text box is displayed for the location to which the log file should be moved. Outputs should not be moved to a Control-M internal directory or subdirectory (for example, a subdirectory underneath the OUTPUT directory). Specify the log file name and full path to which the log file must be moved in the **File Name** text box. The following can occur: **Log file name** **Full path** **Remark** Specified Specified The log file is copied as specified in the **File Name** text box. Specified Not specified The log file is copied to the job owner's home directory using the file name specified. For jobs executed on

Value	Description
	remote hosts, the log file is copied to the agent home directory through which the job was submitted. However, the file is not copied if WMI protocol is used. Not specified Specified The log file is copied to a default file name determined according to the considerations below, to the full path specified. Not specified Not specified The log file is copied to the job owner's home directory using the default file name determined according to the considerations below. For jobs executed on remote hosts, the log file is copied to the agent home directory through which the job was submitted
	File name considerations For Microsoft Windows: **<Mem_Name>|<Job_Name>_<order_num>_<rerun_num>.DAT** For OpenVMS: **<Mem Name>.TXT** For UNIX: **<Mem_Name>|<Job_Name>.LOG<order_num>.<rerun_num>** For OS/2, the full path name must be specified.

Value	Description
Move (*continued*)	**File name considerations (*continued*)** For iSeries (AS/400) and UNISYS computers: The second field indicates the **outq** (output queue) to which the job log spool file should be moved. If an output queue is not specified, the job log spool file is moved to the queue specified by the Control-M Default Output Queue system parameter. The name of the output queue must be specified in one of the following formats: library/outq LIBL/outq outq If **outq** (without library) or ***LIBL/outq** is specified, the specified outq is taken from the ***LIBL** (Library list) of Control-M. If the specified **outq** is not found in the Library list, the **Move** option is not executed. If **library/outq** is specified, the output queue is taken from the specified library. If the specified **outq** is not found in the specified library, the Move option is not executed.
Change Jobs Class	Changes the class of job output. [z/OS only] The **New Class Name** text box and **From Class** text box are displayed. Enter the new class name (1 character) in the text box (Mandatory). An asterisk (*) indicates the job's original MSGCLASS. Optionally, you can specify a class in the **From Class** text box. If a class is specified, Output Handling is limited to only Outputs from the specified class.

All of the following examples presume a job completion status of **OK**.

EXAMPLE: Release the Log file to the default printer

Output Handling

Option: Print

EXAMPLE: [iSeries (AS/400)] Release the Log file to the specified output queue

Output Handling

Option: Release Parm: MYLIB/MYOUTQ

EXAMPLE: OpenVMS and UNIX: Move the Log file

Move the log file to a file called **test.log** in the job owner's home directory.

Output Handling

Option: Move Parm: test.log

EXAMPLE: UNIX: Copy the Log file to a directory

Copy the log file to a directory called **prg2/test/**. Use the default file name.

Output Handling

Option: Copy Parm: prg2/test/

EXAMPLE: UNIX: Copy the Log file to a file

Copy the log file to a file. The name of the log file is determined by the Job Name.

Output Handling

Option: Copy Parm: prg2/%%JOBNAME

Retention Days

Determines the number of days to retain the job in the History Jobs file. For z/OS jobs only.

Additional information	Description
Usage	Optional
Format	001 - 999 days. **NOTE:** When the Retention Days field is left blank, History Jobs file information is kept as specified by the Retention Generation parameter. For more information, see Retention Generations (on page 211).
Variable Support	None
Alternate names	▪ **Control-M/EM Utilities:** RETEN_DAYS ▪ **Reporting Facility:** RETEN DAYS ▪ **Control-M/Server Utilities:** (none) ▪ **Control-M for z/OS:** RETENTION: # OF DAYS TO KEEP ▪ **Control-M/EM API:** reten_days

Jobs in the History Jobs file are easier to restore to the Active Jobs database (for example, for restart) than jobs archived to CDAM. Therefore, it may be desirable to retain a job in the History Jobs file for a period of time.

Using Retention Days you can specify a fixed number of days to keep the job in the History Jobs file. Once the specified number of days is reached, the job is automatically deleted from the History Jobs file during the next New Day processing.

Retention Days and Retention Generations are mutually exclusive. A value can be specified for either, but not both.

When changing job criteria from Retention Days to Retention Generations (or the reverse), previous job criteria are lost and are not acted upon.

For retention criteria to hold across job executions, the jobs must be identical in all respects. (For example, if a job is transferred to a different folder, it is treated as a different job for purposes of retention. In this case, retention values are reset, and retention is calculated from the moment of transfer.)

NOTE: At sites that do not use the History Jobs file, this parameter is not relevant and is not displayed.

Retention Generations

Maximum number of generations of the job to keep in the History Jobs file. For z/OS jobs, only.

Additional information	Description
Usage	Optional.
Format	00 - 99 generations. **NOTE:** When the Retention Generations field is left blank, History Jobs file information is kept as specified by the Retention Days parameter. For more information, see Retention Days (on page 210).
Variable Support	None
Alternate names	▪ **Control-M/EM Utilities:** RETEN_GEN ▪ **Reporting Facility:** RETEN GEN ▪ **Control-M/Server Utilities:** (none) ▪ **Control-M for z/OS:** RETENTION: # OF GENERATIONS TO KEEP ▪ **Control-M/EM API:** reten_gen

Jobs in the History Jobs file are easier to restore to the Active Jobs database (for example, for restart) than jobs archived to CDAM. Therefore, it may be desirable to retain several of the most current generations of the job in the History Jobs file.

Retention Generations enables specification of the number of generations of the job to keep in the History Jobs file. Once the specified number of generations has been reached, as a new generation is added to the History Jobs file, the earliest remaining generation is deleted.

Retention Days and Retention Generations are mutually exclusive. A value can be specified for either, but not both.

When changing job criteria from Retention Days to Retention Generations (or the reverse), previous job criteria are lost and are not acted upon.

For retention criteria to hold across job executions, the jobs must be identical in all respects. (For example, if a job is transferred to a different folder it is treated as a different job for purposes of retention. In this case, retention values are reset, and retention is calculated from the moment of transfer.)

At sites that do not use the History Jobs file, this parameter is not relevant and is not displayed.

Pre-Notifications

The Pre-Notifications parameter gives you the ability to create a message that is sent to one or more specified destinations when certain conditions are encountered.

Additional information	Description
Usage	Optional
Format	The Pre-Notifications parameter consists of the sub-parameters described below
Variable Support	Yes. A variable or expression can be specified as all or part of the value for this parameter.
Alternate names	▪ **Control-M/EM Utilities:** NOTIFICATION ▪ **Reporting Facility:** (none) ▪ **Control-M/Server Utilities:** -notification ▪ **Control-M for z/OS:** NOTIFICATION ▪ **Control-M/EM API:** notification
Sub-parameters	

Additional information	Description
Job's average execution time	Send the message if the job's elapsed runtime is outside a specified limit. The limit is specified in the Time field. A limit can be expressed as the actual elapsed minutes of the job run, or as a deviation from the job's (statistical) average runtime. The limit for **Job's average execution time** can be expressed in any of the following formats: ▪ **Greater than**: The message is sent if the elapsed runtime of the job is greater than n minutes. in z/OS environments, n is a number from 0 through 999. On other computers, n is a number from 1 through 999. ▪ **Less than**: The message is sent if the elapsed runtime of the job is less than n minutes. n is a number from 1 through 999. ▪ **Exceeds average time**: The message is sent if the elapsed runtime of the job exceeds its average execution time by at least n minutes. n is a number from 1 through 999. ▪ **Percent**: The message is sent if the elapsed runtime of the job exceeds/less than its average execution time by at least n%. n is a number from 1 through 900. ▪ **Less than average time**: The message is sent if the elapsed runtime of the job is at least n minutes less than its average execution time. n is a number from 1 through 999. **NOTE:** A job processing definition can contain more than one notification parameter with a When of **Job's average execution time**. For more information, see examples of the use of **Job's average execution time** below.

Additional information	Description
Job not submitted by time	Send the message if the job is still not submitted and cannot be submitted at the time specified in the accompanying Parm (time) field. ▪ Specify the time in hhmm format. **z/OS**: ▪ Specific Time: Specify the time in hhmm format. ▪ Job's calculated time ▪ Days:Sets the number of days relative to the ODAT by which the sending of the Notification message is offset. Valid values: a number from 0 through 120 blank – no offset. If the Parm (time) value is *, leave this field blank. In the following discussion, time is formatted according to international standard notation. (In Control-M for z/OS, you can alternatively specify an * for the time; this results in use of the job's calculated DUE IN time to determine if the job was not submitted on time.) By default, a valid time must be in the range of 00:00 through 23:59. However, if Time Synonym (in File/Options>Job Properties) is checked, a valid time must be in the range from NewDay time until NewDay Time-plus-23:59. For example, if New Day time is 8:00 A.M., the range of valid times is from 08:00 until 31:59. **NOTE:** When specifying the time for this parameter, use the hhmm (24-hour time) format. The notification message is sent only if the job is actually not submittable. That is, if a runtime criterion (in condition, quantitative resource, and so on) is not met at the given time. However, if a job is ordered after the specified time but starts running immediately because it meets all runtime criteria, the **Job not submitted by time** parameter doesn't apply

Additional information	Description
Job not finished by time	Send the message if the job does not finish executing by the time specified in the accompanying Parameter (time) field. Specify the time in hhmm format. ▪ Specify the time in hhmm format. **z/OS**: ▪ Specific Time: Specify the time in hhmm format. ▪ Job's calculated time ▪ Days:Sets the number of days relative to the ODAT by which the sending of the Notification message is offset. Valid values: a number from 0 through 120 blank – no offset. If the Parm (time) value is *, leave this field blank. In the following discussion, time is formatted according to international standard notation. (In Control-M for z/OS, you can alternatively specify an * for the time; this results in use of the job's calculated DUE OUT time to determine if the job is late.) By default, a valid time must be in the range of 00:00 through 23:59. However, if Time Synonym (in File/Options>Job Properties) is checked, a valid time must be in the range from New Day time until New Day Time-plus-23:59. For example, if New Day time is 8:00 A.M., the range of valid times is from 08:00 until 31:59. **NOTE:** When specifying the time for this parameter, use the hhmm (24-hour time) format. No message is sent if the job is being rerun.
Job set to rerun	Send the message if the job's completion status was set to Rerun (not valid for SMART Folders).
Destination	Logical destination of the notification message. Mandatory. Destination is the name of an entry in the Shout Destination table.

<table>
<tr><th>Additional information</th><th colspan="2">Description</th></tr>
<tr><td></td><td colspan="2">Computers other than z/OS:
Valid values
▪ a user logged onto the Control-M installation
▪ a user's mail in the Control-M installation
▪ a specific terminal
▪ system console
Alerts window in all Control-M/EM workstations connected to the specified Control-M installation
NOTE: The Control-M/EM workstations to which an alert is issued may be limited by user authorizations. For more information, see Control-M security in Control-M Workload Automation Administration.
Control-M log
All notification messages are recorded in the Control-M log. Select the log as a destination only when you do not wish to send the message to an additional destination.</td></tr>
<tr><td></td><td>Length</td><td>1 through 16 characters</td></tr>
<tr><td></td><td>Case Sensitive</td><td>Yes</td></tr>
<tr><td></td><td>Invalid Characters</td><td>Blanks; single quotation marks</td></tr>
<tr><td></td><td>Variable Support</td><td>Yes. A variable or expression can be specified as all or part of the value for this parameter.</td></tr>
<tr><td></td><td colspan="2">z/OS:
The following are valid values for Destination:</td></tr>
<tr><td></td><td>U-userid</td><td>Writes the message to the IOA Log file
userid is a user ID consisting of 1 through 8 characters but containing no blanks</td></tr>
</table>

Additional information	Description	
	OPER [-n]	Sends a scrollable message to the operator console n is an optional up to 3-digit route code If a route code is not specified, the default routes are Master Console and Programmer Information (1 and 11). For more information regarding route codes, refer to the IBM publication Routing and Descriptor Codes, GC38-1102.
	OPER2[-n]	Sends an unscrollable message to the operator console n is an optional up to 3-digit route code If a route code is not specified, the default routes are Master Console and Programmer Information (1 and 11). For more information regarding route codes, refer to the IBM publication Routing and Descriptor Codes, GC38-1102.
	TSO - logonid or T - logonid	Where logonid is one of the following: ▪ a valid logon identity consisting of 1 through 7 characters ▪ a valid group identity found within the IOA Dynamic Destination Table An optional second value, indicating the computer or node of the TSO logonid, can be specified, as follows: Under JES2: ▪ ;Nn, ;Mm or ;NnMm, where — m is the ID of the computer in JES2 (not the 4-character SMF system ID) — n is the 1- to 2-character JES/NJE node ID Under JES3: ▪ Lname, where Lname is the logical JES name of the computer (that is, the name as used in the JES3 command *T, not the SMF system ID)
		A notification to a TSO user preforms a TSO SEND command, which may require authorization at the receiving end.

Additional information	Description	
	U-M: mail_ name_ prefix	Sends a message to the recipient identified by mail_name_prefix, a mail name prefix consisting of 1 through 12 characters (z/OS platform: 1 through 8 characters)
	U-EM	Sends a message that is displayed in the Control-M/EM Alerts window
Urgency	Urgency of the notification message directed to the Alerts window. Urgency assigned to the notification message affects the appearance of the message in the **Alerts** window. Urgency levels are: ▪ **R** - Regular (Default) ▪ **U** - Urgent ▪ **V** - Very Urgent	
Message	Text of the notification message. ▪ **Length**:1 through 255 characters z/OS: 1 through 70 characters ▪ **Case Sensitive**: Yes ▪ **Invalid Characters**: None ▪ **Variable Support**: Yes. The message can include variables, including any combination of text, Control-M system variables, job submission variables and User variables (created using the Variable Assignment parameter). However, the length of the message after decoding cannot exceed the length specified above. For more information, see Control-M Variable facility (on page 229).	

A "notification message" is a message sent to one or more destinations when the condition specified by the When parameter is satisfied.

The Notify parameter (described in Notify (on page 156)) can be used to issue notification messages conditioned by an On Statement/Code parameter.

Notification messages can also be issued from the Control-M/Server using the utility. For more information, see ctmshout in *Control-M Utilities*.

When **Job's average execution time** values are specified with a **+** or **–** sign (that is, when elapsed runtime is compared to average runtime), the notification applies only if there are current statistics data for the job (containing statistics for at least one of the last 20 runs of the job).

If current job statistics data exists, all available elapsed-time statistics for the last 20 job runs are averaged to generate the average runtime, and the current runtime is compared to this figure according to the specified criteria.

If no job statistics data exist, or if the data is not current (there are no elapsed-time statistics for any of the last 20 job runs), the notification parameter is not applied.

More about Job's average execution time

The following additional considerations apply to the use of **Job's average execution time**:

- When **Job's average execution time** values are negative (for example, -n; -n%), the check can be performed only after the job has finished running.
- When **Job's average execution time** values are positive (for example, +n; +n%), the check can be performed (and if the elapsed runtime limits are exceeded, the message can be sent) before the job has finished running.
- Relative **Job's average execution time** limits should not exceed 24 hours. When relative **Job's average execution time** limits exceed 24 hours (that is, if +n(%) of the average runtime exceeds 24 hours), the message is sent if and when processing reaches 24 hours.

If a relative **Job's average execution time** is not specified prior to job submission, but is specified afterwards (for example, the job is Held, the parameters changed, and the job then Freed), the **Job's average execution time** value is ignored.

More about destinations

When the destination is a user, the message is also sent to the user's mail.

If not found in the Shout Destination table, the destination is assumed to be a user name. In this instance, the notification message is sent to the user's terminal and the user's mail.

The Shout Destination table is maintained by the Control-M administrator. Several such tables may exist. Each table contains the same logical destinations, but the physical (actual) destinations can vary from table to table.

Only one table is in use ("active") at any time, as determined by the administrator. A notification message sent to a logical destination is directed by Control-M to the corresponding physical destination listed in the active Shout Destination table.

For z/OS jobs

Sends notification message to a specified user when job ends OK

Notification Statements:

- When OK
- Destination SHIFTMNGR
- Urgency R
- Message NIGHTSHIFT RUN COMPLETED
- The message is sent to Control-M logical destination SHIFTMNGR.
- Sends notification message to Control-M/EM when job terminates NOTOK
- If the job terminates with a status of **NOTOK**, send a message to Control-M/EM, indicating the completion code:

Notification Statements:

- When NOTOK
- Destination EM
- Urgency U
- Message Job NOTOK – Completion code=%%COMPSTAT
- The message is sent to the Control-M/EM Alerts window.

Sends Notification message when job runtime is less than expected

Given that a job whose average run time is 50 minutes completes in 40 minutes. The job processing definition contains a notification statement with When **Job's average execution time** . The following table indicates which Time parameter values would and which would not cause a notification message to be issued:

Notification Message Issued	Notification Message Not Issue
>39	>40
<41	<40
-10	-11
-20%	-21%

No notification message although job runtime is more than expected

Given that a job whose average run time is 50 minutes completes in 90 minutes. The job processing definition contains a notification statement with When **Job's average execution time.** The following table indicates which Time parameter values would and which would not cause a notification message to be issued:

Notification Message Issued	Notification Message Not Issue
>89	>90
<91	<90
+41	+40
+79%	+80%

Final Notifications

Sends a notification if a job terminates with a completion status of OK or NOTOK.

Status	Description
OK	Job ended **OK**. The job finished executing with a successful operating system completion status. This status can be overridden with the End Job Not Ok (on page 160) parameter (as a result of an On Statement/Code evaluation).
NOTOK	Job did not end **OK**. The job finished executing with an unsuccessful operating system completion status or due to a submission failure (for example, queue does not exist). This status can be overridden with the End Job Ok (on page 161) parameter (as a result of an On Statement/Code evaluation).

Status parameters

Status parameters describe information that is collected as the result of a job run. They have the following common characteristics:

- These parameters are accessible only from the active jobs database.
- Most of the values for these parameters are supplied by Control-M, and many cannot be modified by the user.
- Most parameters are displayed in Control-M in the Monitoring domain of the Properties pane.
- These parameters can be used as filter criteria for selecting job definitions using the Control-M utilities and the Reporting facility.
- Their values can be displayed in reports generated by the Reporting facility

status parameters are summarized in the following table:

Parameter	Description	
Average Run Time	Average time taken, in minutes, for the job to run. This statistic is compiled from the last successful runs of the job.	
Current Status	Indicates the completion status of the job (for example, **Ended Not OK**).	
Deleted	Indicates if the job was deleted.	
Due In	Time and day at which the next run of the job should start executing.	
	+ *num* **days**	The number of days that the start of job execution is extended after the ODAT. This sub-parameter is relevant only for jobs running under Control-M for z/OS version 6.2.00 or later.
Due Out	Time and day by which the next run of the job should stop executing.	
	Format	HHMM, where HH is a 2-digit number from 00 through 24.

Parameter	Description	
	+ *num* **days**	The number of days that the end of job execution is extended after the ODAT. This sub-parameter is relevant only for jobs running under Control-M for z/OS version 6.2.00 or later.
Elapse	Length of time (in minutes) that the job is expected to run.	
End Time	Indicates the time that the last run of the job ended.	
From Proc	For z/OS jobs only with Control-M/Restart: Procedure step from which the job should be rerun.	
From Step	For z/OS jobs only with Control-M/Restart: Step from which the job should be rerun.	
Hold	Indicates if the job is currently being held.	
Job ID	Unique serial number assigned to the job by the Control-M server.	
Next Time	Determines the next time that the job runs for reruns or cyclic jobs that use the Interval option.	
NJE	Indicates that the job was sent for execution to a computer that is connected to Control-M through NJE (the node does not have a shared spool with Control-M).	
NJE Node	Node ID of the NJE terminal.	
Rerun Counter	Indicates how many times the job was run.	
Restart	Indicates if the job was restarted.	
Search Count	Number of times Control-M has looked for the job.	
Standard Deviation	Standard deviation from the average runtime. This statistic is compiled from the last successful runs of the job.	
Start Time	Indicates the time that the last run of the job started.	
To Proc	For z/OS jobs only with Control-M/Restart: Procedure step until which the job should be rerun.	
To Step	For z/OS jobs only with Control-M/Restart: Step until which the job should be rerun.	

Microsoft Windows parameters overview

Microsoft Windows parameters are for the WIN panel of the Properties Pane. These parameters are only relevant if Control-M/Agent Microsoft Windows version 6.0.0*x* or later is installed in your Control-M/Server.

Parameter	Description
Job Execution Time (on page 225)	Maximum amount of elapsed execution time, in seconds, for the job.
Job Memory (on page 225)	Maximum amount of memory, in megabytes, allowed for the job.
Max Work Set Size (on page 225)	Maximum RAM, in megabytes, for all processes of the job.
Min Work Set Size (on page 226)	Minimum RAM, in megabytes, for all processes of the job.
Priority Class (on page 226)	Highest priority class the job and its "children" can receive.
Process Execution Time (on page 227)	Maximum amount of elapsed execution time, in seconds, for each process in a job.
Process Memory (on page 227)	Maximum amount of memory, in megabytes, allowed for each process in a job.
Scheduling Class (on page 228)	Scheduling class for all processes of a job.

NOTE: The job fails if the maximum value set for a parameter is exceeded. The OSCOMPSTAT will equal -3.

Job Execution Time

Maximum amount of elapsed execution time, in seconds, for the job.

Additional information	Description
Usage	Optional
Format	CPU time, in seconds. Min: 0.1 Max: 1.8 x 1012
Variable Name	%%WIN2K-PER_JOB_USER_TIME_LIMIT

NOTE: BMC Software recommends entering application data using the required panel in the Properties Pane.

Job Memory

Maximum amount of memory, in megabytes, allowed for the job.

Additional information	Description
Usage	Optional
Format	Memory, in megabytes. Min: **0.1** Max: **4200.0**
Variable Name	%%WIN2K-JOB_MEMORY_LIMIT

Max Work Set Size

Maximum RAM, in megabytes, for all processes of the job.

Additional information	Description
Usage	Optional.
Format	RAM, in megabytes. Min.: **0.1** Max.: **4200.0**
Variable Name	%%WIN2K-MAXIMUM_WORKING_SET_SIZE

Min Work Set Size

Minimum RAM, in megabytes, for all processes of the job.

Additional information	Description
Usage	Optional
Format	RAM, in megabytes. Min.: **0.1** Max.: **4200.0**
Variable Name	%%WIN2K-MINIMUM_WORKING_SET_SIZE

Priority Class

Highest priority class the job and its children can receive.

Additional information	Description
Usage	Optional
Format	List box. Valid values: ▪ **IDLE_PRIORITY_CLASS** ▪ **BELOW_NORMAL_PRIORITY_CLASS** ▪ **NORMAL_PRIORITY_CLASS** ▪ **ABOVE_NORMAL_PRIORITY_CLASS** ▪ **HIGH_PRIORITY_CLASS** ▪ **REALTIME_PRIORITY_CLASS**
Variable Name	%%WIN2K-PRIORITY_CLASS

The calling process must enable the SE_INC_BASE_PRIORITY_NAME privilege so that the calling process is allowed to boost the scheduling priority of a process.

Process Execution Time

Maximum amount of elapsed execution time, in seconds, for each process in a job.

Additional information	Description
Usage	Optional
Format	CPU time, in seconds. Minimum: **0.1** Maximum: **1.8 x 1012**
Variable Name	%%WIN2K-PER_PROCESS_USER_TIME_LIMIT

If the user-mode time for any process exceeds the specified amount, that process is terminated.

Process Memory

Maximum amount of memory, in megabytes, allowed for each process in a job.

Additional information	Description
Usage	Optional
Format	Memory, in megabytes. Minimum: **0.1** Maximum: **4200.0**
Variable Name	%%WIN2K-PROCESS_MEMORY_LIMIT

Scheduling Class

Scheduling class for all processes of a job.

Additional information	Description
Usage	Optional
Format	Valid values: **0** – **9**. ▪ **0** - provides the minimum resources ▪ **9 -** provides the maximum resources
Default	5
Variable Name	%%WIN2K-SCHEDULING_CLASS

To use a scheduling class greater than **5**, the calling process must enable the SE_INC_BASE_PRIORITY_NAME privilege. If the **Logon as user** option is turned off, no special change is needed. If the **Logon as user** option is turned on, **Increase Scheduling Priority** must be granted to the owner of the job.

7

Control-M Variable facility

The Variable facility consists of a group of special variables and functions that can be used to make your job processing definitions more dynamic. This facility also enables you to define your own variables.

Certain Variable functions and variables work differently for z/OS jobs. Although some of these differences are described in this chapter, BMC recommends that you consult the *Control-M for z/OS User Guide* for a more complete description of Variable functionality for z/OS jobs.

Using variables and functions, you can

- Access information about the system under which a job is running
- Pass information to a job or modify working parameters of the job
- Pass information in a Notification message or a Do Mail message to a user when the job ends

You cannot include application-specific job parameters in the values of variables. The names of application-specific job parameters are prefixed by two percent signs, the application's abbreviation and a hyphen (%%SAPR3- for SAP, %%OAP- for Oracle, and so on).

Variables in a job processing definition are resolved immediately before the job is submitted, in the order in which they appear in the job processing definition. If a variable is included in the notification, Notify (on page 156), Action (on page 202), and Handle Output (on page 164) of the job, the variables are re-evaluated and resolved when these actions are performed.

If a job is rerun, the variable statements specified in the **Variable Assignment** parameter are resolved before those specified using the Set Variable (on page 150) parameter.

For more information about variable types, see Variables (on page 36). For more information about variable functions, see Variable Expressions (on page 248).

Parameters that accept variables and expressions

Parameter	Description
Variable Assignment	Modifies working parameters for a job and/or passes parameters to a job when the job is submitted. Defines variables that can be displayed in a Notification message or Do Mail message when a job completes. **NOTE:** This parameter is called in certain Control-M/Server utilities and in Control-M Workload Automation.
Command	Variables can be used as part of the command string.
Do Variable	Enables the user to specify variable statements to be resolved (in addition to those specified for Variable Assignment) depending on how a job ended. These can be local variables to be used if the job is rerun, or they can be global variables to be used by other jobs.
Member library	Variables can be used in this parameter to indicate the name of the library or directory in which the file described in the parameter is located.
Member	Variables can be used in this parameter to indicate the name of the file containing the job commands or job script.
Override Path	Variables can be used in this parameter to provide the name of the override path in which to search for the file that was specified using the File name/Member name (on page 28) parameter.
Notification or Notify	Variables can be used in a Notification or Notify message to pass job parameters or other data to a user.
Output Handling or Do Output	Variables can be used in the field which provides supplementary information about the handling of the job's output (for example, a filename) when the job ends **OK**.

The **Variable Assignment** and Set Variable (on page 150) parameters can also use the **%%LIBMEMSYM** variable to reference a list of variable expressions in a separate text file. You can also use the variable **%%POOLSYM** to reference Name Pool variables. For more information, see Variable lists (on page 246).

Variables

All variables are identified by the **%%** prefix. If **%%** is included in the value for a job processing parameter, Control-M assumes that it is referring to a variable or function.

Additional information	Description
Usage	Optional
Format	▪ Name ▪ Value
Length	▪ Name: 1-40 ▪ Value: 1-4000 z/OS: ▪ Name: 1-66 ▪ Value: 1-66
Invalid Characters	None
Alternate Names	▪ **Control-M/EM Utilities:** VARIABLE ▪ **Reporting Facility**: VARIABLE ▪ **Control-M/Server Utilities:-**variable ▪ **Control-M/EM API:** variable

A special **%%#** prefix can be used to indicate that a variable or function should not be resolved. In these cases, the actual name of the variable or function (minus the **#** sign) is output. For example, **Do Notification Variable %%#PARM1 is greater than 100 i**ssues the following message:

Variable %%PARM1 is greater than 100.

Variables are divided into the following types:

- **Job Submission variables** pass parameters to a job or set the job's working parameters. For more information, see Job submission variables (on page 232).
- **System variables** are automatically assigned values using system information available at the time of job submission (for example, **%%DATE** contains the current system date). For more information, see Control-M system variables (on page 238).
- **User-defined variables** can be defined in a number of different ways for inclusion in various job processing parameters. For more information, see User-defined variables (on page 242).
- **Variable lists**. The %%LIBMEMSYM special variable can be used to point to a file containing a list of assignments statements to be applied to a job. This variable enables you to create one or more lists of assignment statements that can be applied to many job processing definitions. For more information, see Variable lists (on page 246).

Job submission variables

Job submission variables pass parameters to a job or set the job's working parameters.

Certain job submission variables are available only for certain computers. the tables below describe variables that are computer-specific.

All job submission variable names must be specified using uppercase letters.

General job submission variables

Name	Description	
%%DBGLVL	numeric	The agent process uses the First Failure Data Capture mechanism to write all the information collected and then sets the debug level to the value specified in the variable when processing the job. BMC recommends that when setting the %%DBGLVL variable, that you also set the First Failure Data Capture Reset interval (sec) parameter in the Control-M Configuration Manager to the value of 5. This causes the tracker process to stop writing debug information 5 seconds after having been invoked by the %%DBGLVL Variable instead of the default of 600 seconds.
%%OVERLIB	string	Override Path(name of an alternate library/directory in which job script is stored) of the job.

Name	Description
% %PARMn	Job submission parameter, where n represents the parameter number. This variable can be used to pass parameters to all types of jobs. On OpenVMS computers **% %PARMn** is used to represent the P1 through P8 parameters. For other computers, n can be a number between 1 and 32. **NOTE:** For Control-M version 2.1*x* on an iSeries (AS/400) computer, the value for variable **% %PARMn** must not include 'single quotation marks'. If quotation marks are necessary, use "double quotation marks" only.

OpenVMS job submission parameters

Name	Description
% %QUEUE	Name of the batch queue to which the job is submitted. For VMS cluster configurations: Specify only those queues that are defined in the system's queue manager where Control-M runs.

The variables described in the following table correspond to parameters of the iSeries (AS/400) Submit Job command. For more information about the valid values for these parameters, see your iSeries (AS/400) SBMJOB command documentation.

iSeries (AS/400) job submission parameters

Name	Description
%%ACGCDE	Accounting code for the job. Valid values are in the *<accountingCode>* format or one of the following special values: ***USRPRF**, ***USER**, ***JOBD**, ***NOCHG**, or ***NONE** **NOTE:** This variable is relevant only for Control-M/Agent for iSeries (AS/400) version 2.25 or later.
%%CURLIB	Name of the current library associated with the job being run.
%%DATE	Date that is assigned to the job when it is started.
%%HOLD	Whether the job is held at the time that it is put on the job queue.
%%INQMSGRPY	Manner in which pre-defined messages issued as a result of running this job are answered.
%%JOBD	Job description. Valid values are in the *<library>/<jobDescription>* format or the value: ***USRPRF**
%%JOBPTY	Job queue scheduling priority.
%%QUEUE	Name of the batch queue to which the job is submitted. Valid values are in the *<library>/<jobQueue>* format or the value: ***JOBD**
%%JOBQ	Same as %%QUEUE.
%%LDA	Local data areas (see Expressions for %%LDA [iSeries (AS/400) only] (on page 252)).
%%LIBL **or** **%%INLLIBL**	Library list. **NOTE:** The %%INLLIBL variable is relevant only for Control-M/Agent for iSeries (AS/400) version 2.25 or later.
%%LOG	Message logging values used to determine the amount and type of information sent to the job log by the job.

Name	Description
%%LOGCLPGM	Whether the commands that are run in a control language program are logged to the job log through the message queue of the CL program.
%%MSGQ	Name of the message queue to which a completion message is sent when the submitted job has completed execution. Valid values are in the *<library>/<messageQueue>* format or one of the following special values: ***USRPRF**, ***WRKSTN**, or ***NONE** **NOTE:** Control-M/Agent for iSeries (AS/400) version 2.25 or later uses a specific message queue as an event mechanism for job completion. If the message queue for a job is changed using the %%MSGQ variable, the event driven mechanism for this job is disabled.
%%OUTPTY	Output priority for spooled output files produced by the job.
%%OUTQ	Qualified name of the output queue used for spooled files that specify OUTQ(*JOB). Valid values are in the *<library>/<outputQueue>* format or one of the following special values: ***CURRENT**, ***USRPRF**, ***DEV**, or ***JOBD**
%%PRTDEV	Qualified name of the default printer device for this job.
%%PRTTXT	Up to 30 characters of text that is printed at the bottom of each page of printed output and on separator pages.
%%RTGDTA	Routing data used to start the first routing step in the job.
%%SWS	Job switches.
%%SYSLIBL	System portion of the initial library list that is used by the submitted job.
%%HEX	Indicates that the value that immediately follows is a hexadecimal value, such as %%PARMn=%%HEX2AF4. This function ensures that the value is transferred to iSeries (AS/400) in the appropriate format.

Name	Description
%%DEC	Indicates that the value that immediately follows is a packed decimal value, such as, %%PARMn=%%DEC1289. This function ensures that the value is transferred to iSeries (AS/400) in the appropriate format.

NOTE: For %%HEX and %% DEC you cannot use %%DEC<something>(%%DECEMBER)or %%HEX<something>(%%HEXAGON).

UNISYS job submission parameters

Name	Description
%%ACCOUNT	Account ID for the job run.
%%ADDRUN	Execution statement to be inserted before the execution of the user's job. Examples: **%%ADDRUN="@prt,i"** or **%%ADDRUN="@add myqual*myfile.myelt"** Control-M/Agent inserts the specified command before the user job's runstream prior to job submission.
%%DEADLINE	Deadline assignment for the job.
%%INJOBPR	Flag that indicates if a job contains programs that print or direct standard output to the PR print queue.
%%NODEID	Host ID of the UNISYS computer to which the job is submitted. If specified, the value for **%%NODEID** overrides the contents of the job processing parameter Host ID. If this variable is assigned the value CTMLOCAL, the job is submitted on the Control-M/Server computer. **NOTE:** Do not use this variable in a SMART Folder definition.
%%OPTION	Options to be appended to the **@START** command.
%%PROJECT	Project ID for the job run.
%%RUNTIME	Estimated run time for the job.

Tandem job submission parameters

Name	Description
% %TANDEM_ASSIGN_xxx	Enables the user to specify a TANDEM ASSIGN as an Variable variable to be used in a Control-M job environment. Where **xxx** is the name of the Tandem ASSIGN variable.
% %TANDEM_BYPASS_JOB	Enables the user to simulate running a dummy job.
% %TANDEM_CPU	Specifies a CPU on which a job should run.
% %TANDEM_DEFINE_xxx	Enables the user to specify a TANDEM DEFINE variable as a variable to be used in a Control-M job environment. Where **xxx** is the name of the Tandem DEFINE variable.
% %TANDEM_INPUT_FILE	Name of a file containing a script to run in a detached job.
% %TANDEM_PARAM_xxx	Enables the user to specify a TANDEM PARAM variable as a variable to be used in a Control-M job environment. Where **xxx** is the name of the Tandem PARAM variable.

Control-M system variables

The following table describes the available system variables. These reserved variables can be used to include system information in job processing parameter values.

Please note the following:

- Start of the week depends upon user preferences specifying whether 1 = Sunday or 1 = Monday. Descriptions in the following table that refer to day of the week presume that 1= Sun. Consult your Control-M administrator to determine which standard is used at your site.
- All System variable names must be specified using uppercase letters.
- Certain system variables can be referenced only after job completion (in postprocessing parameters). These variables are listed separately in Actions system variables table below.

Control-M system variables

Name	Format	Description
% %$DATE	yyyymmdd	Current system date (4-digit year).
% %$NEXT	yyyymmdd	Next scheduling date for the job (4-digit year).
% %$ODATE	yyyymmdd	Original scheduling date of the job (4-digit year).
% %$OYEAR	yyyy	Original scheduling year of the job (4-digit year).
% %$PREV	yyyymmdd	Previous scheduling date for the job (4-digit year).
% %$RDATE	yyyymmdd	Installation current working date (4-digit year).
% %$RYEAR	yyyy	Installation current working year (4-digit year).
% %$YEAR	yyyy	Current system year (4-digit year).
% %APPLGROUP	string	Name of the group to which the job belongs.
% %APPLIC	string	Name of the Application to which the job's group belongs. **NOTE:** For z/OS jobs the **% %APPL** variable is used to reference the Application name.
% %BLANKn	n spaces	Resolves to n blanks, where n is a number between 1 - 214.
% %CENT	yy	First two digits in the current year (for example, 20 in year 2006).

Name	Format	Description
% %CYCLIC	Y|N	This variable is used in the command line of the ctmorder utility to override the Cyclic parameter in cases where the user wishes to order a single run of a job that is defined as cyclic. **Y** – job is cyclic **N** – job should be run only once.
% %DATACENTER	string	Name of the Control-M/Server for the current Control-M installation. **NOTE:** This variable is available on certain Control-M computers as of Control-M version 2.20. However, prior to CONTROL-M/EM version 6.1.02, **% %DATACENTER** returned the host name for the current Control-M/Server.
% %DATE	yymmdd	Current system date.
% %DAY	dd	Current system day.
% %GROUP_ORDID	nnnnnn	Order ID of the Active Folder to which the job belongs. **NOTE:** ▪ This variable is valid only for jobs in an Active folder and it is evaluated in base 10. ▪ This variable has been retained for versions of Control-M earlier than version 7.0.00.
% %JOBNAME	string	Name of the submitted job. This variable can be used to override the value specified for the JOBNAME parameter (for example, in the ctmorder utility). **NOTE:** On a Microsoft Windows computer, JOBNAME must comply with Microsoft naming conventions (for example, it cannot contain **/** and **** characters).
% %JULDAY	nnn	Current system day of the year (Julian format).
% %MEMLIB	string	Mem Lib (name of the library or directory in which job script is stored) of the job.
% %MONTH	mm	Current system month.

Name	Format	Description
%%NEXT	yymmdd	Next scheduling date for the job.
%%ODATE	yymmdd	Original scheduling date of the job.
%%ODAY	dd	Original scheduling day of the job.
%%OJULDAY	nnn	Original scheduling day of the year (Julian format). For example, **36** for February 5th.
%%OMONTH	mm	Original scheduling month of the job.
%%ORDERID	nnnnnn	Unique job order ID under Control-M.
%%OWDAY	d	Original scheduling day of the week of the job (1= Sun., 2= Mon., and 0=Sat.).
%%OWNER	string	Owner (user ID) associated with the job.
%%OYEAR	yy	Original scheduling year of the job.
%%POSTCMD	command	Specifies a command to run immediately after running the job defined by MEMNAME. The return code is ignored. **NOTE:** Not relevant for z/OS jobs. Relevant only for Control-M/Server version 6.0.01 or later, or (without OUTPUT) for any job submitted by Control-M/Agent for Microsoft Windows 2000 version 6.0.01 or later. This variable is not used with SMART Folder.
%%PRECMD	command	Specifies a command to run immediately before running the job defined by MEMNAME. The return code is ignored. **NOTE:** Not relevant for z/OS jobs. Relevant only for Control-M/Server version 6.0.01 or later, or (without OUTPUT) for any job submitted by Control-M/Agent for Microsoft Windows 2000 version 6.0.01 or later. This variable is not used with SMART Folder.
%%PREV	yymmdd	Previous scheduling date for the job.
%%RDATE	yymmdd	Installation current working date.
%%RDAY	dd	Installation current working day.

Name	Format	Description
%%RJULDAY	nnn	Installation current working day of the year (Julian format). For example, **36** for February 5th.
%%RMONTH	mm	Installation current working month.
%%RUNCOUNT	nnnn	Number of times the job order has been submitted for execution (that is, the first time the job is being submitted, this variable returns a value of 1).
%%RWDAY	n	Installation current working day of the week (1=Sun., 2=Mon., and 0=Sat.).
%%RYEAR	yy	Installation current working year.
%%SCHEDTAB	string	Name of the job's folder. **NOTE:** This variable is available on certain Control-M computers as of Control-M version 2.2x.
%%GROUP_ORDID	nnnnnn	Order ID of the group to which the job belongs. **NOTE:** This variable is valid only for jobs in an active folder and it is evaluated in base 10.
%%TIME	hhmmss	Time of day.
%%WDAY	n	Current system day of the week (1=Sun., 2=Mon., and 0=Sat.).
%%YEAR	yy	Current system year.

The following variables are not assigned values until after the job is submitted or completes execution. These variables can be used only for the **Notify**, , and **Set Variable** parameters.

Variables that return runtime statistics for a job must be resolved before the variables can return any values. Variables for job statistics: %%AVG_CPU, %%AVG_TIME, %%SD_CPU, and %%SD_TIME. Run the ctmjsa utility to compile data in the Statistical Details table before specifying the required Variable statistics variable. For more information, see Statistics and reporting and ctmjsa in *Control-M Utilities*.

Actions system variables

Name	Format	Description
%%AVG_CPU	numeric	Average CPU time (in seconds) for previous runs of the current job. **NOTE:** Do not use this variable in a SMART Folder definition.
%%AVG_TIME	numeric	Average run time (in seconds) for previous runs of the current job or SMART Folder.
%%COMPSTAT	numeric	Completion code assigned to the job by the operating system of the computer that executes the job. Initial value: **0**.
%%JOBID	string	Identification assigned to the job by the operating system of the computer that executes the job. **NOTE:** Do not use this variable in a SMART Folder definition.
%%NODEID	string	Host ID of agent computer that submitted the job. **NOTE:** This variable is available for certain computers as of Control-M version 2.2x.
%%SD_CPU	numeric	Standard deviation of the CPU time (in seconds) from the average CPU time for previous runs of the current job. **NOTE:** Do not use this variable in a SMART Folder definition.
%%SD_TIME	numeric	Standard deviation of the elapsed run time (in seconds) from the average elapsed run time for previous runs of the current job or SMART Folder.

User-defined variables

A user-defined variable is created when it is assigned a value using the Variable Assignment parameter or the Do Variable parameter.

There are four types of variables to choose from when adding a variable to your job definition, either when defining a variable assignment or when you set a variable in an on do action:

- Local: Defines variables that can be used by other post-processing actions of the job, such as messages, do-action arguments, or the script itself in the next run.
- Global: Enables you to access the variables by its name which can be used by any job.
- Named Pool: Defines variables in a pool. The variable is referenced by the pool name - %%\\<named_pool>\<variable_name>.

  If **Named Pool** is selected, the **Pool Name** field appears. In the **Pool Name** field, type the name of the pool.

  You can use a local variable as a pool name which enables you to pass a pre-defined pool name to jobs at order time.

  EXAMPLE:

  Variables added to a job:

  Type: Local **Variable Name**: localvar **Value**: val

  Type: Named Pool **Variable Name**: namepool **Pool Name**: %%localvar **Value**: namval

  After you order the job, the named pool variable resolves the localvar variable as the pool name of the named pool variable.
- Smart Folder: Enables other jobs in the folder to access the specified variable either in the script during the job run, or in post-processing.

  NOTE: The **SMART folder** option in the **Variable Type** drop-down list appears when the job is included in the SMART folder.

User variables can also be defined for all jobs in a SMART Folder in the folder properties pane. For more information about assigning a value to a variable, see Variable Expressions (on page 248).

User-defined variables can be used to

- Store intermediate values in a series of Variable parameters
- Store values to be included in a command string in the Command parameter
- Store information to be included in a Notification message
- Store information to be included in an e-mail message created with a Do Mail parameter

NOTE: Global variables can be used to pass information between jobs in a Control-M/Server. For example, job A can set global variable %%A to **Yes,** and job B on another agent in the same Control-M/Server can reset %%A to **No** in response. Global variables can also be created and modified using the ctmvar utility. For more information about this utility, see *Control-M Workload Automation Utilities*.

Resolution of each user variable depends on the specified prefix, and the scope of the specified variable. Each of these concepts is described below.

Syntax

Valid names for user variables are any alphanumeric string (up to 38 characters in length) preceded by a prefix of **% %**. Blanks are not allowed in a user variable name.

The following characters cannot be included as part of the name of a User-defined variable: < > [ ] { } ( ) = ; ` ~ | : ? . + - * / & ^ # @ ! , " '.

NOTE: Application-specific job parameters may not be specified in variable values. The names of application-specific job parameters are prefixed by two percent signs, the application's abbreviation and a hyphen (%%SAPR3- for SAP, %%OAP- for Oracle, and so on).

Names and values for User variables are case sensitive. For example, %%TEST and %%Test are regarded as two separate variables.

NOTE: Names of variables in Control-M for z/OS must always be in uppercase.

All variables are prefixed by **%%**. In addition to the %% prefix, certain characters can be added to determine special characteristics. These special prefixes are described in the following table.

Variable Prefix	Description
%%	Indicates that a variable is global for the Control-M/Server (Control-M/Server and all its connected agents). This prefix is used only when creating or modifying the variable. When the variable is referenced (for example, in a Notification message) it is referenced without the ****. For more information about global variables, see the **Scope** section below.
%%#	Indicates that the variable should not be resolved. This prefix enables inclusion of a variable name as text in job output. The **#** symbol is stripped from the output, and the remaining variable name is included as text in the appropriate location. For example, a Notification message of **Job Daily returned a value for variable %%#PARM1** is output as: **Job Daily returned a value for variable %%PARM1**
%%@	Indicates that the variable should contain a value to be resolved by each job that uses it. For example: **%%\PARM1 = %%@TIME** Indicates that whenever a job uses Global variable %%PARM1, it should be resolved to the execution time of the job (that is, the time at which the variable is referenced). If the above parameter is specified without the @ sign (**%%\PARM1 = %%TIME**), it always resolves to the execution time of the job *that set the variable* (that is, the time at which the variable was created). **NOTE:** This prefix is relevant only for Global variables whose values contain variables. **NOTE:** If the value %%@ (value) is defined via a job in a POOL variable, the value is not resolved in the pool.

Variable Prefix	Description
**% % \ **	Indicates that the variable belongs to a SMART folder. You can update or add variables from the jobs that are included in the SMART folder. The jobs contained in the SMART folder inherit the variable defintion of the SMART folder. When the variable is referenced (for example, in a Notification message) it is referenced without the **\ \.** For more information about global variables, see the **Scope** section below.
% % \ \ <pool name> \ <variable name>	Indicates that the variable belongs to a container of variables that can be referenced by any active job, by the pool name. **EXAMPLE**: %%\\pool\COUNT
% %	For Windows agents: When specifying variables in the **CMDLINE** or **COMMAND** parameters on Control-M/Agent for Windows, the Variable prefix must be specified as **% %**.
	NOTE: If you use an agent utility (such as ctmcreate) in a batch file to specify an variable, ensure that the prefix is %%%%.

Variable Prefix	Description
%%A=%%ODATE	The **%%A** variable is resolved to the original scheduling date of the job. **%%A** is local to the job.
%%\A=%%ODATE	Global variable **%%\A** is assigned to the original scheduling date of the job. **%%A** can be referenced by Control-M/Server or any agent in the data centerControl-M/Server. All references to variable **%%A** resolve to the %%ODATE value for the job in which **%%A** was set.
%%\A=%%@ODATE	Global variable **%%\A** is resolved to the original scheduling date of the job. If **%%A** is referenced by Control-M/Server or a job in any agent in the Control-M/Server, it resolves to the current value of variable **%%ODATE**. @ indicates that **%%\A** should contain a value to be resolved by each job that uses it (in this case, **ODATE**).
%%#A	**%%#A** is not resolved. The text string **%%A** is returned.

Scope

The scope of a variable is the extent to which it is available to other jobs. As mentioned above, each variable can be

- local for a specific job
- common to all jobs in a SMART Folder
- global for an entire Control-M/Server (a Control-M/Server and all its agents)
- system Variables which are predefined and available for any job in the Control-M/Server

Multiple variables (each with a different scope) can have the same name. If more than one variable with the same name has been defined, the variable with the narrowest scope is used.

NOTE: The exception to this rule occurs when the variable is distributed from the Control-M/Server to the Control-M/Agent. The variable from the narrowest scope may not be the one that is used by the Control-M/Agent.

NOTE: Depending on the value of the VARIABLE_INC_SEC variable of the Control-M/Server, duplicate variables from different scopes can be distributed to the agents. For more information, see *Control-M Workload Automation Administration*.

Control-M uses the following logic to determine which value to use when a variable is specified in a job processing definition:

- Control-M checks if a local variable (for the job) has been defined with the specified name. If a local variable exists, the value specified for that variable is used.
- If no local variable exists with the specified name, and the job is in a SMART Folder, Control-M checks for a variable with the specified name in the SMART Folder definition. If the variable is defined in the SMART Folder definition, that value is used.
- If the job is not in a SMART Folder, or the variable is not defined in the SMART Folder definition, Control-M searches for a Global variable with the specified name.
- VARIABLE_INC_SEC must be set to Global to resolve a variable that is included in the script of a job, where the variable is defined in the SMART Folder.
- If no definition is found for a specified variable, the variable is resolved to the reserved word **CTMERR**.

Variable lists

The %%LIBMEMSYM variable can be used in the **Variable Assignment** or **Do Variable** parameters to indicate a text file containing a list of Variable assignments. . This variable enables you to create a central file containing Variable assignment statements that can be read by many job processing definitions.

You can also use the **%%POOLSYM** variable to call variables from a pool. **%%POOLSYM** is relevant only to a Control-M distributed system.

NOTE: Multiple LIBMEMSYM statements can be included in a single job processing definition. In this way you can maintain groups of local or global variables that are relevant to certain jobs, and allow more than one such group to be applied to a job. However, if a variable is defined in more than one list, the last list defined overrides previous lists.

The format for the **%%LIBMEMSYM** variable:

%%LIBMEMSYM=*path_name*

path_name is the full (case sensitive) path name of a text file containing Variable assignment statements. This file must be accessible to Control-M. The file must contain a single assignment on each line.

- For the Variable Assignment parameter, the specified file is accessed at the time the job is submitted for execution by Control-M.
- If a **%%LIBMEMSYM** statement is specified using the Do Variable parameter, local variables defined in the specified file are relevant only during a rerun of the job if and when the On Statement/Code conditions are satisfied.

NOTE: For z/OS jobs, a variable list can be specified using the %%LIBSYM and %%MEMSYM variables. If a %%LIBMEMSYM variable is specified for an z/OS job, it is treated as a user-defined variable, and does not reference a variable list.

EXAMPLE: If a job processing definition contains the following specifications for the **Variable Assignment** parameter:

- %%a=5
- %%LIBMEMSYM=/controlm/ctm/variable.common1
- %%PARM1=%%c

And the file **/controlm/ctm/variable.common1** contains the following entries:

- %%b=%%CALCDATE %%DATE -%%a
- %%yy=%%SUBSTR %%b 1 2
- %%mm=%%SUBSTR %%b 3 2
- %%dd=%%SUBSTR %%b 5 2
- %%c=%%dd/%%mm/%%yy

When the job is submitted for execution by Control-M, variables are resolved as if the Variable Assignment parameter contained the following entries:

- %%a=5
- %%b=%%CALCDATE %%DATE -%%a
- %%yy=%%SUBSTR %%b 1 2
- %%mm=%%SUBSTR %%b 3 2
- %%dd=%%SUBSTR %%b 5 2
- %%c=%%dd/%%mm/%%yy
- %%PARM1=%%c

The format for the **%%POOLSYM** variable:

%%POOLSYM=<*poolname*>

poolname is the name of the pool containing the pool variable.

EXAMPLE: If a job processing definition contains the following specifications for the **Variable Assignment** parameter:

- %%a=5
- %%POOLSYM=*<poolname>*
- %%PARM1=%%c

Variables in *<poolname>* include the following entries:

- %%b=%%CALCDATE %%DATE -%%a
- %%yy=%%SUBSTR %%b 1 2
- %%mm=%%SUBSTR %%b 3 2
- %%dd=%%SUBSTR %%b 5 2
- %%c=%%dd/%%mm/%%yy

When the job is submitted for execution by Control-M, variables are resolved as if the Variable Assignment parameter contained the following entries:

- %%a=5
- %%b=%%CALCDATE %%DATE -%%a
- %%yy=%%SUBSTR %%b 1 2
- %%mm=%%SUBSTR %%b 3 2
- %%dd=%%SUBSTR %%b 5 2
- %%c=%%dd/%%mm/%%yy
- %%PARM1=%%c

Variable Expressions

An Variable expression consists of any of the following:

- An elementary expression. Elementary expressions (on page 249)
- A numeric expression consisting of variables and/or numeric constants using Variable operators. Numeric expressions (on page 249)
- A string expression formed by concatenating variables and/or alphanumeric strings. String Expressions (on page 250)
- An Variable function. Variable functions (on page 251)
- A special expression that can be used to specify Local Data Areas for iSeries (AS/400) jobs. Expressions for %%LDA [iSeries (AS/400) only] (on page 252)

NOTE: Variables can also be used to represent part of the name of an existing variable in an expression. When this method is used the name of the variable is resolved and then the contents of that variable are used for further resolution of the expression. (For more information, see the examples later in this chapter.)

A Notification message can include an Variable expression as part of its text; however, it cannot assign a value to a variable.

The following syntax rules apply to Variable expressions:

- Only one expression can be placed on a line.
- No spaces are allowed before the "=".
- Spaces that immediately follow the "=" are ignored.
- Variable names and values are case sensitive.
- Strings of alphanumeric characters do not require quotation marks. If quotation marks are used, they are considered part of the string.

Elementary expressions

An elementary expression has the following format:

variable=*value*

variable is any user variable or job submission variable.

value is any numeric or alphanumeric string up to 214 characters in length.

- The following expression assigns a value of 100 to the %%Result variable:

  %%Result=100

- The following expression assigns the string **Job "PRDKCZ" finished OK** to the %%Completion variable:

  %%Completion=Job "PRDKCZ" finished OK

Numeric expressions

A numeric expression has the following format:

result=*operand operator operand*

- **Result:** Any user variable or job submission variable.
- **Operand:** Any variable, *or* a numeric constant.
- **Operator:** One of the following Variable operators:
  - **%%PLUS** represents the "+" operator
  - **%%MINUS** represents the "-" operator

Syntax rules

The following syntax rules apply to a numeric expression:

- Only one operator can be used in each expression.
- There should be no spaces immediately before or after the "=".
- There must be spaces before and after the operator.

NOTE: When specifying Variable functions that return dates, such as %%$CALCDATE or %%$DATE, there should be no spaces between the **+** (plus) or **-** (minus) sign and the number of days to add or subtract. In this case, the + and **-** signs are not considered numeric expression operators and are not related to the rules specified here.

- In the following example, %%YESTERDAY resolves to 0 on July 1:

%%YESTERDAY=%%DAY %%MINUS 1

- In the following example, %%PARM1 resolves to 46 on February 5th:

%%X=%%RJULDAY %%MINUS %%DAY
%%PARM1=%%X %%PLUS 15

String Expressions

A string expression has the following basic format:

<variable>=<value>[.]<value>...

- **<variable>:** Any user variable or job submission variable.
- **<value>:**Any variable, or alphanumeric string.
- **. (period):** If specified, indicates that the values before and after the period should be concatenated.

Syntax rules

The following syntax rules apply to a string expression:

- Any spaces in the expression (including those immediately following the "=") are regarded as part of the string and are included in the result.
- A period is used to concatenate two variables. No concatenation character is required to link two strings.
- To include a period as part of the string between two concatenated variables, use two consecutive periods (see example below).
- Any number of variables or strings can be concatenated in an expression.
- If one variable follows another with no period in between, the two variables are combined to form the name of a third variable (see example below). Concatenation progresses from right to left until the entire expression is resolved.

EXAMPLE: %%X resolves to 0312:

- %%X=%%DAY.%%MONTH
- %%Y resolves to 03.12:

- %%Y=%%DAY..%%MONTH
- %%Z resolves to "Today is 03/12/05":
- %%Z= Today is %%DAY/%%MONTH/%%YEAR

The following sequence passes a value to **%%PARM1** based on the day of the month. This example illustrates building the name of a variable in the expression:

- %%BackupTape_01=301
- %%BackupTape_02=302
- %%BackupTape_03=303
- %%PARM1=%%BackupTape_%%DAY

The **%%BackupTape_%%DAY** expression is resolved in two steps (from right to left):

Given that the system date is Dec. 3, %%DAY resolves to 03. The resulting expression is:

%%PARM1=%%BackupTape_03

Next, Control-M resolves the %%BackupTape_03 variable. Since the value of this variable is 303, the resulting expression is:

%%PARM1=303

Variable functions

An Variable function performs an action or process on the specified user variable or job submission variable. Variable functions are used instead of another expression. The following functions are available:

Variable Functions

Function	Description
"%%CALCDATE and %%$CALCDATE"	Adds or subtracts a specified number of days from a specified date. Same as %%CALCDATE, but handles 4-digit years.
%%GETENV <environment variable>	Retrieves the value of an environment variable.
"%%SUBSTR"	Extracts a substring from a specified string.

%%CALCDATE and %%$CALCDATE

%%CALCDATE is a numeric function that adds or subtracts a quantity of days from a given date. This function has the following format:

result=%%CALCDATE *date* +|-*quantity*

- *result*: Any user variable or job submission variable.
- *date*: Date or variable in **yymmdd** (**yyyymmdd** for **%%$CALCDATE**) format. This value must be preceded and followed by a space.
- *quantity*: Number (or variable that resolves to a number) of days to add or subtract from the date.

NOTE: On July 2, 2005, %%A resolves to 050630 in any of the following expressions:

- %%A=%%CALCDATE %%DATE -2
- %%A=%%CALCDATE 050702 -%%DAY
- %%A=%%CALCDATE 050628 +2

NOTE: On July 2, 2005, %%A resolves to 20050630 in any of the following expressions

- %%A=%%$CALCDATE %%$DATE -2
- %%A=%%$CALCDATE 20050702 -%%DAY
- %%A=%%$CALCDATE 20050628 +2

%%SUBSTR

%%SUBSTR is a string function that is used to extract a substring from within a larger string. This function has the following format:

result=%%SUBSTR *variable startpos length*

- *result:* Any user variable or job submission variable.
- *variable:* Any variable.
- *startpos:* Numeric literal or variable that indicates the first position in the original string from which to extract the substring. The first character is position 1
- *length:* A number or variable, indicating the length of the substring to extract.

NOTE: The values specified for both <startpos> and <length> must be (or resolve to) a number greater than zero

NOTE: In the following series of expressions, **%%Number** resolves to **TWO**:

- %%Nstring=ONETWOTHREE
  %%Start=4
  %%Number=%%SUBSTR %%Nstring %%Start 3

%%GETENV <environment variable>

This function retrieves the value of an environment variable. It has the following format: result=%%GETENV <environment variable>.

NOTE: %%A=%%GETENV HOME

%%A resolves to the Control-M/Server user home directory.

Expressions for %%LDA [iSeries (AS/400) only]

%LDA is a special variable that can be used to specify Local Data Areas for iSeries (AS/400) jobs.

The following format can be used when specifying an expression to assign a value to job submission variable %%LDA for a job:

%%LDA[_*startpos*[_*length*] [_*dec*]]=Variable_exp

- *startpos:* A number indicating the starting position in the LDA at which to place the results of the specified expression.
- *length:* A number, indicating how many positions in LDA to reserve for the results of the expression.
- *dec:* A number indicating how many decimal places to include in the packed decimal value passed to the LDA.
- **Variable_exp:** The Variable expression whose resolved value should be placed in the indicated location in the LDA.

EXAMPLE: Assuming that the system date is December 15, the following expression:

%%LDA_2_3=%%DAY

is submitted to iSeries (AS/400) as:

CHGDTAARA *LDA(2 3) VALUE(15)

The following expression inserts the packed value of 00123.40 into the LDA starting in position 11 for a decimal length of 7 (Actual Packed length of 4):

%%LDA_11_7_2=123.4

BIM Variables

The following table lists the BIM Variables.

Variable	Description
%%PROBLEMATIC_JOBS	Name of any job in a service that is not running on time, and, as a result, will impact the service. If more than one job is problematic, the names of all problematic jobs is returned.
%%SERVICE_DUE_TIME	Time by which the entire service should complete.
%%SERVICE_EXPECTED_END_TIME	Time BMC Batch Impact Manager estimates the service will complete.
%%SERVICE_NAME	Name of the service.
%%SERVICE_PRIORITY	Priority level of the service.