

CONTROL-M/Enterprise Manager API Developer Guide



Supporting

CONTROL-M/Enterprise Manager version 6.4.01

September 2008



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 2101 CITYWEST BLVD HOUSTON TX 77042-2827 USA	Telephone	713 918 8800 or 800 841 2031	Fax	713 918 8000
----------------	--	------------------	---------------------------------	------------	--------------

Outside United States and Canada

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

© Copyright 2008 BMC Software, Inc.

BMC, BMC Software, and the BMC Software logo are the exclusive properties of BMC Software, Inc., are registered with the U.S. Patent and Trademark Office, and may be registered or pending registration in other countries. All other BMC trademarks, service marks, and logos may be registered or pending registration in the U.S. or in other countries. All other trademarks or registered trademarks are the property of their respective owners.

AIX, OS/390, and z/OS are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

Oracle is a registered trademark of Oracle Corporation.

SAP is a registered trademark of SAP AG in Germany and in several other countries.

Java, Java runtime environment, JRE, Solaris and Sun Microsystems are trademarks or registered trademarks of Sun Microsystems, Inc., in the U.S. and other countries.

UNIX is the registered trademark of The Open Group in the US and other countries.

BMC Software considers information included in this documentation to be proprietary and confidential. Your use of this information is subject to the terms and conditions of the applicable End User License Agreement for the product and the proprietary and restricted rights notices included in this documentation.

Restricted rights legend

U.S. Government Restricted Rights to Computer Software. UNPUBLISHED -- RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES. Use, duplication, or disclosure of any data and computer software by the U.S. Government is subject to restrictions, as applicable, set forth in FAR Section 52.227-14, DFARS 252.227-7013, DFARS 252.227-7014, DFARS 252.227-7015, and DFARS 252.227-7025, as amended from time to time. Contractor/Manufacturer is BMC SOFTWARE INC, 2101 CITYWEST BLVD, HOUSTON TX 77042-2827, USA. Any contract notices should be sent to this address.

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_home. 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.

Contents

About this book	21
Related publications	21
Conventions	22
Syntax statements	23
Chapter 1 Overview	25
How the CONTROL-M/EM API works	26
Initialization	26
Connecting to CONTROL-M/EM	26
CONTROL-M/EM API sessions	27
Session example	27
Chapter 2 Installation	31
Compatibility	31
CONTROL-M product support	31
Prerequisites	32
Installing the CONTROL-M/EM API	33
Configuration	34
Chapter 3 Upgrade from earlier versions	39
Chapter 4 Configure the CONTROL-M/EM API	43
Preparing your project environment	43
Writing your project	44
Running your project	47
Chapter 5 Issuing requests and handling responses	49
Request types	49
Response types	50
Successful responses	50
Fault responses	51
Response types received when using EMBasicXMLInvoker	52
Response types received when using EMXMLInvoker	53
CONTROL-M/EM API programming methods	54
Deciding which method to use	54

Chapter 6	Class reference	67
ComponentType class		68
EMBasicXMLInvoker class		69
invoke.		70
setPollRequestIntervalMilli		71
setPollRequestTimeoutMilli		71
EMXMLInvoker class.		72
BuildPasswordString.		73
done		73
getProperties.		74
init.		75
invoke.		77
setProperties		77
GASComponent class.		78
GASComponent (Prototype 1)		79
GASComponent (Prototype 2)		79
GSRComponent class.		80
GSRComponent (Prototype 1)		80
GSRComponent (Prototype 2)		81
InvokeException class		81
getMajorCode		82
getMinorCode.		82
getReason		83
Chapter 7	Request reference	85
Introduction to CONTROL-M/EM API requests		87
User Registration		90
Request parameters		90
Response parameters.		91
Fault response.		91
Errors		92
Examples		92
Check user token validity		94
Request parameters		94
Response parameters.		94
Fault response parameters		95
Errors		95
Examples		95
Client Keep Alive		96
Request parameters		97
Response parameters.		97
Fault response parameters		97
Errors		97
Examples		98
User Unregistration		99
Request parameters		99
Response parameters.		100
Fault response parameters		100

Errors	100
Examples	100
Create job definitions	102
Request parameters	102
Response parameters	103
Fault response parameters	103
Errors	103
Examples	104
Create scheduling group definitions	106
Request parameters	107
Response parameters	107
Fault response parameters	108
Errors	108
Examples	109
Delete job definitions	112
Request parameters	113
Response parameters	115
Fault response parameters	116
Errors	116
Examples	117
Upload scheduling table	119
Request parameters	119
Response parameters	120
Polling request parameters	120
Polling response parameters	121
Fault response parameters	121
Errors	121
Examples	122
Order or Force	124
Request parameters	125
Response parameters	127
Polling request parameters	128
Polling response parameters	128
Fault response parameters	130
Errors	130
Examples	130
Job creation	133
Request parameters	134
Response parameters	153
Polling request parameters	154
Polling response parameters	154
Fault response parameters	155
Errors	156
Examples	156
Add condition	159
Request parameters	159
Response parameters	160
Polling request parameters	160
Polling response parameters	160

Fault response parameters	161
Errors	161
Examples	162
Delete condition	164
Request parameters	165
Response parameters	165
Polling request parameters	166
Polling response parameters	166
Fault response parameters	167
Errors	167
Examples	167
Job actions in active environment	170
Hold	171
Free	172
Confirm	174
Rerun	176
Kill	178
Force OK	179
Errors	181
Examples	182
Job tracking	184
Request parameters	184
Response parameters	185
Fault response parameters	186
Errors	187
Examples	188
Retrieve jobs in active environment	194
Request parameters	194
Response parameters	198
Fault response parameters	201
Errors	201
Examples	202
Change alert status	205
Request parameters	205
Response parameters	206
Fault response parameters	206
Errors	206
Examples	207
Retrieve BIM Services list	209
Request parameters	209
Response parameters	210
Fault response parameters	211
Examples	211
Fault Response	213
Fault Example	214

Chapter 8	Advanced features and optimization	215
------------------	---	------------

Modifying initialization properties	215
Prototype 1	216

Prototype 2	216
Prototype 3	217
Getting and setting CONTROL-M/EM API properties	219
getProperties	219
setProperties	220
Polling interval timeout configuration	221
setPollRequestIntervalMilli	222
setPollRequestTimeoutMilli	222

Chapter 9 Diagnostics and troubleshooting 223

CONTROL-M/EM API logging	223
Default logging behavior	224
Modifying logging behavior	224
Environment configuration troubleshooting	226
CLASSPATH: missing libraries or directories	226
Java virtual machine parameters	227
Incompatible object argument for a function call	229
Application runtime and communication troubleshooting	229
An exception is thrown by the invoke method	230
An error occurs when an XML file is submitted.	231
Application cannot be started	231

Chapter 10 Uninstalling CONTROL-M/EM API 235

Appendix A Request format examples 237

Add Condition or Delete Condition request	238
Example 1: Add a condition to a CONTROL-M	238
Example 2: Delete a condition from a CONTROL-M	238
Job Creation request	239
Example 1: Create a job requiring confirmation	239
Example 2: Create a cyclic Job	240
Example 3: Create a job including In and Out conditions	241
Example 4: Create a job that requires resources	242
Example 5: Create a job that includes On-Do statements	243
Example 6: Create a job that includes On-Do statements	244
Example 7: Create an active group scheduling table	245
Example 8: Create an active job in an existing group scheduling table	246
Order or Force request	247
Example 1: Order a UNIX job	247
Example 2: Force a UNIX job	247
Example 3: Force a UNIX job into a 'recent' group scheduling table	248
Example 4: Force a UNIX job into a 'recent' group scheduling table allowing duplication	249
Example 5: Force a scheduling table that contains a group scheduling table ..	250

Appendix B Error codes and exceptions 251

Severity	252
----------------	-----

Error code reference	253
NULL exception errors (Major code 000)	254
Low-level API exceptions (Major code 100)	254
Parser exceptions (Major code 200)	255
CONTROL-M/Server errors: Group 1 (Major code 300)	255
CONTROL-M/Server errors: Group 2 (Major code 301)	257
CONTROL-M/Server errors: Group 3 (Major code 302)	258
Generic request exceptions (Major code 401)	258
Poll request errors (Major code 403)	259
Add or Delete Condition request errors (Major code 404)	259
Order or Force request errors (Major code 405)	259
Job tracking request errors (Major code 406)	260
Authorization request errors (Major code 407)	260
Alerts request errors (Major code 408)	261
Create active job request errors (Major code 409)	261
Upload scheduling table request errors (Major code 411)	261
Create job/scheduling group definitions request errors (Major code 412)	262
Delete job definitions request errors (Major code 413)	262
Retrieve active jobs request errors (Major code 440)	263
Job actions request errors (Major code 450)	263
CONTROL-M/EM API Java client errors (Major code 500)	264
Gateway messages (Major code 600)	265

Appendix C Job and Scheduling Group XML parameters	273
---	------------

Glossary	297
-----------------	------------

Index	301
--------------	------------

Figures

CONTROL-M/EM API process flow	27
Sample class	45
Differences in request methods	55
EMBasicXMLInvoker request session	56
EMXMLInvoker request session	60
EMXMLInvoker polling request steps	61
request_register XML parameters	90
response_register XML parameters	91
request_check_user_token XML parameters	94
response_check_user_token XML parameters	94
request_client_keep_alive XML parameters	97
request_unregister XML parameters	99
response_unregister XML parameter	100
request_def_create_jobs XML parameters	102
response_def_create_jobs XML parameter	103
request_def_create_sched_group XML parameters	107
response_def_create_sched_group XML parameters	107
request_def_delete_jobs XML parameters	113
delete_jobs_criterion XML parameters	113
include or exclude XML parameter	114
param XML parameters	114
response_def_delete_jobs XML parameters	115
request_def_upload_table XML parameters	119
request_poll XML parameters	120
response_poll_def_upload_table XML parameters	121
request_order_force XML parameters	125
scheduling_group_info XML parameters	126
autoedit_assignment XML parameters	127
response_order_force XML parameters	127
request_poll_order_force XML parameters	128
response_poll_order_force XML parameters	128
job XML parameters	129
job_data XML parameters	129
request_create_aj XML parameters	134
on_do_statement XML parameters	141
on_statements XML parameters	141
on_statement XML parameters	142
on_statement_sysout XML parameters	142
do_statements_type XML parameters	143
do_autoedit XML parameters	144

do_cond XML parameters	144
do_ctrule XML parameters	145
do_forcejob XML parameters	145
do_ifrerun XML parameters	145
do_mail XML parameters	146
do_shout XML parameters	147
do_sysout XML parameters	147
autoedit_assignment XML parameters	148
ctb_step XML parameters	148
in_condition XML parameters	149
control_resources XML parameters	149
quantitative_resource XML parameters	150
pipe XML parameters	150
out_condition XML parameters	151
step_range XML parameters	151
shouts XML parameters	152
interval_sequence XML parameters	153
specific_times XML parameters	153
response_create_aj XML parameters	153
request_poll_create_aj XML parameters	154
response_poll_create_aj XML parameters	154
job XML parameters	154
job XML parameters	155
job_data XML parameters	155
request_add_condition XML Parameters	159
condition XML parameters	159
response_add_condition XML parameters	160
request_poll_add_condition XML parameters	160
response_poll_add_condition XML parameters	160
response_data XML parameters	161
request_delete_condition XML parameters	165
condition XML parameters	165
response_delete_condition XML parameters	165
request_poll_delete_condition XML parameters	166
response_poll_delete_condition XML parameters	166
response_data_XML parameters	166
request_aj_hold XML parameters	171
response_aj_hold XML parameters	171
request_poll_aj_hold XML parameters	171
response_poll_aj_hold XML parameters	172
request_aj_free XML parameters	172
response_aj_free XML parameters	173
request_poll_aj_free XML parameters	173
response_poll_aj_free XML parameters	174
request_aj_confirm XML parameters	174
response_aj_confirm XML parameters	175
request_poll_aj_confirm XML parameters	175
response_poll_aj_confirm XML parameters	175
request_aj_rerun XML parameters	176

response_aj_rerun XML parameters	176
request_poll_aj_rerun XML parameters	177
response_poll_aj_rerun XML parameters	177
request_aj_kill XML parameters	178
response_aj_kill XML parameters	178
request_poll_aj_kill XML parameters	178
response_poll_aj_kill XML parameters	179
request_aj_force_ok XML parameters	179
response_aj_force_ok XML parameters	180
request_poll_aj_force_ok XML parameters	180
response_poll_aj_force_ok XML parameters	181
request_job_track XML parameters	184
job XML parameters	184
response_job_track XML parameters	185
job XML parameters	185
job_data XML parameters	186
fault_job_track	186
job XML parameters	187
request_act_retrieve_jobs XML Parameters	194
retrieve_jobs_criterion	195
include and exclude XML parameters	195
search_criterion XML parameter	195
param XML parameters	196
response_act_retrieve_jobs XML parameters	198
request_change_alert_status XML parameters	205
response_change_alert_status XML parameters	206
request_get_bim_services_info XML parameters	209
response_get_bim_services_info XML parameters	210
bim_services XML parameters	210
emapi_log.cfg file example with default parameters	225
on_do_statement XML parameters	281
on_statements XML parameters	281
on_statement XML parameters	282
on_statement_sysout XML parameters	282
do statements type XML parameters	283
do_autoedit XML parameters	284
do_cond XML parameters	284
do_ctbrule XML parameters	285
do_forcejob XML parameters	285
do_ifrerun XML parameters	285
do_mail XML parameters	286
do_shout XML parameters	287
do_sysout XML parameters	287
do_remedy XML parameters	288
autoedit_assignment XML parameters	289
in_condition XML parameters	289
control_resources XML parameters	289
quantitative_resource XML parameter	290
out_condition XML parameters	291

step_range XML parameters	291
shouts XML parameters	292
tag XML parameters	293
job_tag XML parameters	295
interval_sequence XML parameters	296
specific_times XML parameters	296

Tables

CONTROL-M/EM API (EMXMLInvoker) use flow	28
CONTROL-M/EM API primary subdirectories	34
Configuration files	35
Request types	50
EMBasicXMLInvoker response types	52
EMXMLInvoker response types	53
Request types listed by component type	63
CONTROL-M/EM API classes	68
EMBasicXMLInvoker methods	69
EMXMLInvoker methods	72
GASComponent constructors	78
GASComponent constructors	80
InvokeException class methods	82
Requests listed in this chapter	87
request_register XML parameters description	90
response_register XML parameters description	91
request_check_user_token XML parameters Description	94
response_check_user_token XML parameters description	94
request_client_keep_alive XML parameter description	97
response_client_keep_alive XML parameters description	97
request_unregister XML parameter description	99
response_unregister XML parameter description	100
request_def_create_jobs XML Parameters Description	102
sched_table XML parameters description	102
response_def_create_jobs XML parameters description	103
sched_table XML Parameters Description	103
request_def_create_sched_group XML Parameters Description	107
sched_table XML parameters description	107
response_def_create_sched_group XML parameters description	108
sched_table XML Parameters Description	108
request_def_delete_jobs XML parameters Description	113
sched_table XML Parameters Description	113
delete_jobs_criterion XML parameters description	113
include or exclude XML parameter description	114
search_criterion XML parameters description	114
param XML parameters description	114
Valid values for name	115
response_def_delete_jobs XML parameters description	115
sched_table XML Parameters Description	116
request_def_upload_table XML parameters Description	120

sched_table XML Parameters Description	120
response_def_upload_table XML Parameters Description	120
request_poll_def_upload_table XML parameters Description	120
response_poll_def_upload_table XML parameters description	121
request_order_force XML parameters description	125
scheduling_group_info XML parameters description	127
autoedit_assignment XML parameters description	127
response_order_force XML parameters description	128
request_poll_order_force XML parameters description	128
response_poll_order_force XML parameters description	128
job XML parameters description	129
job_data XML parameters description	129
request_create_aj XML parameters description	134
active_job XML Parameters Description	135
on_do_statement XML parameters description	141
on_statements XML parameters description	141
on_statement XML parameters description	142
on_statement_sysout XML parameters description	142
do_statements_type XML parameters description	143
do_autoedit XML parameters description	144
do_cond XML parameters description	144
do_ctbrule XML parameters description	145
do_forcejob XML parameters description	145
do_ifrerun XML parameters description	146
do_mail XML parameters description	146
do_shout XML parameters description	147
do_sysout XML parameters description	147
autoedit_assignment XML parameters description	148
ctb_step XML parameters description	149
in_condition XML parameters description	149
control_resources XML parameters description	149
quantitative_resource XML parameters description	150
pipe XML parameters description	151
out_condition XML parameters description	151
step_range XML parameters description	151
shouts XML parameters description	152
interval_sequence XML parameters description	153
specific_times XML parameters description	153
response_create_aj XML parameters description	153
request_poll_create_aj XML parameters description	154
response_poll_create_aj XML parameters description	154
jobs XML Parameters Description	154
job XML parameters description	155
job_data XML parameters description	155
request_add_condition XML Parameters Description	159
condition XML parameters description	159
response_add_condition XML parameters description	160
request_poll_add_condition XML parameters description	160
response_poll_add_condition XML parameters description	161

response_data XML parameters description	161
request_delete_condition XML parameters description	165
condition XML parameters description	165
response_delete_condition XML parameters description	166
request_poll_delete_condition XML parameters description	166
response_poll_delete_condition XML parameters description	166
response_data XML parameters description	167
request_aj_hold XML parameters description	171
response_aj_hold XML parameters description	171
request_poll_aj_hold XML parameters description	172
response_poll_aj_hold XML parameters description	172
request_aj_free XML parameters description	173
response_aj_free XML parameters description	173
request_poll_aj_free XML parameters description	173
response_poll_aj_free XML parameters description	174
request_aj_confirm XML parameters description	174
response_aj_confirm XML parameters description	175
request_poll_aj_confirm XML parameters description	175
response_poll_aj_confirm XML parameters description	176
request_aj_rerun XML parameters description	176
response_aj_rerun XML parameters description	177
request_poll_aj_rerun XML parameters description	177
response_poll_aj_rerun XML parameters description	177
request_aj_kill XML parameters description	178
response_aj_kill XML parameters description	178
request_poll_aj_kill XML parameters description	179
response_poll_aj_kill XML parameters description	179
request_aj_force_ok XML parameters description	180
response_aj_force_ok XML parameters description	180
request_poll_aj_force_ok XML parameters description	180
response_poll_aj_force_ok XML parameters description	181
request_job_track XML parameters Description	184
job XML parameters description	184
response_job_track XML parameters description	185
job XML parameters description	185
job_data XML parameters description	186
fault_job_track XML parameters description	186
job XML parameters description	187
request_act_retrieve_jobs XML Parameters Description	194
retrieve_jobs_criterion XML parameters description	195
include and exclude XML parameters description	195
search_criterion XML parameter description	196
param XML parameters description	196
Valid name parameter values	196
response_act_retrieve_jobs XML parameters description	198
job_data XML Parameters Description	199
request_change_alert_status XML parameters description	206
response_change_alert_status XML parameters description	206
request_get_bim_services_info XML parameters description	209

response_get_bim_services_info XML parameters description	210
bim_services XML parameters description	210
fault XML parameter description	213
error XML Parameters Description	213
CONTROL-M/EM API properties parameters	219
Priority levels for log messages	225
RollingFileAppender example properties	226
Log message priority levels	253
Error and exception major codes	253
NULL exceptions	254
Low level API exceptions	255
Parser exceptions	255
CONTROL-M/Server errors: Group 1	255
CONTROL-M Server errors: Group 2	257
CONTROL-M Server errors: Group 3	258
Generic request exceptions	258
Poll request errors	259
Add or Delete Condition request	259
Order/Force request errors	260
Job tracking request errors	260
Authorization request errors	260
Alerts request errors	261
Create active job request errors	261
Upload scheduling table request errors	261
Create job/scheduling group definitions request errors	262
Delete job definitions request errors	263
Retrieve active jobs request errors	263
Job actions request errors	263
CONTROL-M/EM API Java client errors	264
Gateway messages	265
job and sched_group XML parameters description	273
date XML parameter description	281
on_do_statement XML Parameters Description	281
on_statements XML parameters description	282
on_statement XML parameters description	282
on_statement_sysout XML parameters description	283
do statements type XML Parameters Description	283
do_autoedit XML parameters description	284
do_cond XML parameters description	284
do_ctbrule XML parameters description	285
do_forcejob XML parameters description	285
do_ifrerun XML parameters description	286
do_mail XML parameters description	286
do_shout XML parameters description	287
do_sysout XML parameters description	287
do_remedy XML parameters description	288
autoedit_assignment XML parameters description	289
in_condition XML parameters description	289
control_resources XML parameters description	290

quantitative_resource XML parameters description	290
out_condition XML parameters description	291
step_range XML parameters description	291
shouts XML parameters description	292
tag XML parameters description	293
date parameters description	295
job_tag XML parameters description	296
interval_sequence XML parameters description	296
specific_times XML parameters description	296

About this book

This book contains detailed information about the CONTROL-M/Enterprise Manager API. This book is intended for developers who create applications that send job scheduling requests to the CONTROL-M/Enterprise Manager (EM) product for processing. Familiarity with CONTROL-M/EM API features and job processing parameters is assumed.

Like most BMC documentation, this book is available in printed and online formats. To request printed books or to view online books and notices (such as release notes and technical bulletins), see the Customer Support website at http://www.bmc.com/support_home. Most product shipments also include the books on a documentation CD.

NOTE



Online books are formatted as PDF or HTML files. To view, print, or copy PDF books, use the free Adobe Reader from Adobe Systems. If your product installation does not install the reader, you can obtain the reader at <http://www.adobe.com>.

Related publications

The following related publications supplement this book:

Category	Document	Description
installation documents	<i>CONTROL-M Installation Guide</i>	describes the installation processes for implementing CONTROL-M/EM API databases and the CONTROL-M/EM API product on Microsoft Windows and UNIX [®] platforms.

Category	Document	Description
core documents	<i>CONTROL-M for z/OS User Manual</i>	describes all CONTROL-M concepts, features, facilities and operating instructions in detail. It can be used as a learning guide as well as a reference guide.
	<i>CONTROL-M Administrator Guide</i>	describes administrator responsibilities, customization, maintenance, and security of CONTROL-M/EM, CONTROL-M/Server and CONTROL-M/Agent.
	<i>CONTROL-M User Guide</i>	describes all CONTROL-M/EM concepts, features, facilities, and operating instructions. It can be used as a learning guide as well as a reference guide.
	<i>CONTROL-M Parameter Guide</i>	describes the parameters used for creating job processing definitions.
	<i>CONTROL-M Utility Guide</i>	describes the utilities used for creating and managing objects in the job production environment and maintaining various aspects of CONTROL-M®/Enterprise Manager.
notices	release notes, flashes, technical bulletins	describes up-to-date information such as: <ul style="list-style-type: none"> ■ updates to the installation instructions ■ last-minute product information

Conventions

This book uses the following special conventions:

- All syntax, operating system terms, and literal examples are presented in this typeface.
- Variable text in path names, system messages, or syntax is displayed in *italic* text:
testsys/instance/fileName
- The symbol => connects items in a menu sequence. For example, **Actions => Create Test** instructs you to choose the **Create Test** command from the **Actions** menu.

Syntax statements

The following example shows a sample syntax statement:

```
COMMAND KEYWORD1 [KEYWORD2 | KEYWORD3] KEYWORD4={YES | NO} fileName...
```

The following table explains conventions for syntax statements and provides examples:

Item	Example
Items in italic type represent variables that you must replace with a name or value. If a variable is represented by two or more words, initial capitals distinguish the second and subsequent words.	<i>alias</i> <i>databaseDirectory</i> <i>serverHostName</i>
Brackets indicate a group of optional items. Do not type the brackets when you enter the option. A comma means that you can choose one or more of the listed options. You must use a comma to separate the options if you choose more than one option.	[<i>tableName</i> , <i>columnName</i> , <i>filed</i>] [-full , -incremental , -level] (UNIX)
Braces indicate that at least one of the enclosed items is required. Do not type the braces when you enter the item.	{ <i>DBDName</i> <i>tableName</i> } UNLOAD device={ <i>disk</i> <i>tape</i> , <i>fileName</i> <i>deviceName</i> } { -a -c } (UNIX)
A vertical bar means that you can choose only one of the listed items. In the example, you would choose either <i>commit</i> or <i>cancel</i> .	{ <i>commit</i> <i>cancel</i> } { -commit -cancel } (UNIX)
An ellipsis indicates that you can repeat the previous item or items as many times as necessary.	<i>columnName</i> . . .

Overview

This chapter presents the following topics:

How the CONTROL-M/EM API works.	26
Initialization	26
Connecting to CONTROL-M/EM	26
CONTROL-M/EM API sessions	27
Session example.	27

The CONTROL-M/Enterprise Manager API is an open interface for external applications that enables you to exploit the capabilities of CONTROL-M/EM and CONTROL-M.

The CONTROL-M/EM API enables users of your application to perform the following functions in the CONTROL-M Business Integrated Scheduling environment:

- log a user in or out of CONTROL-M/EM
- create jobs and group scheduling tables in the CONTROL-M Active Jobs file
- order and force jobs
- order and force Scheduling Groups
- track job execution
- add or delete Conditions
- manipulate Alerts
- check the validity of user tokens
- create Job and Scheduling Group Definitions
- delete Job Definitions
- upload Scheduling Tables
- retrieve jobs from CONTROL-M Active Jobs file
- perform job actions such as Hold, Free, Rerun, Confirm, Kill and Force OK

How the CONTROL-M/EM API works

The CONTROL-M/EM API is a set of Java classes that enable Java developers to send requests to CONTROL-M/EM server components from within their own applications.

Initialization

Before sending requests, the CONTROL-M/EM API must be initialized. Initializing the API initializes the CORBA layer that forwards requests from the client side of the API to the CONTROL-M/EM GUI Server and the Global Alerts Server.

Connecting to CONTROL-M/EM

In this version of the API, a developer can interact with two CONTROL-M/EM server components:

- GUI Server
- Global Alerts Server

The developer initializes an instance of either **EMXMLInvoker** or **EMBasicXMLInvoker**, and chooses with which component to communicate. From then on, an unlimited number of requests can be sent using the invoke method of the EMXMLInvoker or EMBasicXMLInvoker that was initiated. For more information about the difference between the EMXMLInvoker and EMBasicXMLInvoker methods, refer to [“CONTROL-M/EM API programming methods” on page 54](#).

Authentication and user tokens

Each CONTROL-M/EM API request must contain a user token. This token is obtained using a register request.

Send a register request containing the CONTROL-M/EM user name and password. Upon successful authentication a new session is created for this user. This session is identified by the returned user token.

When the session is finished, send an unregister request to log off and release resources allocated for this user session.

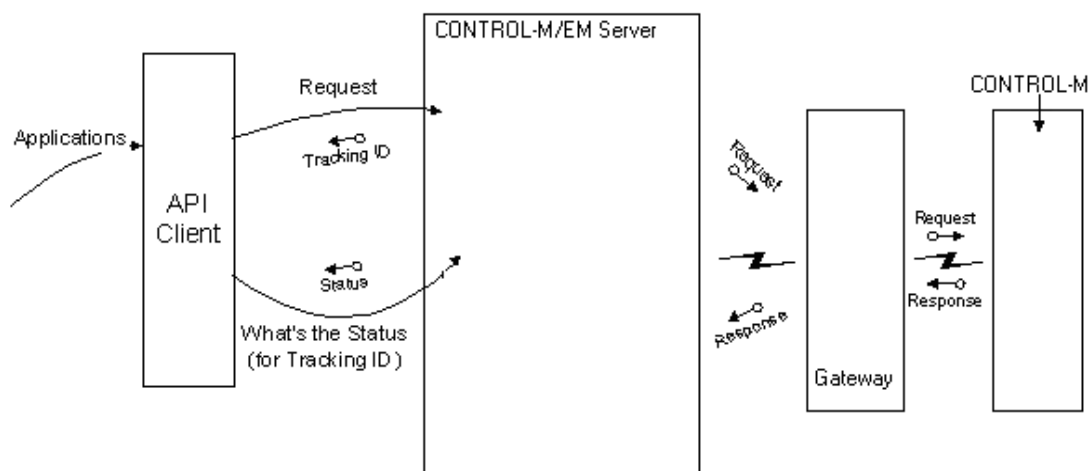
**NOTE**

A program can use multiple user tokens and issue several requests at once (for example, for a multi-threaded program). However, you should not issue another request with the same user token if the previous request is still in session.

CONTROL-M/EM API sessions

The CONTROL-M/EM API enables users of your application to submit requests to a CONTROL-M installation.

Figure 1 CONTROL-M/EM API process flow



To illustrate this concept, an example of a CONTROL-M/EM API session is presented below. The actions of the program user are shown together with the corresponding actions that your program must perform to use the CONTROL-M/EM API.

Session example

The user logs on to a program, submits a Job Creation request to CONTROL-M/EM, and logs off.

To do so, the user performs the following actions:

1. The user logs on to the host computer of the program that uses the CONTROL-M/EM API.
2. The user starts the program that uses the CONTROL-M/EM API.
3. The user logs in to CONTROL-M/EM (sends a Registration request).
4. The user performs an action that results in a Job Creation request (for example) being sent to CONTROL-M/EM.
5. The user receives confirmation that the request was submitted.
6. The user performs additional actions that result in other requests being sent to CONTROL-M/EM.
7. The user logs out of CONTROL-M/EM (sends an Unregistration request).

In reality, more actions are being performed by both the program that the user is using and by CONTROL-M/EM. The user is not aware of these actions, which are described in [Table 1](#).

Table 1 CONTROL-M/EM API (EMXMLInvoker) use flow(part 1 of 2)

Step	User actions	API actions
1	Log in End user logs on to a remote computer to use a program that uses the CONTROL-M/EM API.	None.
2	User starts Client-Side Application End user starts the application.	Program start Required action: <ul style="list-style-type: none"> ■ Initialize the API using the <code>init</code> method of the <code>EMBasicXMLInvoker</code> class. Invoke an instance of the <code>EMBasicXMLInvoker</code> class with the details of the EM GUI server that would respond to the EM/API requests.
3	User Logs in to CONTROL-M/EM End user must supply a username and password.	API (client side) sends Register request Required actions: <ul style="list-style-type: none"> ■ Call the <code>BuildPasswordString()</code> function immediately before sending every API “Register” request because a different string is used for each request in some modes. Submit a register request using the <code>invoke</code> method of the <code>EMBasicXMLInvoker</code> instance created in step 2.

Table 1 CONTROL-M/EM API (EMXMLInvoker) use flow(part 2 of 2)

Step	User actions	API actions
4	None.	CONTROL-M/EM returns response Required action: <ul style="list-style-type: none"> ■ This response contains a unique user token, which should be used in calls for this user. If the Registration request was unsuccessful, the response contains a list of errors.
5	Submit request Required action: <ul style="list-style-type: none"> ■ User requests that a job be submitted to CONTROL-M/EM. 	API (client side) translates request Required action: <ul style="list-style-type: none"> ■ API translates user action into a Job Creation request that is sent to CONTROL-M/EM using the <code>invoke</code> method of the <code>EMBasicXMLInvoker</code> instance created in step 2.
6	None.	CONTROL-M/EM returns response Required actions: CONTROL-M/EM automatically sends either a fault or success response.
7	None.	API (client side) processes response Required actions: <ul style="list-style-type: none"> ■ The program parses the response. If response is an error, notify user ■ If response is actual outcome of the request, notify user of outcome
8	User makes additional requests.	Additional requests can be sent and handled using the same <code>EMBasicXMLInvoker</code> instance obtained in step 2 and the user token obtained in step 4.
9	User Logs off of CONTROL-M/EM	API (client side) sends Unregister request Required actions: <ul style="list-style-type: none"> ■ Submit Unregister request. Use the <code>invoke</code> method, supplying the Unregister request string as a parameter. The request includes the user token obtained in Step 4.
10	None.	CONTROL-M/EM returns response Required action: <ul style="list-style-type: none"> ■ CONTROL-M/EM sends a response indicating either a successful or failed unregister request.
11	None.	Program stop Required action: <ul style="list-style-type: none"> ■ Stop the API using the <code>done</code> method.

Installation

This chapter presents the following topics:

Compatibility	31
CONTROL-M product support.	31
Prerequisites	32
Installing the CONTROL-M/EM API.	33
Configuration	34

The CONTROL-M/Enterprise Manager API is supplied as a compressed file. When decompressed, this file creates a directory structure containing the CONTROL-M/EM API files.

NOTE



To upgrade from previous versions of the CONTROL-M/EM API, follow the instructions in Chapter 3, “Upgrade from earlier versions”.

Compatibility

This section provides information about software compatibility for the CONTROL-M/EM API.

CONTROL-M product support

The CONTROL-M/EM API is fully compatible with the following version of CONTROL-M/EM:

- CONTROL-M/EM version 6.4.01

The CONTROL-M/EM API is fully compatible with the following versions of CONTROL-M:

- CONTROL-M/Server for UNIX and Windows version 6.4.0x
- CONTROL-M for z/OS version 6.3.0x
- CONTROL-M/Server for UNIX and Windows version 6.3.0x
- CONTROL-M/Server for Windows version 6.2.0x
- CONTROL-M/Server for UNIX version 6.2.0x
- CONTROL-M for OS/390 and z/OS version 6.2.0x
- CONTROL-M/Server for Windows version 6.1.0x
- CONTROL-M/Server for UNIX version 6.1.0x
- CONTROL-M for OS/390 and z/OS version 6.1.0x

NOTE

CONTROL-M/Server for UNIX and Windows version 6.4.01 is expected to be available shortly after the release of CONTROL-M/EM 6.4.01.

Prerequisites

- 1 Before installing the CONTROL-M/EM API, ensure that the following requirements are met:

- CONTROL-M/EM is installed on your network.
- One of the following is installed on the computer hosting your project's working directory:
 - Java Developer's Kit (JDK) version 1.4.x or later
 - Java Runtime Environment (JRE) version 1.4.x or later

- 2 The **JAVA_HOME** environment variable points to the JDK (or JRE) directory/library on the computer hosting your project's working directory.

NOTE

JAVA_HOME refers to the directory where the Java 2 Runtime Environment (JRE) was installed. The Java 2 SDK (also called the JDK) contains the JRE, but at a different level in the file hierarchy. For example, if the Java 2 SDK or JRE was installed in **/home/user1**, **JAVA_HOME** would be:

/home/user1/jre1.4.x	[JRE]
/home/user1/jdk1.4.x/jre	[SDK]

Installing the CONTROL-M/EM API

The CONTROL-M/EM API can be installed in any location from which you can connect, using TCP/IP, to the selected CONTROL-M/EM installation.



NOTE

- Because of Java limitations, BMC Software recommends that you do not install the CONTROL-M/EM API in a directory with a path that contains spaces or other special characters.
- If you are installing the CONTROL-M/EM API on an account where an earlier version of CONTROL-M/EM API is installed, review [Chapter 3, “Upgrade from earlier versions”](#) before continuing.

On Microsoft Windows

Unzip the **emapi-640-nt.zip** file from the `cdPath\tools\emapi\` directory on the CONTROL-M/EM installation CD to any location.

The **emapi-640** directory tree is created. All CONTROL-M/EM API files are located under this directory.

The primary subdirectories are created under the `emapi-640` directory. They are described in [Table 2](#).

On UNIX

■ For Solaris, AIX, HP-UX or HP-UX Itanium:

Use the following command to open and uncompress the **emapi-640-UNIX.TAR.Z** compressed tar file in the `cdPath/TOOLS/EMAPI_FILES/` directory on the CONTROL-M/EM installation CD to any directory:

```
uncompress -c cdPath/TOOLS/EMAPI_FILES/emapi-640-UNIX.TAR.Z | tar xvf -
```

The **emapi-640** directory tree is created. All CONTROL-M/EM API files are located under this directory.

The following primary subdirectories are created under the `emapi-640` directory. They are described in [Table 2](#).

■ On RedHat or Suse:

Use the following command to open and uncompress the **emapi-640-UNIX.TAR.gz** compressed tar file in the *cdPath/TOOLS/EMAPI_FILES/* directory on the CONTROL-M/EM installation CD to any directory:

```
gunzip -c cdPath/TOOLS/EMAPI_FILES/emapi-640-UNIX.TAR.gz | tar xvf -
```

The **emapi-640** directory tree is created. All CONTROL-M/EM API files are located under this directory.

The primary subdirectories are created under the **emapi-640** directory. They are described in [Table 2](#).

Table 2 CONTROL-M/EM API primary subdirectories

Directory	Description
emapi-640\etc	Contains the jacorb.properties file required for JacORB runtime (CORBA Java client library).
emapi-640\keystore	Contains the emapi.keystore demo file required for JacORB runtime when using SSL. For more information, see the <i>CONTROL-M SSL Guide</i> .
emapi-640\classes	Contains the EMAPI libraries and third-party libraries that include the following files: <ul style="list-style-type: none"> ■ jacorb.jar Java implementation of CORBA (refer to the JacORB copyright/license in this guide). ■ NamingViewer.jar Java library used by the NamingViewer utility described on page 35. ■ emapi.jar The API implementation files. ■ log4j-1.2.8.jar Files used by the API logging facility.
emapi-640\xmldata	Contains the schema files that validate the XML formatted request strings sent and received using the API.
emapi-640\examples	Contains sample implementations of the API.

Configuration

The processes for configuring the CONTROL-M/EM API on computers running Microsoft Windows and on computers running UNIX are identical, except for differences in the names of some of the files used.

These differences, and the names of other files that are used during configuration, are described in [Table 3](#).

Table 3 Configuration files

Name	Description
emapi-configure.bat (Windows)	Creates configuration files and enables the API to access other key components.
emapi-configure (UNIX)	These components include: <ul style="list-style-type: none"> ■ CORBA Naming Service ■ CONTROL-M/EM GUI Server ■ Global Alerts Server.
emapi_env.bat (Windows)	Automatically adds API directory pathnames to the CLASSPATH environment variable.
emapi_env.sh (UNIX, users working in Korn Shell environment)	Note: The emapi-configure(.bat) file creates this file when it is run.
emapi_env.csh (UNIX, users working in C Shell and TC Shell environments)	
ctmemapi.properties	Contains the locations of: <ul style="list-style-type: none"> ■ CONTROL-M/EM GUI Server ■ Global Alerts Server ■ XMLDATAPATH property Note: This file is created by the emapi-configure(.bat) file when it is run.
emapi-admin.bat (Windows)	Starts an interactive utility for changing the hostnames of the CONTROL-M/EM GUI Server and Global Alerts Server.
emapi-admin (UNIX)	
NamingViewer.vbs (Windows)	Starts a Java GUI utility that displays the Naming Service repository graphically. For more information, see the <i>CONTROL-M Administrator Guide</i> .
NamingViewer (UNIX)	
changePass.bat (Windows)	Encrypts the keystore password for demo certificates. This utility is located in the emapi_root directory. For more information, see the <i>CONTROL-M SSL Guide</i> .
changePass (UNIX)	

1 Locate CONTROL-M/EM components and obtain the following information:

- The CORBA Naming Service host name and port
- Host names of the CONTROL-M/EM GUI Server and Global Alerts Server

2 Change the current directory to the CONTROL-M/EM API client's home directory, `emapi-640`.

3 Run the **emapi-configure.bat** file
(UNIX: **emapi-configure**).

At the prompt, the following text is displayed:

```
This script configures the CONTROL-M/EM API library
Execute 'emapi-configure -h' for more details.
Press any key to begin. [continue]:
```

4 Press any key.

- If CONTROL-M/EM is not installed on this computer, go to [step 5](#).
- If CONTROL-M/EM is installed on the computer, the following prompt is displayed:

```
CONTROL-M/Enterprise Manager configuration detected
-----
CONTROL-M/Enterprise Manager configuration detected. Do you want to use the same
configuration? [y/n]
```

- Press **y** to use the current CORBA configuration properties.
- Press **n** to supply new CORBA configuration information.

5 At the following prompt, enter the name of the CORBA Naming Service host computer and press **Enter**.

```
Naming Service hostname
-----
Enter the Naming Service hostname (press Enter for default) [TLW2K122]:
```

6 At the following prompt, enter the port number that the CORBA Naming Service uses and press **Enter**.

```
Naming Service port
-----
Enter the Naming Service port (press Enter for default) [3075]:
```

7 At the following prompt, enter the name of the CONTROL-M/EM GUI Server host computer and press **Enter**.

```
CONTROL-M/Enterprise Manager - GUI Server hostname
-----
Enter the GUI Server hostname (press Enter for default) [TLW2K122]:
```

- 8** At the following prompt, enter the name of the CONTROL-M/EM Global Alerts Server host computer and press **Enter**.

```
Control -M/Enterprise Manager - Global Alerts Server hostname
-----
Enter the Global Alerts Server hostname (press Enter for default) [TLVW2K122]:
```

Installation is complete.

You are now ready to prepare your project and its environment to use the CONTROL-M/EM API. These issues are discussed in [Chapter 4, “Configure the CONTROL-M/EM API.”](#)

Upgrade from earlier versions

This chapter provides instructions for upgrading to CONTROL-M/Enterprise Manager API version 6.4.01.

To upgrade the CONTROL-M/EM API:

- 1 Install the CONTROL-M/EM API version 6.4.01. For more information, refer to [Chapter 2, “Installation.”](#)
- 2 Perform the configuration steps described in [Chapter 4, “Configure the CONTROL-M/EM API.”](#)
- 3 Set your environment using the provided `emapi_env.bat` (`emapi_env.sh` on UNIX) script.



NOTE

If you do not use the provided script, ensure that the startup script and the CLASSPATH point to the new classes.

- 4 Update your applications and XML requests.

For information on the required changes see [“Upgrade considerations.”](#)

- 5 (optional) Remove the `emapi-6xx` directory from the previous version.

Upgrade considerations

The CONTROL-M/EM migration procedure (described in the *CONTROL-M Installation Guide* and *CONTROL-M Migration Guide*) does not upgrade the CONTROL-M/EM API. Install and configure the CONTROL-M/EM API separately after the migration.



WARNING

Do not copy the CONTROL-M/EM API version 6.4.01 files directly over the previous installation (such as, in the **emapi-610** directory). This may cause unpredictable behavior.

From version 6.3.0x to version 6.4.01

Code developed for CONTROL-M/EM API version 6.3.0x is compatible with version 6.4.01 except for the following:

- The `encodePassword` method of `EMXMLInvoker` and `EMBasicXMLInvoker` is not supported in CONTROL-M/EM API version 6.4.01. Replace all calls to the `encodePassword` method with calls to the `BuildPasswordString` method. For more information see “[BuildPasswordString](#)” on page 73.
- CONTROL-M/EM API version 6.4.01 includes the following new parameters that were added to CONTROL-M/Server version 6.4.01.
 - `cyclic_type`
 - `interval_sequence`
 - `specific_times`
 - `tolerance`
 - `attach_sysout`



NOTE

To use the new parameters, set the `xml ns: ctmem` namespace declaration to `http://www.bmc.com/ctmem/schema640`.

You can use these parameters when submitting the following requests:

- `request_create_adj`
- `request_def_create_jobs`
- `request_create_sched_group`

For more information on the parameters and requests, see [Chapter 7, “Request reference,”](#) and [Appendix C, “Job and Scheduling Group XML parameters.”](#)

From version 6.2.0x to version 6.3.01

- The structure of request and response messages was changed as follows:
 - references to .dtd files removed
 - the `ctmem:message_package` envelope is replaced by SOAP-ENV envelope (see [“SOAP Envelope for CONTROL-M/EM requests and responses”](#) on page 89.)
 - addition of a new SOAP:Fault response, which differentiates responses denoting complete failure from responses containing valid data.
 - the `ctmem:user_token` appears as a child element of a request tag. Previously, it appeared as an attribute of that tag.
 - the `ctmem:name` attribute of request and response elements containing the operation name is removed; request and response tag names are built by prefixing the operation name with the following strings:
 - `request_`
 - `response_`
 - `fault_`
 - `request_poll_`
 - `response_poll_`
 - `fault_poll_`

Example of an XML request using the previous format:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE ctmem:message_package SYSTEM "EMAPI_add_delete_condition.dtd">
<ctmem:message_package xmlns:ctmem="http://www.bmc.com/ctmem/schema" ctmem:version="1.0">
  <ctmem:request ctmem:name="add_condition" ctmem:user_token="575668226">
    <ctmem:control_m>CTM01</ctmem:control_m>
    <ctmem:condition>
      <ctmem:name>Cond01</ctmem:name>
      <ctmem:odate>0101</ctmem:odate>
    </ctmem:condition>
  </ctmem:request>
</ctmem:message_package>
```

Example of an XML request using the current format:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_add_condition xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>575668226</ctmem:user_token>
      <ctmem:control_m>CTM01</ctmem:control_m>
      <ctmem:condition>
        <ctmem:name>Cond01</ctmem:name>
        <ctmem:odate>0101</ctmem:odate>
      </ctmem:condition>
    </ctmem:request_add_condition>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

From version 6.1.03 to version 6.2.01

The CORBA implementation used in CONTROL-M/EM API was changed from IONA to JacORB. Therefore, set the property `org.omg.CORBA.ORBClass` to `org.jacorb.orb.ORB` and the property `org.omg.CORBA.ORBSingletonClass` to `org.jacorb.orb.ORBSingleton`.

Configure the CONTROL-M/EM API

This chapter presents the following topics:

Preparing your project environment.	43
Writing your project	44
Running your project.	47

You must perform a number of steps in your project to prepare it to use the CONTROL-M/EM API.

Preparing your project environment

Prior to using the CONTROL-M/EM API, you must configure the API for use in your project development environment.

When you want to release your application for use, you must configure the CONTROL-M/EM API running environment.



NOTE

Your project is in the development environment when you are creating and testing the project on your development computer.

The running environment is the project environment when your application is released for use.

Follow these steps for both the development environment and the running environment.

- 1 Set the environment variables for the environment by running the **emapi_env.bat** (or **emapi_env.sh**) file.

For **emapi_env.sh** on UNIX, run the following command:

```
. emapi_env.sh
```

Running the command with the above syntax ensures that the variables are valid after the command is run.

- 2 Copy the **emapi-640\ctmemapi.properties** file to your project's working directory.

Writing your project

Certain steps must be performed in your project to enable it to make use of the CONTROL-M/EM API.

The following procedure describes a class that uses the CONTROL-M/EM API. You can use this class as a basis for building your project.

A sample class is displayed in [Figure 2](#).

Figure 2 Sample class

```

import com.bmc.ctmem.emapi.*;
public class EMAPISample {
    public EMAPISample() {

    }

    /** run once before submitting requests */
    public void do_init(String[] args) {
        EMXMLInvoker.init(args);
    }

    /** run once before exiting the program */
    public void do_terminate() {
        EMXMLInvoker.done();
    }

    /** This submits the XMLRequest received as a parameter
     * and returns the response */
    public String submit_request(String XMLRequest) {
        String XMLResponse="";
        // Creates a component
        ComponentType gsr_comp = new GSRCComponent();
        // Creates a new EMXMLInvoker instance
        EMXMLInvoker my_invoker = new EMXMLInvoker(gsr_comp);
        try {
            // Submits the request given as a parameter
            XMLResponse = my_invoker.invoke(XMLRequest);
        }

        catch(InvokeException i) {
            // must handle InvokeException
        }

        return XMLResponse;
    }
}

```

- 1** Import the CONTROL-M/EM API into your project with the following command:

```
import com.bmc.ctmem.emapi.*;
```

- 2** Create a class that uses the CONTROL-M/EM API functionality that you want to employ in your project.

The first thing this class must do is call the init method. This method is described in [“Initializing and stopping the CONTROL-M/EM API services” on page 61.](#)

— **EXAMPLE** —

```
EMXMLInvoker.init();
```

- 3 Select the CONTROL-M/EM component with which the project will communicate (the CONTROL-M/EM GUI Server or the Global Alerts Server).

— **EXAMPLE** —

GUI Server

```
ComponentType gsr_comp = new GSComponent();
```

For more information, see [“Submitting a request using the EMXMLInvoker class” on page 62.](#)

- 4 Create an **EMXMLInvoker** instance and submit a request.

— **EXAMPLE** —

```
EMXMLInvoker my_invoker = new EMXMLInvoker(gsr_comp);
try {
    XMLResponse = my_invoker.invoke(XMLRequest);
}
catch(InvokeException i) {
}
```

For more information, see [“Submitting a request using the EMXMLInvoker class” on page 62.](#)

The class is now ready. You can use it in your project.

- 5 Compile your project.

To run your project, you must pass it the relevant Java Virtual Machine variables. For more information, see [“Running your project” on page 47.](#)

Running your project

- 1 Set the environment variables for the CONTROL-M/EM API by running the **emapi_env.bat** (or **emapi_env.sh**) file.

For **emapi_env.sh** on UNIX, run the following command:

```
. emapi_env.sh
```

Running the command with the above syntax ensures that the variables are valid after the command is run.

- 2 Ensure that the environment variables are set for the Java environment and that the path includes the Java installation location.

If they are not set, run the following command:

For Microsoft Windows

```
set JAVA_HOME=java_installation_location
set PATH=java_installation_location;%PATH%
```

For UNIX

```
setenv JAVA_HOME java_installation_location
setenv PATH java_installation_location:$PATH
```

- 3 Copy the **emapi-640\ctmemapi.properties** file to your project working directory.
- 4 Run the **java** command from the project working directory, using the following CORBA parameters:

NOTE



These parameters are needed by the CONTROL-M/EM API. Pass them to your project as the first and second runtime parameters. The command must be entered as a single line. It is divided here to fit on the page.

For Microsoft Windows

```
java.exe -Dorg.omg.CORBA.ORBClass=org.jacorb.orb.ORB
-Dorg.omg.CORBA.ORBSingletonClass=org.jacorb.orb.ORBSingleton -classpath
%CLASSPATH% projectMainClass
```

For UNIX

```
java -Dorg.omg.CORBA.ORBClass=org.jacorb.orb.ORB
-Dorg.omg.CORBA.ORBSingletonClass=org.jacorb.orb.ORBSingleton -classpath
$CLASSPATH projectMainClass
```

projectMainClass is your project main class.

Optionally, you can pass these parameters using one of the alternative methods described in [“Java virtual machine parameters” on page 227](#).

Issuing requests and handling responses

This chapter presents the following topics:

Request types	49
Response types	50
CONTROL-M/EM API programming methods	54

The CONTROL-M/EM API is a set of Java classes that manipulate an existing set of functions in the server side of the API. These classes enable Java developers to send requests to CONTROL-M/EM server components from within the developers' applications.

Request types

The CONTROL-M/EM API supports two types of requests:

- *Synchronous* requests are processed by CONTROL-M/EM. Responses are received directly from CONTROL-M/EM.
- *Asynchronous* requests are sent through CONTROL-M/EM to CONTROL-M/Server. These requests receive a tracking ID as a response. By sending a polling request that includes this tracking ID, the user can retrieve feedback about what action CONTROL-M/EM takes, or a notice that the request is still being processed.

Client programs using the EMBasicXMLInvoker do not require the polling process, as polling is done by the class itself. For more information, see [“CONTROL-M/EM API programming methods” on page 54](#).

[Table 4](#) shows the synchronous and asynchronous requests.

Table 4 Request types

Request type	Request name
Synchronous	<ul style="list-style-type: none"> ■ polling requests ■ user registration request ■ check user token validity ■ user unregistration request ■ keep alive request ■ alert status modification request ■ job tracking request ■ create job and scheduling group definitions ■ delete job definitions ■ retrieve jobs in active environment ■ retrieve BIM Services list
Asynchronous	<ul style="list-style-type: none"> ■ Order or Force request ■ add or delete condition request ■ job creation request ■ upload scheduling table ■ job actions in active environment

Response types

Every time a request is made, CONTROL-M/EM replies by sending one or more response strings. You can parse the contents of these response strings for use in your project.

CONTROL-M/EM API responses are wrapped by the SOAP envelope. (For more information, see [“SOAP Envelope for CONTROL-M/EM requests and responses” on page 89](#).) The responses are divided into the following categories:

- Successful responses
- Fault responses

Successful responses

Successful responses appear directly below the SOAP-ENV:Body node. They contain the information requested or indicate success. Every request has a specific ctmem response. For example, ctmem:response_unregister is the response for a ctmem:request_unregister request.

```

<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    . . . SPECIFIED_REQUEST_RESPONSE . . .
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Fault responses

Fault responses report failed requests. The fault node is located below the SOAP detail node. The fault node contains information about the errors that caused the operation to fail. Every request has a specific fault response. For example, `ctmem:fault_unregister` is the fault response for the `ctmem:request_unregister` request.

```

<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>... </faultcode>
      <faultstring>.... </faultstring>
      <detail>
        . . . FAULT_RESPONSE_OF_THE_REQUEST . . .
      </detail>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```



NOTE

When the CONTROL-M/EM API is running a request that applies to multiple jobs, the request might succeed on some jobs and fail on others. In this case, the general **status** field returned in a **successful** response, receives a value of **PARTIAL_SUCCESS** instead of **OK**. Each individual job status is either **OK** or **ERROR**. A **PARTIAL_SUCCESS** can only occur when the CONTROL-M/EM API runs the following requests:

- track job
- order or force job

Response types received when using EMBasicXMLInvoker

Table 5 shows the types of responses you receive when using the EMBasicXMLInvoker.

Table 5 EMBasicXMLInvoker response types

Response type	Description
<i>response_operationName</i> Example: <i>response_unregister</i>	<p>This response is the final result of a successful synchronous request.</p> <p>The first field is status and the value is OK except for <i>response_track_jobs</i> where it can also be PARTIAL_SUCCESS. For more information, see the note on page 51.</p>
<i>fault_operationName</i> Example: <i>fault_unregister</i>	<p>For synchronous requests, this indicates a failure.</p> <p>For asynchronous requests, this response indicates a failure in sending the request to the CONTROL-M/Server. For example, if the user is not authorized, CONTROL-M/Server is unavailable or does not have the right status, or the request has invalid values.</p>
<i>response_poll_operationName</i> Example: <i>response_poll_add_condition</i>	<p>This response is a final successful, or partially successful, response for an asynchronous request.</p> <p>The first field is status and the value is OK except for <i>response_poll_order_force</i> where it can also be PARTIAL_SUCCESS. For more information, see the note on page 51.</p>
<i>fault_poll_operationName</i> Example: <i>fault_poll_add_condition</i>	<p>This response is the result of an error on an asynchronous request that was sent to the CONTROL-M/Server. This response can be an error returned from the CONTROL-M/Server itself (for example, an order request for a nonexistent scheduling table), or an error from the communication layer between CONTROL-M/EM and CONTROL-M/Server (for example, a request timeout or communication loss).</p>

Response types received when using EMXMLInvoker

Table 6 shows the types of responses you receive when using the EMXMLInvoker.

Table 6 EMXMLInvoker response types

Response type	Description
<i>response_operationName</i> Example: <i>response_unregister</i>	<p>This response is the final result of a successful synchronous request.</p> <p>The first field is status and the value is OK except for <i>response_track_jobs</i> where it can also be PARTIAL_SUCCESS. For more information, see the note on page 51.</p> <p>For asynchronous requests, the response contains the token ID that should be used to poll CONTROL-M/EM for the final result of the request.</p>
<i>fault_operationName</i> Example: <i>fault_unregister</i>	<p>For synchronous requests, this response indicates a failure.</p> <p>For asynchronous requests, this response indicates a failure in sending the request to the CONTROL-M/Server. For example, if the user is not authorized, CONTROL-M/Server is unavailable or does not have the right status, or the request has invalid values.</p>
<i>response_poll_operationName</i> Example: <i>response_poll_add_condition</i>	<p>This response is received as an answer to a polling request.</p> <p>The first field is status.</p> <p>A value of EXEC indicates that the request is still being processed by the CONTROL-M/Server and the response has not yet arrived.</p> <p>A value of either OK or PARTIAL_SUCCESS indicates the final response of an asynchronous request.</p> <p>Note: The value PARTIAL_SUCCESS can appear only in <i>response_poll_order_force</i>. For more information, see the note on page 51.</p>
<i>fault_poll_operationName</i> Example: <i>fault_poll_add_condition</i>	<p>This is an error response received from a polling request.</p> <p>This response can be an error returned from the CONTROL-M/Server itself (for example, an order request for a nonexistent scheduling table), or an error from the communication layer between CONTROL-M/EM and CONTROL-M/Server (for example, a request timeout or communication loss).</p>

CONTROL-M/EM API programming methods

The CONTROL-M/EM API supports two methods for creating, sending, and handling requests:

- EMXMLInvoker uses the CONTROL-M/EM API to send and receive XML requests using the CONTROL-M/EM API. The `init`, `done`, `invoke`, `getProperties`, and `setProperties` methods are used with this class.
- EMBasicXMLInvoker eliminates the effort of polling by not returning the response until either the final response is available from the server, or the API times out.

Deciding which method to use

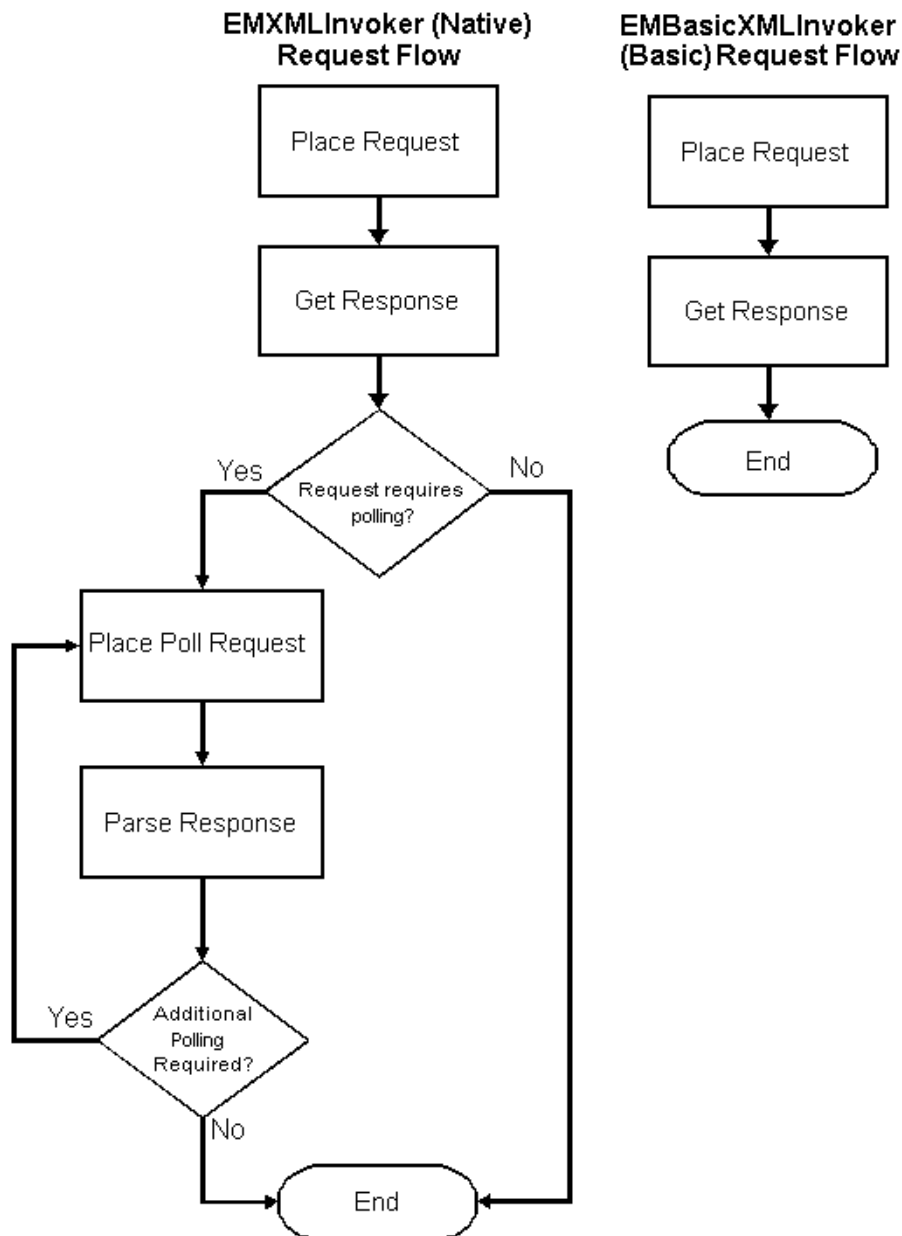
The EMXMLInvoker and EMBasicXMLInvoker classes can be used together in the same project, and even in the same session. However, you must decide which class is most suitable for your purposes. Because the EMBasicXMLInvoker class inherits from the EMXMLInvoker class you need to initialize the CONTROL-M/EM API only once.

The two classes differ as follows:

- The EMXMLInvoker requires you to poll manually for responses (see [“When to use the EMXMLInvoker class” on page 59](#)). The EMBasicXMLInvoker polls automatically for responses (see [“When to use the EMBasicXMLInvoker class” on page 55](#)).
- The EMBasicXMLInvoker uses an XML schema to validate XML-formatted requests and responses, but the EMXMLInvoker class does not (see [“XML string validation” on page 65](#)).

Figure 3 illustrates the structural differences between the two request methods.

Figure 3 Differences in request methods



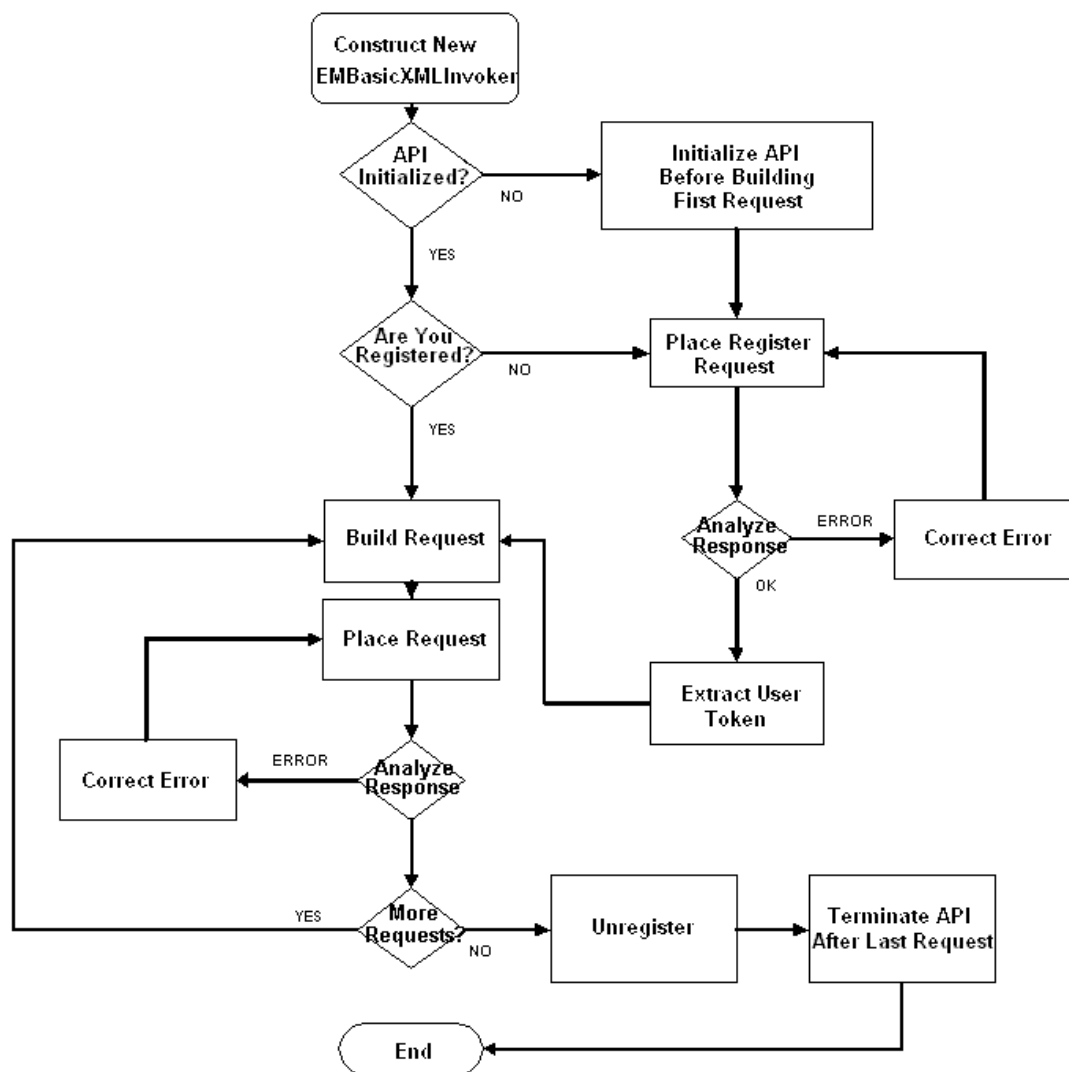
When to use the EMBasicXMLInvoker class

The EMBasicXMLInvoker class eliminates the need for polling by not returning the response until either the final response is available from the server, or the API times out. The advantage is that the user does not have to process the responses or place additional requests.

However, using only the EMBasicXMLInvoker class has the following disadvantages:

- The API call might be blocked for several seconds, generally until the final response is available or the call times out. The time out period is configurable.
- After receiving the response, the EMBasicXMLInvoker class performs additional processing to determine its course of action.
- Because the EMBasicXMLInvoker class performs XML parsing, this class makes it more resource intensive than the EMXMLInvoker class.

Figure 4 EMBasicXMLInvoker request session



Using EMBasicXMLInvoker class calls

The `init` and `done` methods for the `EMBasicXMLInvoker` class are used for starting and stopping the API. These methods are identical to those of the `EMXMLInvoker` class. For more information, see “[Initializing and stopping the CONTROL-M/EM API services](#)” on page 61.

The `invoke` method returns a string that contains a text response from CONTROL-M/EM in XML format.

Submit a request using the EMBasicXMLInvoker class

`invoke` is the method of `EMBasicXMLInvoker` used to send requests to CONTROL-M/EM. The `invoke` method accepts one parameter (in `String` type), written in XML format. This parameter contains the request.

Specifying a component to process the request

Before invoking an object with an instance of the `EMBasicXMLInvoker`, you must assign to it a reference to the CONTROL-M/EM component that will process the request. An instance of `EMBasicXMLInvoker` must hold either `GSRComponent` or `GASComponent`.

Example

```
ComponentType myComponent = new GSRComponent();

EMBasicXMLInvoker gsInvoker = new EMBasicXMLInvoker(myComponent);
```

The `GSRComponent` class represents the CONTROL-M/EM GUI Server component in your network. The `GASComponent` class represents the CONTROL-M/EM Global Alerts Server component in your network.

The component that you specify when you invoke the object (`GSRComponent` or `GASComponent`) must be appropriate to the task that you want to perform. Request types and the components that process those requests are listed in [Table 7](#).

NOTE



Identifying information for these components was recorded in the `ctmemapi.properties` file when you configured the API. This file contains the host name for each component.

In networks in which more than one GUI Server or GAS is installed, only one of each component is listed in the `ctmemapi.properties` file. The API works only with the components listed in the file. You cannot modify the file to include more than one GUI Server or GAS.

Using the `invoke` method

Prototype

```
public String invoke(String xml Request) throws InvokeException
```

In this statement, `xml Request` is a string that is formatted according to the XML request specification that is presented in this book.

A request in XML format (`xml Request`), specifying the action that the CONTROL-M installation is to perform, is required for each call.

To invoke an object with the `EMBasicXMLInvoker`:

- 1 Create an instance of the `EMBasicXMLInvoker` containing a reference to a CONTROL-M/EM server component type. This specified server component type component processes the request that you create in [step 2](#).

Example

```
GSRComponent gsrComponent = new GSRComponent();rw\
EMBasicXMLInvoker gsrInvoker = new EMBasicXMLInvoker(gsrComponent);
```

- 2 Send a request to the specified CONTROL-M/EM server component.

Example

```
String xml Request = "<?xml?>..."; // xml request
String xml Response;
try{
    xml Response = gsrComponent.invoke(xml Request);
}
catch(InvokeException ex){
    // handle invoke failures
}
// handle xml response
```

NOTE



The `invoke` method can throw an exception if an application fails to process the `invoke` call (for example, if communication between CONTROL-M/EM and CONTROL-M fails). For more information, see [“Application runtime and communication troubleshooting” on page 229](#).

When to use the EMXMLInvoker class

The EMXMLInvoker class sends XML-formatted requests to CONTROL-M/EM and returns responses to the program that issued the requests. This class does not process the response data.

Certain requests (such as the Job Creation request) take time for CONTROL-M to process. As a result, they indicate only if the request has been submitted successfully.

An application using the EMXMLInvoker class might have to make several requests to check whether the expected response is available. To facilitate this, the application sends a poll request.

To identify the original request (for which the polling is being performed) the poll request includes a tracking ID, which is supplied in a response from CONTROL-M/EM immediately after certain requests are submitted. These requests are as follows:

- an Order or Force Request
- an Add or Delete Condition Request
- a Job Creation Request

The program can submit this tracking ID in a poll request several times, until the required response is available.

Figure 5 EMXMLInvoker request session

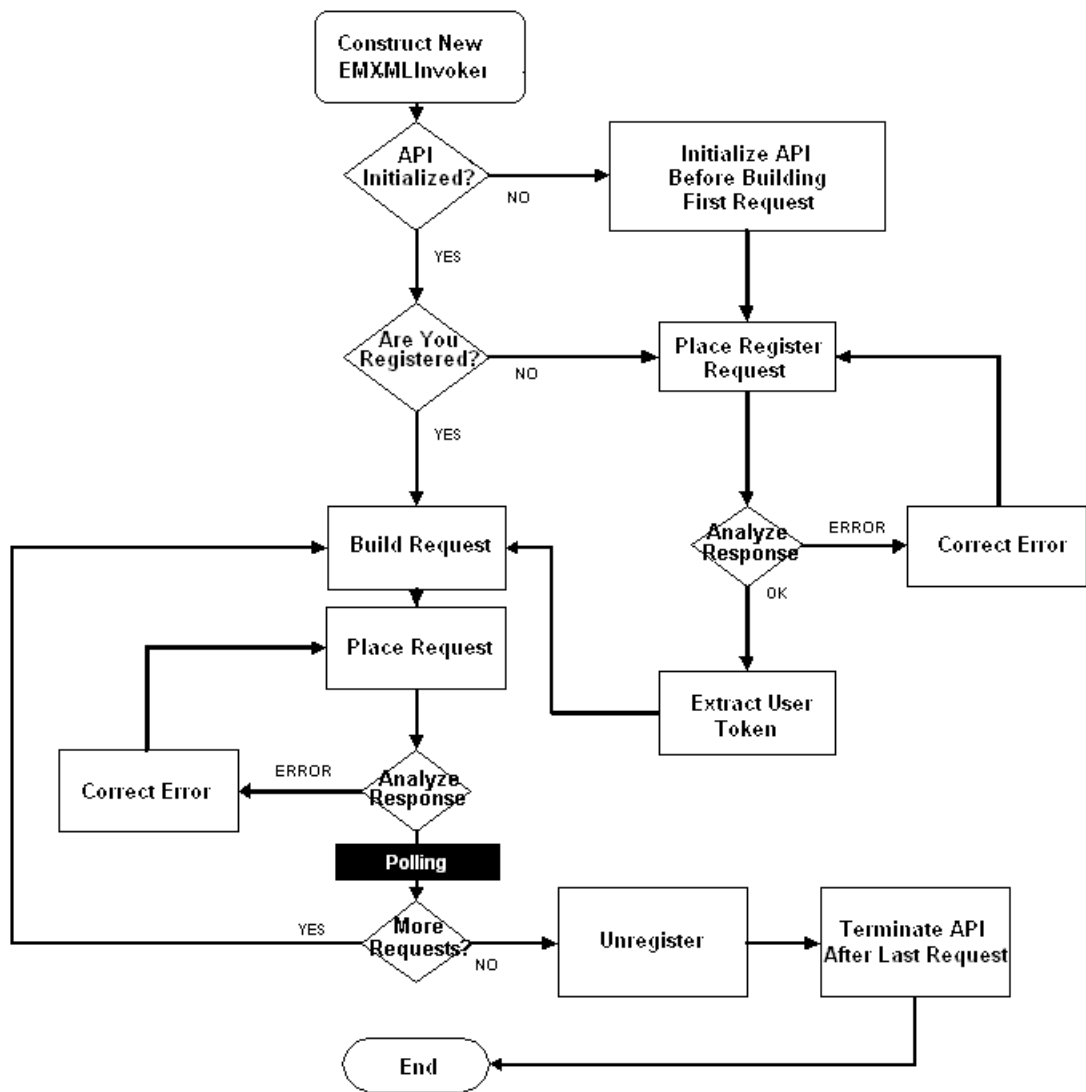
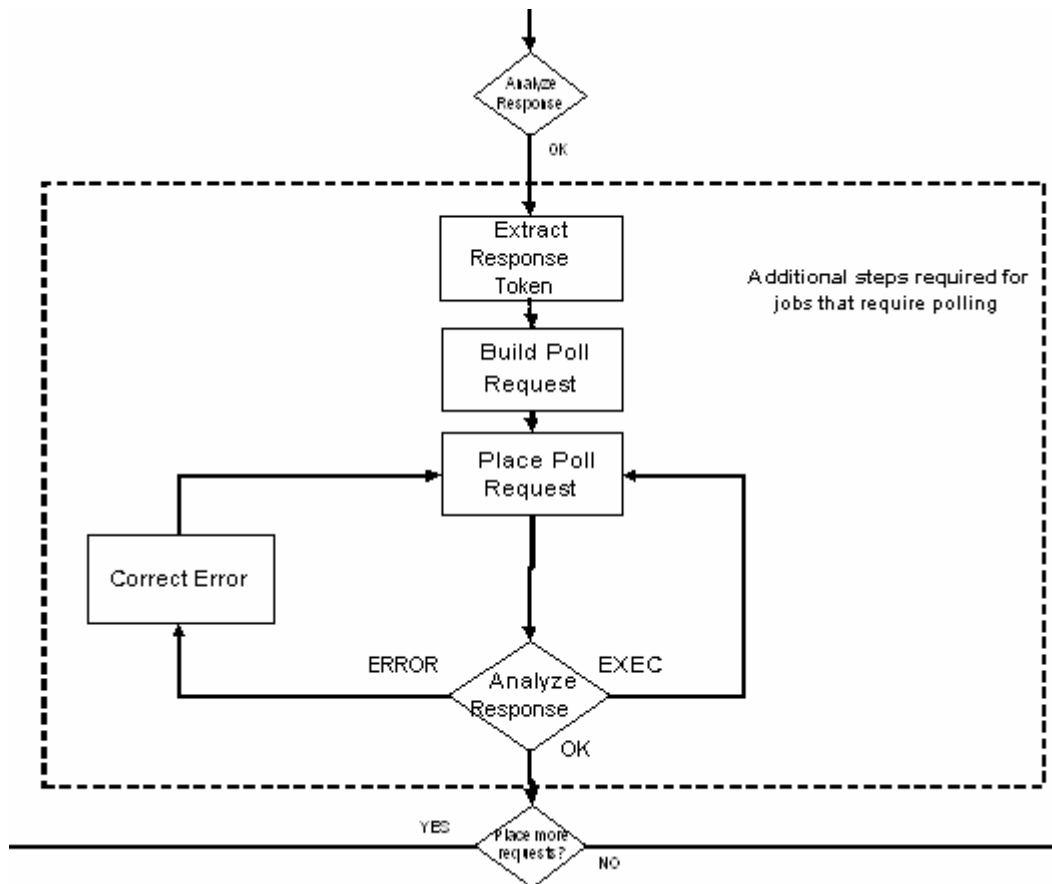


Figure 6 on page 61 illustrates the polling process (indicated by the black box in Figure 5) in detail.

Figure 6 EMXMLInvoker polling request steps



The advantage of the polling process is that the API always replies immediately to the request. No extra time or resources are spent processing the responses.

Using EMXMLInvoker class calls

The EMXMLInvoker uses CONTROL-M/EM API to send and receive XML requests. The `init`, `done`, `invoke`, `getProperties`, and `setProperties` methods are used with this class.

Initializing and stopping the CONTROL-M/EM API services

Before using the CONTROL-M/EM API in your project, you must initialize the API.

You start and stop the CONTROL-M/EM API by initializing the CORBA services. It stops when the CONTROL-M/EM API stops using the CORBA services.

When your application finishes using the API, you can stop the API.



NOTE

The CONTROL-M/EM API uses services that need to be initialized and terminated only once. There is no need to initialize and stop for every new user session or call.

Starting the API services

The API services are started by using the EMXMLInvoker static method, `init`. The prototype described in this section uses a default CORBA configuration and should be used in most circumstances.



NOTE

There are two additional `init` prototypes. They are described in “[Modifying initialization properties](#)” on page 215.

Prototype

```
public static void init()
```

`init` is a method of the EMXMLInvoker.

The `init` method should be called at program startup. It does not need to be run before every request or before every interactive session.

Stopping the API services

When you finish using the CONTROL-M/EM API in your application, the API must be stopped. The API services are stopped using the EMXMLInvoker static method, `done`.

There is only one `done` method prototype:

Prototype

```
public static void done()
```

Submitting a request using the EMXMLInvoker class

`invoke` is a method of EMXMLInvoker used to send requests to CONTROL-M/EM.

The `invoke` method sends a request to a CONTROL-M/EM component that is specified during invocation.

The `invoke` method accepts one parameter (a `String`), written in XML format. This parameter contains the request that the user is submitting to CONTROL-M/EM.

The `invoke` method returns a `String`, containing a text response from CONTROL-M/EM in XML format.

Specifying a component to process the request

Before invoking an object with an instance of the `EMXMLInvoker`, you must assign to it a reference to the CONTROL-M/EM component that processes the request. An instance of `EMXMLInvoker` must hold either `GSRComponent` or `GASComponent`.

Example

```
ComponentType myComponent = new GSRComponent();

EMXMLInvoker gsInvoker = new EMXMLInvoker(myComponent);
```

- The `GSRComponent` class represents the CONTROL-M/EM GUI Server component in your network.
- The `GASComponent` class represents the CONTROL-M/EM Global Alerts Server component in your network.

The component you specify when you invoke the object (`GSRComponent` or `GASComponent`) must be appropriate to the task that you want to perform. Request types and the components that process those requests are listed in [Table 7](#).

Table 7 Request types listed by component type

Request type	GUI Server (GSRComponent)	Global Alerts Server (GASComponent)
Job Creation	X	
Order/Force	X	
Add/Delete Condition	X	
Alert Status Modification		X
Polling	X	
User Registration	X	X
User Unregistration	X	X
Time-out Reset	X	X
Job Tracking	X	

**NOTE**

Identifying information for these components was recorded in the **ctmemapi.properties** file when you configured the API. This file contains the hostname for each component.

In networks in which more than one GUI Server or GAS is installed, only one of each component is listed in the **ctmemapi.properties** file. The API works only with the components listed in the file. You cannot modify the file to include more than one GUI Server or GAS.

Using the `invoke` method**Prototype**

```
public String invoke(String xmlRequest) throws InvokeException
```

In this statement, `xmlRequest` is a string that is formatted according to the XML request specification that is presented in this book.

A request in XML format (`xmlRequest`), specifying the action that the CONTROL-M installation is to perform, is required for each call. The various types of requests that you can make are described in [Chapter 7, “Request reference.”](#)

To invoke an object with the `EMXMLInvoker`:

- 1 Create an instance of the `EMXMLInvoker` containing a reference to a CONTROL-M/EM server component type. This specified server component type component processes the request that you create in [step 2](#).

Example

```
GSRComponent gsrComponent = new GSRComponent();
EMXMLInvoker gsrInvoker = new EMXMLInvoker(gsrComponent);
```

- 2 Send a request to the specified CONTROL-M/EM server component.

Example

```
String xmlRequest = "<?xml?>..."; // xml request
String xmlResponse;
try{
    xmlResponse = gsrComponent.invoke(xmlRequest);
}
catch(InvokeException ex){
    // handle invoke failures
}
// handle xml response
```




NOTE

The `invoke` method can throw an exception if the application fails to process the `invoke` call (for example, if communication between CONTROL-M/EM and CONTROL-M fails). For more information, see [“Application runtime and communication troubleshooting”](#) on page 229.

XML string validation

The `EMBasicXMLInvoker` uses an XML schema to validate XML-formatted request and response strings.

The location of this schema is determined by the `com.bmc.ctmem.emapi.XMLDATAPATH` property. If this path is not available to the class at run time, the operation fails.



NOTE

The `XMLDATAPATH` property is defined in the `ctmemapi.properties` file.

The `EMXMLInvoker` class does not examine the XML-formatted data that it sends or receives, and it does not depend on the presence of the XML schema.

An `EMXMLInvoker` request session adds a series of polling request steps to the `EMBasicXMLInvoker` request session that is displayed in [Figure 4](#) on page 56.

The XML standard does not include support for the following characters. If these characters are used, they are translated in the XML file as listed in the following table:

Character	Meaning	Translated to
<	less than	<
>	greater than	>
&	ampersand	&
“	quotation marks	"
‘	apostrophe	'
ASCII 10	line feed	

ASCII 13	carriage return	

For information about using the `EMXMLInvoker`, see [“Using EMXMLInvoker class calls”](#) on page 61.

For information about using the `EMBasicXMLInvoker`, see [“Using EMBasicXMLInvoker class calls”](#) on page 57.

Class reference

This chapter presents the following topics:

ComponentType class	68
EMBasicXMLInvoker class	69
invoke	70
setPollRequestIntervalMilli	71
setPollRequestTimeoutMilli	71
EMXMLInvoker class	72
BuildPasswordString	73
done	73
getProperties	74
init	75
invoke	77
setProperties	77
GASComponent class	78
GASComponent (Prototype 1)	79
GASComponent (Prototype 2)	79
GSRComponent class	80
GSRComponent (Prototype 1)	80
GSRComponent (Prototype 2)	81
InvokeException class	81
getMajorCode	82
getMinorCode	82
getReason	83

This chapter provides reference information about all public functions that you can access when using the CONTROL-M/EM API in your project. References to additional material are listed in the **See Also** section included under each function heading.

NOTE



The information in this chapter provides you with the necessary information to create requests in the form of XML strings.

Table 8 CONTROL-M/EM API classes

Class	Description
ComponentType	An abstract base class representing a CONTROL-M/EM component supported for use with the CONTROL-M/EM API. For more information, see “ComponentType class” on page 68 .
EMBasicXMLInvoker	The EMBasicXMLInvoker class is inherited from the EMXMLInvoker class. For more information, see “EMBasicXMLInvoker class” on page 69 .
EMXMLInvoker	The primary class of the API. For more information, see “EMXMLInvoker class” on page 72 .
GASComponent	The GASComponent class represents the Global Alerts Server (GAS). For more information, see “GASComponent class” on page 78 .
GSRComponent	The GSRComponent class represents the CONTROL-M/EM GUI Server. For more information, see “GSRComponent class” on page 80 .
InvokeException	This class is used to process error feedback from CONTROL-M/EM. For more information, see “InvokeException class” on page 81 .

ComponentType class

An abstract base class representing a CONTROL-M/EM component supported for use with the CONTROL-M/EM API.



NOTE

The Global Alerts Server (GAS) and the GUI Server are supported for use with the CONTROL-M/EM API.

When you send a request (for example, to change the status of an Alert) the CONTROL-M/EM component to which this request is referred is not indicated in the request. Instead, the request is referred by the ComponentType instance to the appropriate component.

The GSRComponent class and the GASComponent class are derived from the ComponentType class. These classes are used when creating an instance of the EMXMLInvoker or the EMBasicXMLInvoker.

For more information, see [“GASComponent class” on page 78](#) and [“GSRComponent class” on page 80](#).

EMBasicXMLInvoker class

The EMBasicXMLInvoker class is inherited from the EMXMLInvoker class. It shares many of the methods of that class.

The EMBasicXMLInvoker class uses the methods listed in [Table 9](#).

Table 9 EMBasicXMLInvoker methods(part 1 of 2)

Method	Description
BuildPasswordString	An EMXMLInvoker method that encodes a given text string for use in a Registration request. For more information, see “BuildPasswordString” on page 73 .
done	An EMXMLInvoker static method that stops the CONTROL-M/EM API services by breaking the connection with the CORBA processes. The done implementation is the same as it is under EMXMLInvoker . For more information, see “done” on page 73 .
encodePassword	Use the BuildPasswordString method instead of the encodePassword method to encode a given text string for use in a Registration request. The encodePassword method is no longer supported in CONTROL-M/EM API version 6.4.01. For more information, see “BuildPasswordString” on page 73 .
getProperties	An EMXMLInvoker static method used to obtain the CONTROL-M/EM Global Alerts Server or GUI Server host names from the ctmem.properties file. For more information, see “getProperties” on page 74 .
init	An EMXMLInvoker static method that starts the CONTROL-M/EM API services and initializes CORBA with default values or with values specified with its optional parameters. The init implementation is the same as it is under EMXMLInvoker . For more information, see “init” on page 75 .
invoke	Used to send a request to CONTROL-M/EM. the request is sent as an XML text string. When the invoke method is used with the EMBasicXMLInvoker class, polling for responses from CONTROL-M/EM is automatic. For more information, see “invoke” on page 70 .
setPollRequestIntervalMilli	Determines the interval, in milliseconds, between automatic poll requests. For more information, see “setPollRequestIntervalMilli” on page 71 .

Table 9 EMBasicXMLInvoker methods(part 2 of 2)

Method	Description
setPollRequestTimeoutMilli	Determines the total time, in milliseconds, allotted for polling following a request. For more information, see “ setPollRequestTimeoutMilli ” on page 71.
setProperties	An EMXMLInvoker static method used to specify the CONTROL-M/EM Global Alerts Server or GUI Server host names and a location from which to obtain them. For more information, see “ setProperties ” on page 77.

invoke

Used to send a request to CONTROL-M/EM. the request is sent as an XML text string. When the invoke method is used with the **EMBasicXMLInvoker** class, polling for responses from CONTROL-M/EM is automatic.

Syntax

```
public String invoke(String xmlRequest) throws InvokeException
```

Parameters

The **xmlRequest** string is a request that the user sends to CONTROL-M/EM. The string is a text file in an XML format that the CONTROL-M/EM API can accept and interpret.

Return codes

Response in XML format. Response data that addresses the request that was sent. It is returned as an XML formatted string.

See also

- “[invoke](#)” on page 77
- Chapter 7, “Request reference”

setPollRequestIntervalMilli

Sets the interval between automatically-sent poll requests. This time is measured from the **end** of the current poll request. This value must be less than or equal to the value specified with the **setPollRequestTimeoutMilli** method.

Syntax

```
public void setPollRequestIntervalMilli (final long timeout)
```

Parameters

ti meout. Time, in milliseconds.

Return codes

None.

See also

- “Polling interval timeout configuration” on page 221
- “setPollRequestTimeoutMilli” on page 71

setPollRequestTimeoutMilli

Total time that is allotted for polling. This value must be greater than or equal to the value specified with the **setPollRequestIntervalMilli** method.

Syntax

```
public void setPollRequestTimeoutMilli (final long timeout)
```

Parameters

ti meout. Time, in milliseconds.

Return codes

None.

See also

- “Polling interval timeout configuration” on page 221
- “setPollRequestIntervalMilli” on page 71

EMXMLInvoker class

The EMXMLInvoker class is the primary class of the API. Its methods are used to initiate and stop the API, get and set API properties, and send requests to CONTROL-M/EM.

The EMXMLInvoker class uses the methods listed in [Table 10](#).

Table 10 EMXMLInvoker methods

Method	Description
BuildPasswordString	An EMXMLInvoker method that encodes a given text string for use in a Registration request. For more information, see “BuildPasswordString” on page 73 .
done	An EMXMLInvoker static method that stops the CONTROL-M/EM API services by breaking the connection with the CORBA processes. For more information, see “done” on page 73 .
encodePassword	Use the BuildPasswordString method instead of the encodePassword method to encode a given text string for use in a Registration request. The encodePassword method is no longer supported in CONTROL-M/EM API version 6.4.01. For more information, see “BuildPasswordString” on page 73 .
getProperties	An EMXMLInvoker static method used to obtain the CONTROL-M/EM Global Alerts Server or GUI Server host names from the ctmem.properties file. For more information, see “getProperties” on page 74 .
init	An EMXMLInvoker static method that starts the CONTROL-M/EM API services and initializes CORBA with default values or with values specified with its optional parameters. For more information, see “init” on page 75 .
invoke	Used to send a request to CONTROL-M/EM. the request is sent as an XML text string. When the invoke method is used with the EMXMLInvoker class, the user must activate polling to receive a response from CONTROL-M/EM. For more information, see “invoke” on page 77 .
setProperties	An EMXMLInvoker static method used to specify the CONTROL-M/EM Global Alerts Server or GUI Server host names and a location from which to obtain them. For more information, see “setProperties” on page 77 .

BuildPasswordString

This **EMXMLInvoker** method prepares a text string for use as a user password in the User Registration request.

The **BuildPasswordString()** function must be called immediately before sending every API “Register” request because a different string is used for each request in some modes.

Syntax

```
public String BuildPasswordString(password_string) throws
    InvokeException
```

Parameters

password_string. Text string, subject to all limitations that apply to a CONTROL-M/EM password.

Return codes

The prepared string to be used in the password field of the registration request.

See also

- [“User Registration” on page 90.](#)
- For information about acceptable passwords, see the *CONTROL-M Administrator Guide*.

done

An **EMXMLInvoker** static method that stops the CONTROL-M/EM API services by breaking the connection with the CORBA processes.

Syntax

```
public static void done()
```

Parameters

None.

Return codes

None.

See also

- “Initializing and stopping the CONTROL-M/EM API services” on page 61
- “init” on page 75

getProperties

Used to obtain the CONTROL-M/EM Global Alerts Server or GUI Server host names from the **ctmem.properties** file.

Syntax

```
public Properties getProperties();
```

Parameters

Parameter	Description
<i>host_of_the_GUI_Server</i>	Host name of the CONTROL-M/EM GUI Server. Default: com.bmc.ctmem.emapi.GSR.hostname
<i>host_of_the_Alerts_Server</i>	Host name of the Global Alerts Server. Default: com.bmc.ctmem.emapi.GAS.hostname
<i>path_to_XML_data_files</i>	Location of the XML schema. Default: com.bmc.ctmem.emapi.XMLDATAPATH

Return codes

None.

See Also

- “setProperties” on page 77
- “Getting and setting CONTROL-M/EM API properties” on page 219

init

An **EMXMLInvoker** static method that starts the CONTROL-M/EM API services and initializes CORBA with default values or with values specified with its optional parameters. There are three prototypes.

The `init` method must be run at program startup.

Prototype 1

This is the default implementation of `init`.

Syntax

```
public static void init()
```

Parameters

None.

Return codes

None.

See also

- “done” on page 73
- “Initializing and stopping the CONTROL-M/EM API services” on page 61

Prototype 2

This `init` prototype enables you to include an array of strings representing a list of arguments.

Syntax

```
public static void init(String[] args)
```

Parameters

For a list of CORBA parameters suitable for use in `args`, see the manufacturer’s documentation.

Return codes

None.

See also

- “done” on page 73
- “Initializing and stopping the CONTROL-M/EM API services” on page 61
- “Modifying initialization properties” on page 215

Prototype 3

This `init` prototype enables you to include an array of strings representing a list of arguments.

Syntax

```
public static void init(String[] args, Properties props)
```

Parameters

For a list of CORBA parameters suitable for use in `args`, see the manufacturer’s documentation.

The `props` parameter (`Properties`) can contain CORBA parameters, using the same options as in the command line that was passed as the first parameter (`args`).

NOTE

The `Properties` class is part of the `java.util` package.



Return codes

None.

See also

- “done” on page 73
- “Initializing and stopping the CONTROL-M/EM API services” on page 61
- “Modifying initialization properties” on page 215

invoke

Used to send a request to CONTROL-M/EM. the request is sent as an XML text string. When the invoke method is used with the **EMXMLInvoker** class, the user must activate polling to receive a response from CONTROL-M/EM.

Syntax

```
public String invoke(String xmlRequest) throws InvokeException
```

Parameters

xml request string. This string is a request that the user sends to CONTROL-M/EM. The string is a text file in an XML format that the CONTROL-M/EM API can accept and interpret.

Return codes

Response in XML format. Response data that addresses the request that was sent. It is returned as an XML formatted string.

See also

- “invoke” on page 70
- Chapter 7, “Request reference”

setProperty

Used to specify the CONTROL-M/EM Global Alerts Server or GUI Server host names and the source from which to obtain them.

Syntax

```
public void setProperty(Properties props);
```

Parameters

Parameter	Description
<i>host_of_the_GUI_Server</i>	Set to the hostname of the CONTROL-M/EM GUI Server. Default: com.bmc.ctmem.emapi.GSR.hostname
<i>host_of_the_Alerts_Server</i>	Set to the hostname of the Global Alerts Server. Default: com.bmc.ctmem.emapi.GAS.hostname
<i>path_to_XML_data_files</i>	Location of the XML schema. Default: com.bmc.ctmem.emapi.XMLDATAPATH

Return codes

None.

See also

- “getProperties” on page 74
- “Getting and setting CONTROL-M/EM API properties” on page 219

GASComponent class

Represents the Global Alerts Server (GAS). When creating an instance of **EMXMLInvoker** (or **EMBasicXMLInvoker**) with a reference to a **GASComponent** type, the request is sent to the specified CONTROL-M/EM Global Alerts Server.

The **GASComponent** class uses the constructors listed in [Table 11](#).

Table 11 **GASComponent constructors**

Constructor	Description
GASComponent (Prototype 1)	Using this constructor, the request is sent to the Global Alerts Server hostname specified in the ctmemapi.properties file. For more information, see “ GASComponent (Prototype 1) ” on page 79.
GASComponent (Prototype 2)	Using this constructor, the request is sent to the Global Alerts Server hostname specified with the hostname parameter. For more information, see “ GASComponent (Prototype 2) ” on page 79.

GASComponent (Prototype 1)

Using this constructor, the request is sent to the Global Alerts Server hostname specified under **com.bmc.ctmem.emapi.GAS.hostname=***host_name* in the **ctmemapi.properties** file.

Syntax

```
GASComponent();
```

Parameters

None.

Return codes

None.

See also

- “ComponentType class” on page 68
- “GASComponent (Prototype 2)” on page 79
- “GSRComponent class” on page 80

GASComponent (Prototype 2)

Using this constructor, the request is sent to the Global Alerts Server hostname specified with the **hostname** parameter.

Syntax

```
GASComponent(String hostname);
```

Parameters

hostname. This is the Global Alerts Server hostname.

Return codes

None.

See also

- “ComponentType class” on page 68
- “GASComponent (Prototype 1)” on page 79
- “GSRComponent class” on page 80

GSRComponent class

Represents the GUI Server. When creating an instance of **EMXMLInvoker** (or **EMBasicXMLInvoker**) with a reference to a `GSRComponent` type, the request is sent to the specified CONTROL-M/EM GUI Server.

The GASComponent class uses the constructors listed in [Table 12](#).

Table 12 GASComponent constructors

Constructor	Description
GSRComponent (Prototype 1)	Using this constructor, the request is sent to the GUI Server hostname specified in the <code>ctmemapi.properties</code> file. For more information, see “ GSRComponent (Prototype 1) ” on page 80.
GSRComponent (Prototype 2)	Using this constructor, the request is sent to the GUI Server hostname specified with the <code>hostname</code> parameter. For more information, see “ GSRComponent (Prototype 2) ” on page 81.

GSRComponent (Prototype 1)

Using this constructor, the request is sent to the GUI Server hostname specified under `com.bmc.ctmem.emapi.GSR.hostname=host_name` in the `ctmemapi.properties` file.

Syntax

```
GSRComponent();
```

Parameters

None.

Return codes

None.

See also

- “ComponentType class” on page 68
- “GASComponent class” on page 78
- “GSRComponent (Prototype 2)” on page 81

GSRComponent (Prototype 2)

Using this constructor, the request is sent to the GUI Server hostname specified with the `hostname` parameter.

Syntax

```
GSRComponent(String hostname);
```

Parameters

`hostname`. This is the GUI Server hostname.

Return codes

None.

See also

- “ComponentType class” on page 68
- “GSRComponent (Prototype 1)” on page 80
- “GASComponent class” on page 78

InvokeException class

The **InvokeException** class enables the CONTROL-M/EM API user to obtain error information when an exception is thrown.

The **InvokeException** class includes the methods listed in [Table 13](#).

Table 13 InvokeException class methods

Method	Description
getMajorCode	Used to obtain the Major Code that identifies the error family to which an error belongs. For more information, see “getMajorCode” on page 82 .
getMinorCode	Used to obtain the Minor Code of an error. The Minor Code provides a unique identifier for the error in the family to which it belongs. For more information, see “getMinorCode” on page 82 .
getReason	Used to obtain the text description of an error. For more information, see “getReason” on page 83 .

getMajorCode

Used to obtain the Major Code that identifies the error family for an error.

Syntax

```
public int getMajorCode()
```

Parameters

None

Return codes

`int`. An integer that identifies the error family to which the error belongs.

See also

- [“getMinorCode” on page 82](#)
- [“getReason” on page 83](#)
- Appendix B, “Error codes and exceptions”

getMinorCode

Used to obtain the Minor Code of an error. The Minor Code provides a unique identifier for the error in the family to which it belongs.

Syntax

```
public int getMinorCode()
```

Parameters

None.

Return codes

`int`. An integer that provides a unique identifier for the error in its family.

See also

- “getMajorCode” on page 82
- “getReason” on page 83
- Appendix B, “Error codes and exceptions”

getReason

Used to obtain the text description of an error.

Syntax

```
public String getReason()
```

Parameters

None.

Return codes

`String`. This string is a text description of the error.

See also

- “getMajorCode” on page 82
- “getMinorCode” on page 82
- Appendix B, “Error codes and exceptions”

Request reference

This chapter serves as a reference for all CONTROL-M/EM API requests and presents the following topics:

Introduction to CONTROL-M/EM API requests	87
User Registration	90
Request parameters	90
Response parameters	91
Fault response	91
Errors	92
Examples	92
Check user token validity	94
Request parameters	94
Response parameters	94
Fault response parameters	95
Errors	95
Examples	95
Client Keep Alive	96
Request parameters	97
Response parameters	97
Fault response parameters	97
Errors	97
Examples	98
User Unregistration	99
Request parameters	99
Response parameters	100
Fault response parameters	100
Errors	100
Examples	100
Create job definitions	102
Request parameters	102
Response parameters	103
Fault response parameters	103
Errors	103
Examples	104
Create scheduling group definitions	106
Request parameters	107

Response parameters	107
Fault response parameters	108
Errors	108
Examples	109
Delete job definitions	112
Request parameters	113
Response parameters	115
Fault response parameters	116
Errors	116
Examples	117
Upload scheduling table	119
Request parameters	119
Response parameters	120
Polling request parameters	120
Polling response parameters	121
Fault response parameters	121
Errors	121
Examples	122
Order or Force	124
Request parameters	125
Response parameters	127
Polling request parameters	128
Polling response parameters	128
Fault response parameters	130
Errors	130
Examples	130
Job creation	133
Request parameters	134
Response parameters	153
Polling request parameters	154
Polling response parameters	154
Fault response parameters	155
Errors	156
Examples	156
Add condition	159
Request parameters	159
Response parameters	160
Polling request parameters	160
Polling response parameters	160
Fault response parameters	161
Errors	161
Examples	162
Delete condition	164
Request parameters	165
Response parameters	165
Polling request parameters	166
Polling response parameters	166
Fault response parameters	167
Errors	167

Examples.....	167
Job actions in active environment	170
Hold.....	171
Free	172
Confirm.....	174
Rerun.....	176
Kill	178
Force OK.....	179
Errors.....	181
Examples.....	182
Job tracking.....	184
Request parameters.....	184
Response parameters	185
Fault response parameters.....	186
Errors.....	187
Examples.....	188
Retrieve jobs in active environment	194
Request parameters.....	194
Response parameters	198
Fault response parameters.....	201
Errors.....	201
Examples.....	202
Change alert status.....	205
Request parameters.....	205
Response parameters	206
Fault response parameters.....	206
Errors.....	206
Examples.....	207
Retrieve BIM Services list	209
Request parameters.....	209
Response parameters	210
Fault response parameters.....	211
Examples.....	211
Fault Response	213
Fault Example	214

Introduction to CONTROL-M/EM API requests

The requests described in this chapter are listed in the following table:

Table 14 Requests listed in this chapter(part 1 of 2)

Request type	Description
User Registration	Sends the username and password of the user to the target server component.
Check user token validity	Checks if the specified user identification is still valid.

Table 14 Requests listed in this chapter(part 2 of 2)

Request type	Description
Client Keep Alive	Resets the registration timeout counter to zero.
User Unregistration	Sends the user token to the server component, which erases the user from its active users list.
Create job definitions	Creates or updates a regular scheduling table definition.
Create scheduling group definitions	Creates or updates a scheduling group definition.
Delete job definitions	Deletes a regular scheduling table definition.
Upload scheduling table	Uploads a scheduling table.
Order or Force	Inserts a job or Group Scheduling table into the Active Jobs file immediately (force) or subject to scheduling criteria (order).
Job creation	Creates a job processing or Group Scheduling table definition and inserts it into the Active Jobs file.
Add condition	Adds conditions.
Delete condition	Deletes conditions.
Job actions in active environment	Performs actions on jobs that are currently in the active environment.
Job tracking	Polls the Response repository in the CONTROL-M/EM GUI Server to receive completion confirmation from earlier job processing requests.
Retrieve jobs in active environment	Retrieves jobs that are currently in the active environment.
Change alert status	Changes the status of an alert (for example, from not_noticed to handled).
Retrieve BIM Services list	Retrieves the list of services active in the BMC Batch Impact Manager server.

Considerations

NOTE



CONTROL-M/EM API does not support non-English characters in request and response parameters.

Many XML elements in this chapter are CONTROL-M job parameters.

To create a successful request, particularly a Job Creation request or an Order or Force Job request, BMC Software recommends that you become familiar with CONTROL-M job parameters.



TIP

For details about CONTROL-M job parameters, see the *CONTROL-M Parameters Guide*.

The element names in this chapter often differ from the names of the job parameters to which they correspond. Search for parameters by name in the index of the *CONTROL-M Parameters Guide*.

For example, to find information about the **rerun_interval** element (in the Job Creation request), search the *CONTROL-M Parameters Guide* index for **rerun_interval**. You are directed to the Interval parameter.

SOAP Envelope for CONTROL-M/EM requests and responses

SOAP ("Simple Object Access Protocol") is a simple XML-based protocol that allows applications to exchange information. For more information, refer to <http://www.w3.org>.

CONTROL-M/Enterprise Manager API uses SOAP envelopes to wrap the CONTROL-M/EM requests and responses. Additionally, FAULT responses are wrapped within a SOAPFAULT element.

Request

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    . . . SPECIFIED_REQUEST . . .
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    . . . SPECIFIED_REQUEST_RESPONSE . . .
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

The fault responses are wrapped within a SOAP Fault envelope.

Fault response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body>
<SOAP-ENV:Fault>
<faultcode> ... </faultcode>
<faultstring>.... </faultstring>
<detail>
. . . . . FAULT_RESPONSE_OF_THE_REQUEST . . . . .
</detail>
</SOAP-ENV:Fault>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

NOTE

The fault parameters for all requests are described in “Fault Response” on page 213



User Registration

Sends the username and password of the user to the target server component. The server component returns a unique token, which must accompany all subsequent requests made during the session.

Request parameters

Figure 7 request_register XML parameters

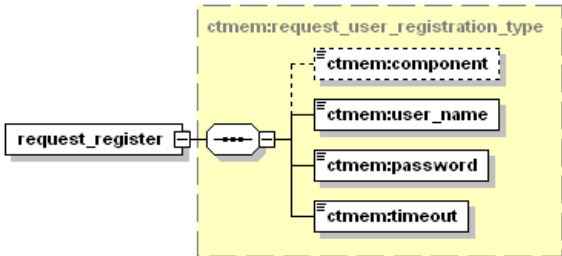


Table 15 request_register XML parameters description

Parameter	Description
component	This parameter is used <i>only</i> by CONTROL-M Business Process Integration. For more information, refer to the relevant product documentation.
user_name	CONTROL-M/EM username of the person making the request. String.

Table 15 request_register XML parameters description

Parameter	Description
password	CONTROL-M/EM user password of the person making the request. Note: This password must be sent as an encrypted string. Therefore, you must use the BuildPasswordString method to encrypt the password prior to making the User Registration request. For more information see, “BuildPasswordString” on page 73 .
timeout	Indicates a length of time, in seconds, until the user’s current user token is automatically invalidated. Integer. Default: 720. Optional. You should synchronize the value of this parameter with that of the EM_REFRESH_INTERVAL environment variable in CONTROL-M/Enterprise Manager. For more information, refer to “Preparing your project environment” on page 43 . Note: Use the Timeout Reset Request to restart the count from 0.

Response parameters

Figure 8 response_register XML parameters

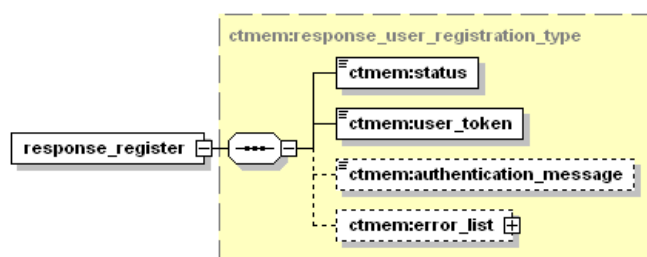


Table 16 response_register XML parameters description

Parameter	Description
status	Description of message content (for example, Error). String.
user_token	Unique ID that identifies the user. String.
authentication_message	Free text message containing authentication information. String

Fault response

XML parameters for fault_register, as well as a sample fault response are described in [“Fault Response” on page 213](#).

Errors

See “Authorization request errors (Major code 407)” on page 260.

Examples

Successful example

Request

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_register xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_name>emuser</ctmem:user_name>
      <ctmem:password>empass</ctmem:password>
      <ctmem:timeout>40</ctmem:timeout>
    </ctmem:request_register>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

```
<?xml version="1.0" encoding="iso-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:response_register xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:status>OK</ctmem:status>
      <ctmem:user_token>12345630</ctmem:user_token>
      <ctmem:authentication_message/>
    </ctmem:response_register>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Failure example

Request

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_register xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_name>emuser</ctmem:user_name>
      <ctmem:password>failed</ctmem:password>
      <ctmem:timeout>40</ctmem:timeout>
    </ctmem:request_register>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Server</faultcode>
      <faultstring>Error response from EM Server.</faultstring>
      <detail>
        <ctmem:fault_register xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
          <ctmem:error_list ctmem:highest_severity="Error">
            <ctmem:error ctmem:major="407" ctmem:minor="3" ctmem:severity="Error">
              <ctmem:error_message>Register failed.</ctmem:error_message>
            </ctmem:error>
          </ctmem:error_list>
        </ctmem:fault_register>
      </detail>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Check user token validity

Checks if the specified user identification is still valid.

Request parameters

Figure 9 request_check_user_token XML parameters

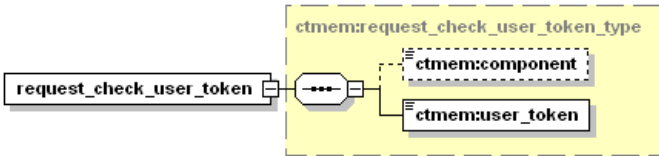


Table 17 request_check_user_token XML parameters Description

Parameter	Description
component	This parameter is used <i>only</i> by CONTROL-M Business Process Integration. For more information, refer to the relevant product documentation.
user_token	Serial identification number supplied to the user during registration. String.

Response parameters

Figure 10 response_check_user_token XML parameters

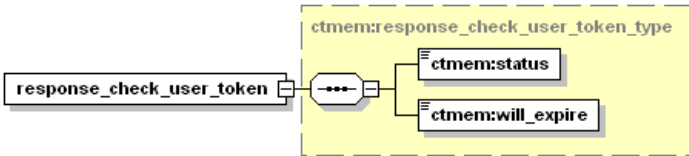


Table 18 response_check_user_token XML parameters description

Parameter	Description
status	Description of status token validity. String. Valid values: <ul style="list-style-type: none">■ VALID■ INVALID
will_expire	Amount of time, in seconds, remaining until the expiration of the user identification token. If the status is INVALID this parameter is set to zero.

Fault response parameters

XML parameters for `fault_check_user_token`, as well as a sample fault response are described in “[Fault Response](#)” on page 213.

Errors

See “[Authorization request errors \(Major code 407\)](#)” on page 260.

Examples

Successful example

Request

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_check_user_token xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>12345630</ctmem:user_token>
    </ctmem:request_check_user_token>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

```
<?xml version="1.0" encoding="iso-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:response_check_user_token xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:status>OK</ctmem:status>
      <ctmem:will_expire>587</ctmem:will_expire>
    </ctmem:response_check_user_token>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Failure example

Request

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_check_user_token xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>12345630</ctmem:user_token>
    </ctmem:request_check_user_token>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

```
<?xml version="1.0" encoding="iso-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Server</faultcode>
      <faultstring>Error response from EM Server.</faultstring>
      <detail>
        <ctmem:fault_check_user_token xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
          <ctmem:error_list ctmem:highest_severity="Error">
            <ctmem:error ctmem:major="407" ctmem:minor="1" ctmem:severity="Error">
              <ctmem:error_message>Invalid user token.</ctmem:error_message>
            </ctmem:error>
          </ctmem:error_list>
        </ctmem:fault_check_user_token>
      </detail>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Client Keep Alive

When a CONTROL-M/EM API user registers with CONTROL-M/EM, the user receives a user token that is in effect for a limited period of time.

The Timeout Reset request resets the registration timeout counter to zero. Timeout Reset requests can be sent intermittently to keep a user's registration valid during lengthy sessions.

Request parameters

Figure 11 request_client_keep_alive XML parameters

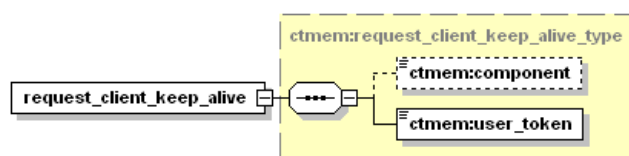


Table 19 request_client_keep_alive XML parameter description

Parameter	Description
component	This parameter is used <i>only</i> by CONTROL-M Business Process Integration. For more information, refer to the relevant product documentation.
user_token	Serial identification number supplied to the user during registration. The user indicated by this number is the user. String.

Response parameters

Table 20 response_client_keep_alive XML parameters description

Parameter	Description
status	Describes the condition of the element that contains it.(for example, Error). String.
user_token	Unique ID that identifies the user. String.

Fault response parameters

XML parameters for fault_client_keep_alive, as well as a sample fault response are described in “[Fault Response](#)” on page 213.

Errors

See “[Authorization request errors \(Major code 407\)](#)” on page 260.

Examples

Successful example

Request

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_client_keep_alive xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>12345630</ctmem:user_token>
    </ctmem:request_client_keep_alive>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:response_client_keep_alive xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:status>OK</ctmem:status>
    </ctmem:response_client_keep_alive>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Failure example

Request

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_client_keep_alive xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>12345630</ctmem:user_token>
    </ctmem:request_client_keep_alive>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

```
<?xml version="1.0" encoding="iso-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Server</faultcode>
      <faultstring>Error response from EM Server.</faultstring>
      <detail>
        <ctmem:fault_client_keep_alive xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
          <ctmem:error_list ctmem:highest_severity='Error' >
            <ctmem:error ctmem:major='407' ctmem:minor='1' ctmem:severity='Error' >
              <ctmem:error_message>Invalid user token.</ctmem:error_message>
            </ctmem:error>
          </ctmem:error_list>
        </ctmem:fault_client_keep_alive>
      </detail>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

User Unregistration

When the application finishes using the API, use the Unregister request to send the user token to the server component. The server component erases the user from its active users list. The user token is invalidated when the request is complete.

Request parameters

Figure 12 request_unregister XML parameters

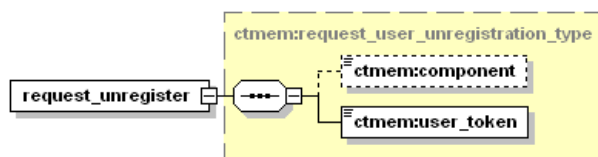


Table 21 request_unregister XML parameter description

Parameter	Description
component	This parameter is used <i>only</i> by CONTROL-M Business Process Integration. For more information, refer to the relevant product documentation.
user_token	Unique ID that identifies the user. String.

Response parameters

Figure 13 response_unregister XML parameter

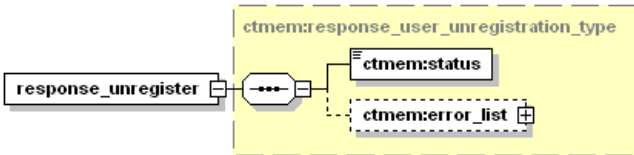


Table 22 response_unregister XML parameter description

Parameter	Description
status	Description of message content (for example, Error). String.

Fault response parameters

XML parameters for fault_unregister, as well as a sample fault response are described in “Fault Response” on page 213.

Errors

See “Authorization request errors (Major code 407)” on page 260.

Examples

Successful example

Request

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_unregister xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>12345630</ctmem:user_token>
    </ctmem:request_unregister>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:response_unregister xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:status>OK
    </ctmem:status>
    </ctmem:response_unregister>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Failure example

Request

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_unregister xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>12345630</ctmem:user_token>
    </ctmem:request_unregister>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

```
<?xml version="1.0" encoding="iso-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Server</faultcode>
      <faultstring>Error response from EM Server.</faultstring>
      <detail>
        <ctmem:fault_unregister xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
          <ctmem:error_list ctmem:highest_severity='Error'>
            <ctmem:error ctmem:major='407' ctmem:minor='4' ctmem:severity='Error'>
              <ctmem:error_message>Unregister failed.</ctmem:error_message>
            </ctmem:error>
          </ctmem:error_list>
        </ctmem:fault_unregister>
      </detail>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Create job definitions

Creates or updates a regular scheduling table definition. It is possible to create or update only one scheduling table per request, as follows:

- If the scheduling table does not exist, it is created.
- If the scheduling table exists, jobs that are defined in the request are added to the existing scheduling table.

Request parameters

Figure 14 request_def_create_jobs XML parameters

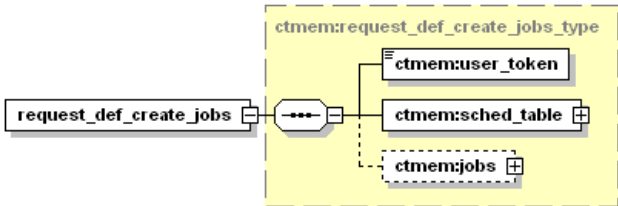


Table 23 request_def_create_jobs XML Parameters Description

Parameter	Description
user_token	Serial identification number supplied to the user during registration. String.
sched_table	Scheduling table wrapper. The scheduling table is identified by the elements listed in Table 24
jobs	A sequence of job . For a list of the parameters for job, refer to Appendix C, “Job and Scheduling Group XML parameters.”

Table 24 sched_table XML parameters description

Parameter	Description
control_m	Name of the CONTROL-M installation that processes the request. String.
table_name	Name of the scheduling table. String.
table_library	CONTROL-M for z/OS only: Name of the library in which the scheduling table is located. String.
user_daily	Name of the user daily. String.

Response parameters

Figure 15 response_def_create_jobs XML parameter

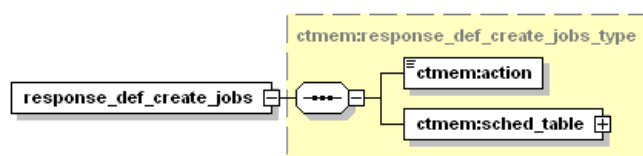


Table 25 response_def_create_jobs XML parameters description

Parameter	Description
action	Description of successful action. String. Valid values: <ul style="list-style-type: none"> ■ CREATED ■ ADDED
sched_table	Scheduling table wrapper. The scheduling table is identified by the elements listed in Table 26

Table 26 sched_table XML Parameters Description

Parameter	Description
control_m	Name of the CONTROL-M installation that processes the request. String.
table_name	Name of the scheduling table. String.
table_library	CONTROL-M for z/OS only: Name of the library in which the scheduling table is located. String.

Fault response parameters

XML parameters for fault_def_create_jobs, as well as a sample fault response are described in [“Fault Response” on page 213](#).

Errors

See [“Create job/scheduling group definitions request errors \(Major code 412\)” on page 262](#).

Examples

Successful example

Request

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body>
  <ctmem:request_def_create_jobs xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
    <ctmem:user_token>12345630</ctmem:user_token>
    <ctmem:sched_table>
      <ctmem:control_m>ctm630</ctmem:control_m>
      <ctmem:table_name>apitest</ctmem:table_name>
      <ctmem:user_daily>apitest</ctmem:user_daily>
    </ctmem:sched_table>
    <ctmem:jobs>
      <ctmem:job>
        <ctmem:job_name>apitest1</ctmem:job_name>
        <ctmem:mem_name>apiMemName1</ctmem:mem_name>
        <ctmem:task_type>dummy</ctmem:task_type>
        <ctmem:application>apiWinApp1</ctmem:application>
        <ctmem:group>apiGroup1</ctmem:group>
        <ctmem:owner>controlm</ctmem:owner>
        <ctmem:author>emuser</ctmem:author>
        <ctmem:on_do_statements>
          <ctmem:on_do_statement>
            <ctmem:on_statements>
              <ctmem:on_statement>
                <ctmem:code>ok</ctmem:code>
                <ctmem:statement>*</ctmem:statement>
              </ctmem:on_statement>
            </ctmem:on_statements>
          </ctmem:do_statements>
          <ctmem:do_cond>
            <ctmem:condition>cond1</ctmem:condition>
            <ctmem:date>ODAT</ctmem:date>
            <ctmem:sign>delete</ctmem:sign>
          </ctmem:do_cond>
        </ctmem:do_statements>
      </ctmem:on_do_statement>
    </ctmem:on_do_statements>
  </ctmem:job>
  <ctmem:job>
    <ctmem:job_name>apitest2</ctmem:job_name>
    <ctmem:mem_name>apiMemName2</ctmem:mem_name>
    <ctmem:mem_lib>d:\</ctmem:mem_lib>
    <ctmem:task_type>job</ctmem:task_type>
    <ctmem:application>apiWinApp2</ctmem:application>
    <ctmem:group>apiGroup2</ctmem:group>
    <ctmem:command>test.bat</ctmem:command>
    <ctmem:owner>controlm</ctmem:owner>
    <ctmem:author>emuser</ctmem:author>
  </ctmem:job>
</ctmem:jobs>
</ctmem:request_def_create_jobs>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```


Response

```
<?xml version="1.0" encoding="iso-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:response_def_create_jobs xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:action>CREATED</ctmem:action>
      <ctmem:sched_table>
        <ctmem:control_m>ctm630</ctmem:control_m>
        <ctmem:table_name>api test</ctmem:table_name>
      </ctmem:sched_table>
    </ctmem:response_def_create_jobs>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Failure example

Request

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_def_create_jobs xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>12345630</ctmem:user_token>
      <ctmem:sched_table>
        <ctmem:control_m>ctm630</ctmem:control_m>
        <ctmem:table_name>api test</ctmem:table_name>
        <ctmem:user_daily>api test</ctmem:user_daily>
      </ctmem:sched_table>
      <ctmem:jobs>
        <ctmem:job>
          <ctmem:job_name>api test2</ctmem:job_name>
          <ctmem:mem_name>api MemName2</ctmem:mem_name>
          <ctmem:mem_lib>d:</ctmem:mem_lib>
          <ctmem:task_type>command</ctmem:task_type>
          <ctmem:application>api WinApp2</ctmem:application>
          <ctmem:group>api Group2</ctmem:group>
        </ctmem:job>
      </ctmem:jobs>
    </ctmem:request_def_create_jobs>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

```
<?xml version="1.0" encoding="iso-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Server</faultcode>
      <faultstring>Error response from EM Server.</faultstring>
      <detail>
        <ctmem: fault_def_create_jobs xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
          <ctmem: error_list ctmem: highest_severity='Error' >
            <ctmem: error ctmem: major='412' ctmem: minor='1' ctmem: severity='Error' >
              <ctmem: error_message>Create jobs definitions failed, invalid params.
            </ctmem: error_message>
            </ctmem: error>
            <ctmem: error ctmem: major='412' ctmem: minor='14' ctmem: severity='Error' >
              <ctmem: error_message>Create jobs definitions validation error: Job[1]:
                Field: CmdLine Error: Field must have a value.
              </ctmem: error_message>
            </ctmem: error>
            <ctmem: error ctmem: major='412' ctmem: minor='14' ctmem: severity='Error' >
              <ctmem: error_message>Create jobs definitions validation error: Job[1]:
                Field: Owner Error: EM50011E: The field must have a value.
              </ctmem: error_message>
            </ctmem: error>
            <ctmem: error ctmem: major='412' ctmem: minor='14' ctmem: severity='Error' >
              <ctmem: error_message>Create jobs definitions validation error: Job[1]:
                Field: Author Error: EM50011E: The field must have a value.
              </ctmem: error_message>
            </ctmem: error>
          </ctmem: error_list>
        </ctmem: fault_def_create_jobs>
      </detail>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Create scheduling group definitions

Creates or updates a scheduling group definition. It is possible to create or update only one scheduling group per request, as follows:

- If the scheduling group does not exist, it is created.
- If the scheduling group exists, jobs that are defined in the request are added to the existing scheduling group.

The scheduling group entity, which is specified in the `sched_group` request, will not override the existing scheduling group entity definitions.

Request parameters

Figure 16 request_def_create_sched_group XML parameters

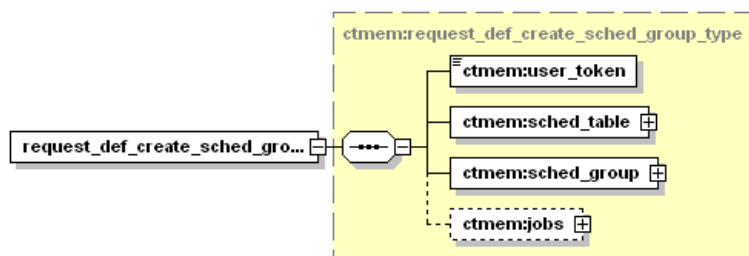


Table 27 request_def_create_sched_group XML Parameters Description

Parameter	Description
user_token	Serial identification number supplied to the user during registration. String.
sched_table	Scheduling table wrapper. The scheduling group table is identified by the elements listed in Table 28
sched_group	Scheduling group fields list wrapper. String. For a list of the parameters for <code>sched_group</code> , refer to Appendix C, “Job and Scheduling Group XML parameters.”
jobs	A sequence of job . For a list of the parameters for job, refer to Appendix C, “Job and Scheduling Group XML parameters.”

Table 28 sched_table XML parameters description

Parameter	Description
control_m	Name of the CONTROL-M installation that processes the request. String.
table_name	Name of the scheduling table. String.
table_library	CONTROL-M for z/OS only: Name of the library in which the scheduling table is located. String
user_daily	Name of the user daily. String.

Response parameters

Figure 17 response_def_create_sched_group XML parameters

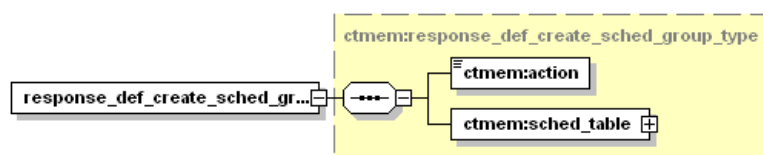


Table 29 response_def_create_sched_group XML parameters description

Parameter	Description
action	Description of successful action. String. Valid values: <ul style="list-style-type: none">■ CREATED■ ADDED
sched_table	Scheduling table wrapper. The scheduling group table is identified by the elements listed in Table 30 .

Table 30 sched_table XML Parameters Description

Parameter	Description
control_m	Name of the CONTROL-M installation that processes the request. String.
table_name	Name of the scheduling table. String.
table_library	CONTROL-M for z/OS only: Name of the library in which the scheduling table is located. String.

Fault response parameters

XML parameters for `fault_def_create_sched_group`, as well as a sample fault response are described in [“Fault Response” on page 213](#).

Errors

See [“Create job/scheduling group definitions request errors \(Major code 412\)” on page 262](#).

Examples

Successful example

Request

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body>
  <ctmem:request_def_create_sched_group xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
    <ctmem:user_token>12345630</ctmem:user_token>
    <ctmem:sched_table>
      <ctmem:control_m>ctm630</ctmem:control_m>
      <ctmem:table_name>api testSG</ctmem:table_name>
      <ctmem:table_library></ctmem:table_library>
      <ctmem:user_daily>system</ctmem:user_daily>
    </ctmem:sched_table>
    <ctmem:sched_group>
      <ctmem:job_name>api test</ctmem:job_name>
      <ctmem:mem_name>api MemName</ctmem:mem_name>
      <ctmem:application>api WinApp</ctmem:application>
      <ctmem:group>api testSG</ctmem:group>
      <ctmem:owner>control m</ctmem:owner>
      <ctmem:author>emuser</ctmem:author>
      <ctmem:sched_tags>
        <ctmem:tag>
          <ctmem:tag_name>tag1</ctmem:tag_name>
          <ctmem:month_days>ALL</ctmem:month_days>
          <ctmem:FEB>yes</ctmem:FEB>
        </ctmem:tag>
      </ctmem:sched_tags>
    </ctmem:sched_group>
    <ctmem:jobs>
      <ctmem:job>
        <ctmem:job_name>api test1</ctmem:job_name>
        <ctmem:mem_name>api MemName1</ctmem:mem_name>
        <ctmem:task_type>dummy</ctmem:task_type>
        <ctmem:application>api WinApp1</ctmem:application>
        <ctmem:group>api testSG</ctmem:group>
        <ctmem:owner>control m</ctmem:owner>
        <ctmem:author>emuser</ctmem:author>
        <ctmem:job_sched_tags>
          <ctmem:job_tag>
            <ctmem:tag_name>tag1</ctmem:tag_name>
          </ctmem:job_tag>
        </ctmem:job_sched_tags>
      </ctmem:job>
    </ctmem:jobs>
  </ctmem:request_def_create_sched_group>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

```
<?xml version="1.0" encoding="iso-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:response_def_create_sched_group xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:action>CREATED
    </ctmem:action>
    <ctmem:sched_table>
      <ctmem:control_m>ctm630
    </ctmem:control_m>
      <ctmem:table_name>api testSG
    </ctmem:table_name>
    </ctmem:sched_table>
    </ctmem:response_def_create_sched_group>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Failure example

Request

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body>
  <ctmem:request_def_create_sched_group xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
    <ctmem:user_token>12345630</ctmem:user_token>
    <ctmem:sched_table>
      <ctmem:control_m>ctm630</ctmem:control_m>
      <ctmem:table_name>api testSG1</ctmem:table_name>
      <ctmem:table_library></ctmem:table_library>
      <ctmem:user_daily>system</ctmem:user_daily>
    </ctmem:sched_table>
    <ctmem:sched_group>
      <ctmem:job_name>api test</ctmem:job_name>
      <ctmem:mem_name>api MemName</ctmem:mem_name>
      <ctmem:application>api WinApp</ctmem:application>
      <ctmem:group>api testSG1</ctmem:group>
      <ctmem:owner>control m</ctmem:owner>
      <ctmem:author>emuser</ctmem:author>
      <ctmem:sched_tags>
        <ctmem:tag>
          <ctmem:tag_name>tag1</ctmem:tag_name>
          <ctmem:month_days>ALL</ctmem:month_days>
          <ctmem:FEB>yes</ctmem:FEB>
        </ctmem:tag>
      </ctmem:sched_tags>
    </ctmem:sched_group>
    <ctmem:jobs>
      <ctmem:job>
        <ctmem:job_name>api test1</ctmem:job_name>
        <ctmem:mem_name>api MemName1</ctmem:mem_name>
        <ctmem:task_type>dummy</ctmem:task_type>
        <ctmem:application>api WinApp1</ctmem:application>
        <ctmem:group>api testSG2</ctmem:group>
        <ctmem:owner>control m</ctmem:owner>
        <ctmem:author>emuser</ctmem:author>
        <ctmem:job_sched_tags>
          <ctmem:job_tag>
            <ctmem:tag_name>tag1</ctmem:tag_name>
          </ctmem:job_tag>
        </ctmem:job_sched_tags>
      </ctmem:job>
      <ctmem:job>
        <ctmem:job_name>api test2</ctmem:job_name>
        <ctmem:mem_name>api MemName2</ctmem:mem_name>
        <ctmem:mem_lib>d:</ctmem:mem_lib>
        <ctmem:task_type>job</ctmem:task_type>
        <ctmem:application>api WinApp2</ctmem:application>
        <ctmem:group>api testSG3</ctmem:group>
        <ctmem:command>test.bat</ctmem:command>
        <ctmem:owner>control m</ctmem:owner>
        <ctmem:author>emuser</ctmem:author>
        <ctmem:job_sched_tags>
          <ctmem:job_tag>
            <ctmem:tag_name>tag1</ctmem:tag_name>
          </ctmem:job_tag>
        </ctmem:job_sched_tags>
      </ctmem:job>
    </ctmem:jobs>
  </ctmem:request_def_create_sched_group>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

```
<?xml version="1.0" encoding="iso-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Server
      </faultcode>
      <faultstring>Error response from EM Server.
      </faultstring>
      <detail>
        <ctmem: fault_def_create_sched_group xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
          <ctmem: error_list ctmem: highest_severity='Error' >
            <ctmem: error ctmem: major='412' ctmem: minor='1' ctmem: severity='Error' >
              <ctmem: error_message>Create jobs definitions failed, invalid params.
              </ctmem: error_message>
            </ctmem: error>
            <ctmem: error ctmem: major='412' ctmem: minor='14' ctmem: severity='Error' >
              <ctmem: error_message>Create jobs definitions validation error: Job[3]:
Group name of job entity differs from group name of scheduling group entity.
              </ctmem: error_message>
            </ctmem: error>
            <ctmem: error ctmem: major='412' ctmem: minor='14' ctmem: severity='Error' >
              <ctmem: error_message>Create jobs definitions validation error: Job[2]:
Group name of job entity differs from group name of scheduling group entity.
              </ctmem: error_message>
            </ctmem: error>
          </ctmem: error_list>
        </ctmem: fault_def_create_sched_group>
      </detail>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Delete job definitions

This request will delete one or more jobs from a scheduling table according to the user defined deletion criteria.

Scheduling group entities will not be deleted.

Scheduling table entities will not be deleted.

If no jobs are deleted according to the specified deletion criteria, a fault response with the appropriate message is returned.

Request parameters

Figure 18 request_def_delete_jobs XML parameters

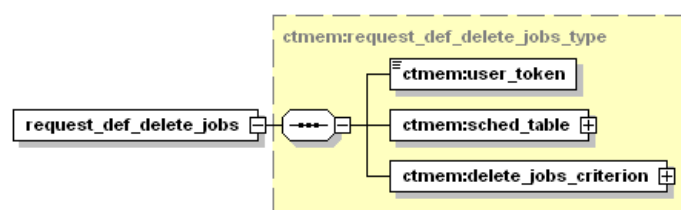


Table 31 request_def_delete_jobs XML parameters Description

Parameter	Description
user_token	Serial identification number supplied to the user during registration. String.
sched_table	Scheduling table wrapper. The scheduling group table is identified by the elements listed in Table 32 .
delete_jobs_criterion	Delete jobs criteria wrapper. String. Consists of include and exclude filters that allow specifying items to include with or exclude from the deletion criteria, as described in Table 33 .

Table 32 sched_table XML Parameters Description

Parameter	Description
control_m	Name of the CONTROL-M installation that processes the request. String.
table_name	Name of the scheduling table. String.
table_library	CONTROL-M for z/OS only: Name of the library in which the scheduling table is located. String.

Figure 19 delete_jobs_criterion XML parameters

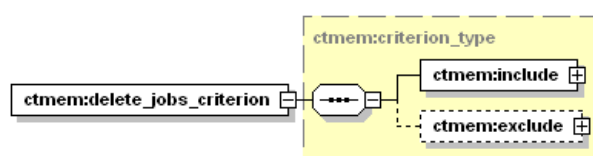


Table 33 delete_jobs_criterion XML parameters description

Parameter	Description
include	Mandatory. Filter that consists of a sequence of search_criterion elements. For more information, refer to Table 34 .
exclude	Optional. Filter that consists of a sequence of search_criterion elements. For more information, refer to Table 34 .

Figure 20 include or exclude XML parameter

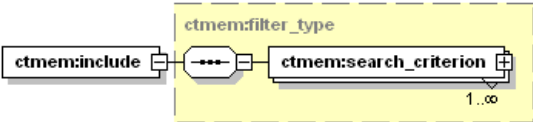


Table 34 include or exclude XML parameter description

Parameter	Description
search_criterion	Search criteria wrapper that consists of a sequence of param elements. At least one search_criterion element must appear under the exclude element. The amount of search_criterion elements is unbounded. The relationship between search_criterion elements in one filter is OR. For more information about param, refer to Table 35.

Table 35 search_criterion XML parameters description

Parameter	Description
param	Parameters used to build the search criteria. At least one param element should appear under a search_criterion element. The amount of param elements is unbounded. The relationship between param elements in the same search_criterion is AND. Elements of param are listed in Table 36. String.

Figure 21 param XML parameters

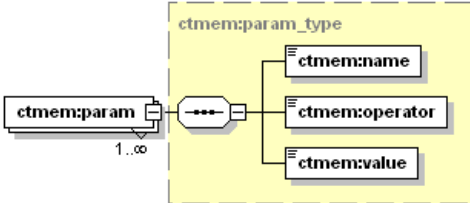


Table 36 param XML parameters description

Parameter	Description
name	Name of the scheduling table definition parameter used as a search criteria. String. Mandatory. For a list of valid values for the name parameter, refer to Table 37
operator	Operator used in search criteria. String. Valid values: <ul style="list-style-type: none">■ EQ■ NE■ LT■ GT■ LIKE
value	Value used in search criteria. Any valid value of a job parameter. Wildcards and search patterns can be used in combination with LIKE operator. String. Mandatory.

Table 37 Valid values for name

Parameter	Description
APPLICATION	Name of the application to which the job's group belongs.
GROUP_NAME	Name of the group to which the job belongs.
MEMNAME	Name of the file that contains the job script.
JOB_NAME	Name of the job.
DESCRIPTION	Description of the job.
AUTHOR	CONTROL-M/EM user who defined the job. Note: This argument is used by the CONTROL-M security mechanism and under certain circumstances, cannot be modified. For more information, see the Security chapter and the description of the AuthorSecurity system parameter in the <i>CONTROL-M Administrator Guide</i> .
NODE_ID	Node ID of the host on which the job was most recently run. Note: This parameter is not available for MVS jobs.
MEM_LIB	Name of the path that contains the job script file.

NOTE

For more information about the parameters described in [Table 37](#), refer to the CONTROL-M/Enterprise Manager documentation.

Response parameters

Figure 22 response_def_delete_jobs XML parameters

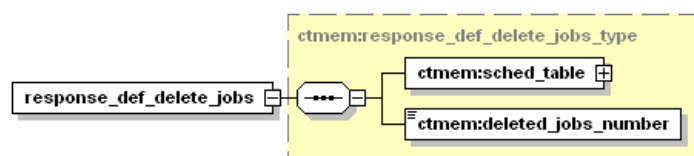


Table 38 response_def_delete_jobs XML parameters description

Parameter	Description
sched_table	Scheduling table wrapper. The scheduling group table is identified by the elements listed in Table 39 .
deleted_jobs_number	Number of jobs that were deleted. String.

Table 39 sched_table XML Parameters Description

Parameter	Description
control_m	Name of the CONTROL-M installation that processes the request. String.
table_name	Name of the scheduling table. String.
table_library	CONTROL-M for z/OS only: Name of the library in which the scheduling table is located. String.

Fault response parameters

XML parameters for `fault_def_delete_jobs`, as well as a sample fault response are described in [“Fault Response” on page 213](#).

Errors

See [“Delete job definitions request errors \(Major code 413\)” on page 262](#).

Examples

Successful example

Request

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body>
  <ctmem:request_def_delete_jobs xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
    <ctmem:user_token>12345630</ctmem:user_token>
    <ctmem:sched_table>
      <ctmem:control_m>ctm630</ctmem:control_m>
      <ctmem:table_name>apitestSG</ctmem:table_name>
    </ctmem:sched_table>
    <ctmem:delete_jobs_criterion>
      <ctmem:include>
        <ctmem:search_criterion>
          <ctmem:param>
            <ctmem:name>JOB_NAME</ctmem:name>
            <ctmem:operator>LIKE</ctmem:operator>
            <ctmem:value>*</ctmem:value>
          </ctmem:param>
        </ctmem:search_criterion>
      </ctmem:include>
      <ctmem:exclude>
        <ctmem:search_criterion>
          <ctmem:param>
            <ctmem:name>MEMNAME</ctmem:name>
            <ctmem:operator>EQ</ctmem:operator>
            <ctmem:value>apiMemName2</ctmem:value>
          </ctmem:param>
        </ctmem:search_criterion>
      </ctmem:exclude>
    </ctmem:delete_jobs_criterion>
  </ctmem:request_def_delete_jobs>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

```
<?xml version="1.0" encoding="iso-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:response_def_delete_jobs xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:sched_table>
        <ctmem:control_m>ctm630
        </ctmem:control_m>
        <ctmem:table_name>api testSG
        </ctmem:table_name>
      </ctmem:sched_table>
      <ctmem:deleted_jobs_number>1
      </ctmem:deleted_jobs_number>
    </ctmem:response_def_delete_jobs>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Failure example

Request

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body>
  <ctmem:request_def_delete_jobs xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
    <ctmem:user_token>12345630</ctmem:user_token>
    <ctmem:sched_table>
      <ctmem:control_m>ctm630</ctmem:control_m>
      <ctmem:table_name>apitestSG</ctmem:table_name>
    </ctmem:sched_table>
    <ctmem:delete_jobs_criterion>
      <ctmem:include>
        <ctmem:search_criterion>
          <ctmem:param>
            <ctmem:name>UNKNOWN</ctmem:name>
            <ctmem:operator>LIKE</ctmem:operator>
            <ctmem:value>*</ctmem:value>
          </ctmem:param>
        </ctmem:search_criterion>
      </ctmem:include>
    </ctmem:delete_jobs_criterion>
  </ctmem:request_def_delete_jobs>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

```
<?xml version="1.0" encoding="iso-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Server
      </faultcode>
      <faultstring>Error response from EM Server.
      </faultstring>
      <detail>
        <ctmem:fault_def_delete_jobs
          xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
          <ctmem:error_list ctmem:highest_severity='Error' >
            <ctmem:error ctmem:major='413' ctmem:minor='3' ctmem:severity='Error' >
              <ctmem:error_message>Failed to delete jobs from scheduling table.
              </ctmem:error_message>
            </ctmem:error>
            <ctmem:error ctmem:major='413' ctmem:minor='6' ctmem:severity='Error' >
              <ctmem:error_message>Delete jobs definitions validation error:
                [UNKNOWN] is not valid filter field.
              </ctmem:error_message>
            </ctmem:error>
          </ctmem:error_list>
        </ctmem:fault_def_delete_jobs>
      </detail>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Upload scheduling table

Uploads a scheduling table. Only one scheduling table can be uploaded per request. It is possible to force upload a table using the optional force parameter.

Request parameters

Figure 23 request_def_upload_table XML parameters

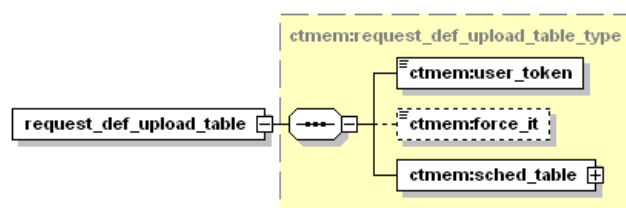


Table 40 request_def_upload_table XML parameters Description

Parameter	Description
user_token	Serial identification number supplied to the user during registration. String.
force_it	Forces the uploading of a scheduling table. String. Optional.
sched_table	Scheduling table wrapper. The scheduling group table is identified by the elements listed in Table 41

Table 41 sched_table XML Parameters Description

Parameter	Description
control_m	Name of the CONTROL-M installation that processes the request. String.
table_name	Name of the scheduling table. String.
table_library	CONTROL-M for z/OS only: Name of the library in which the scheduling table is located. String

Response parameters

Table 42 response_def_upload_table XML Parameters Description

Parameter	Description
response_token	Used in a polling request.

Polling request parameters

Figure 24 request_poll XML parameters

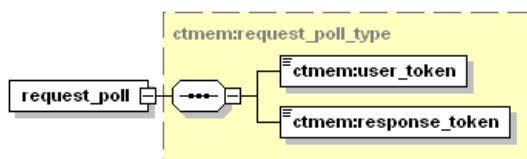


Table 43 request_poll_def_upload_table XML parameters Description

Parameter	Description
user_token	Serial identification number supplied to the user during registration. String.
response_token	Used in a polling request. This token is received in the immediate response of response_def_upload_table.

Polling response parameters

Figure 25 response_poll_def_upload_table XML parameters

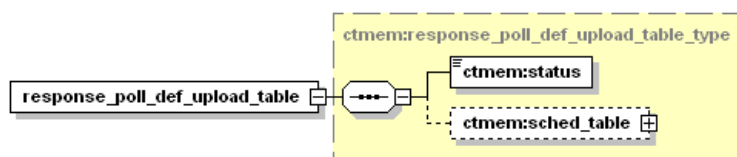


Table 44 response_poll_def_upload_table XML parameters description

Parameter	Description
status	Status of polling. String. Valid values: <ul style="list-style-type: none"> ■ OK ■ EXEC

Fault response parameters

XML parameters for fault_def_upload_table and fault_poll_def_upload_table, as well as a sample fault response are described in [“Fault Response” on page 213](#).

Errors

See [“Upload scheduling table request errors \(Major code 411\)” on page 261](#).

Examples

Successful example

Request

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_def_upload_table xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>12345630</ctmem:user_token>
      <ctmem:force_it>no</ctmem:force_it>
      <ctmem:sched_table>
        <ctmem:control_m>ctm630</ctmem:control_m>
        <ctmem:table_name>apitest</ctmem:table_name>
      </ctmem:sched_table>
    </ctmem:request_def_upload_table>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

```
<?xml version="1.0" encoding="iso-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:response_def_upload_table xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:status>OK</ctmem:status>
      <ctmem:response_token>91</ctmem:response_token>
    </ctmem:response_def_upload_table>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Polling request

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_poll_def_upload_table xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>12345630</ctmem:user_token>
      <ctmem:response_token>91</ctmem:response_token>
    </ctmem:request_poll_def_upload_table>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Polling response

```
<?xml version="1.0" encoding="iso-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:response_poll_def_upload_table xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:status>OK</ctmem:status>
    </ctmem:response_poll_def_upload_table>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Failure example

Request

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_def_upload_table xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>12345630</ctmem:user_token>
      <ctmem:force_it>no</ctmem:force_it>
      <ctmem:sched_table>
        <ctmem:control_m>ctm630</ctmem:control_m>
        <ctmem:table_name>inexistent</ctmem:table_name>
      </ctmem:sched_table>
    </ctmem:request_def_upload_table>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

```
<?xml version="1.0" encoding="iso-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Server
    </faultcode>
    <faultstring>Error response from EM Server.
    </faultstring>
    <detail>
      <ctmem:fault_def_upload_table xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
        <ctmem:error_list ctmem:highest_severity="Error">
          <ctmem:error ctmem:major="411" ctmem:minor="3" ctmem:severity="Error">
            <ctmem:error_message>Cannot get Scheduling Table from database.
            </ctmem:error_message>
          </ctmem:error>
        </ctmem:error_list>
      </ctmem:fault_def_upload_table>
    </detail>
  </SOAP-ENV:Fault>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Order or Force

Does the following operations with jobs and Group Scheduling tables:

- Order a job. The job is entered into the Active Jobs file only when its scheduling criteria are met.
- Force a job. The Job is entered into the Active Jobs file whether or not its scheduling criteria are met.
- Order or force a Group Scheduling table. All jobs in the table are ordered or forced.

The mandatory parameters for ordering a job differ from the mandatory parameters that are specified when ordering or forcing a Group Scheduling table. Optional parameters can be supplied for both jobs and Group Scheduling tables.

Request parameters

Figure 26 request_order_force XML parameters

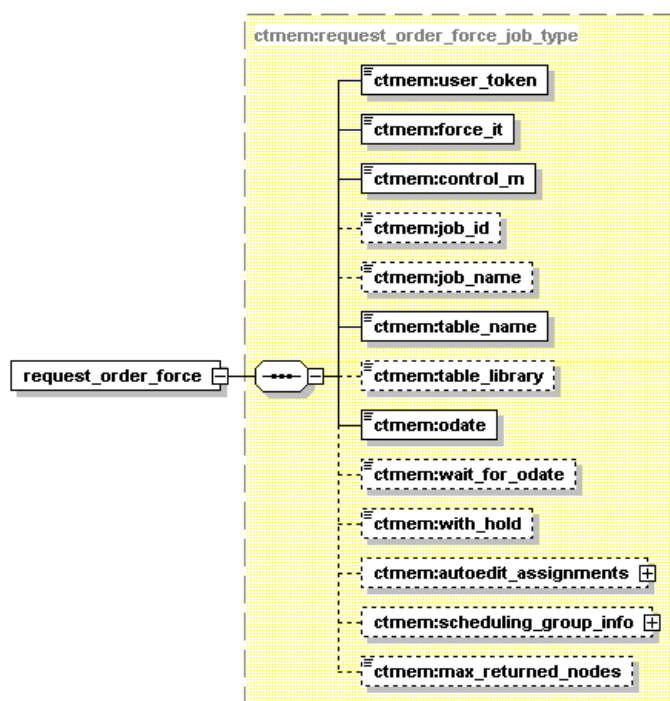


Table 45 request_order_force XML parameters description(part 1 of 2)

Parameter	Description
user_token	Serial identification number supplied to the user during registration. String.
control_m	Data center name. Mandatory for both jobs and Scheduling tables. String.
force_it	Indicates whether the job is ordered or forced. Mandatory for both jobs and Scheduling tables. <ul style="list-style-type: none"> ■ no - orders job/table ■ yes - forces job/table
job_id	Occurrence number of the job within the scheduling table. String. Optional Used to identify a specific job, in case there are multiple occurrences of the same job name in the scheduling table. If the value of the job_id parameter is blank, 0, or 1, the first occurrence of the job is used.
job_name	Name of the job. Mandatory for jobs. Must be left empty for tables. String.

Table 45 request_order_force XML parameters description(part 2 of 2)

Parameter	Description
odate	Enables you to order the job with a specific date. Mandatory for both jobs and Scheduling tables. Valid values: <ul style="list-style-type: none"> ■ Numerical date (yyyymmdd format) ■ ODAT
wait_for_odate	Indicates whether the job submission should wait for a specified order date (odate), or be submitted as soon as the execution criteria for the job is satisfied. Valid values: <ul style="list-style-type: none"> ■ yes (wait for odate) ■ no (do not wait for odate -- default) Note: Relevant for CONTROL-M versions 6.2.01 or later.
with_hold	Holds all jobs immediately after they are ordered.
table_library	CONTROL-M for z/OS only: Name of the library in which the scheduling table is located. String
table_name	Name of the Scheduling table. Mandatory for both jobs and Scheduling tables. String.
scheduling_group_info	A sequence of scheduling_group_info . See Table 46 below.
autoedit_assignments	A sequence of autoedit_assignment . See Table 47 below.
max_returned_nodes	Limits the number of returned entities. Optional. Note: Should not exceed the value of the EMAPIActiveJobsLoadLimit system parameter.

Figure 27 scheduling_group_info XML parameters

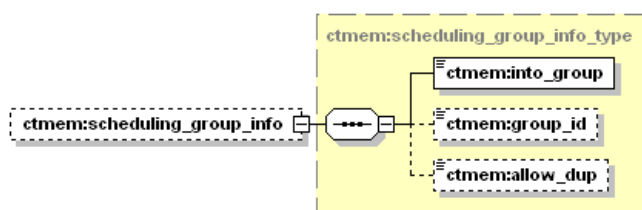
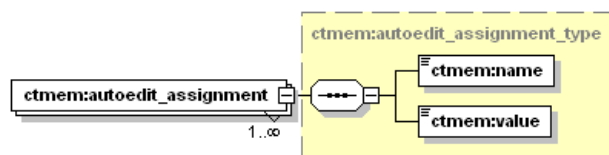


Table 46 `scheduling_group_info` XML parameters description

Parameter	Description
into_group	Indicates into which Group Scheduling table the job is placed. Optional for jobs. Not used for tables. <ul style="list-style-type: none"> ■ recent ■ new ■ standalone ■ selected ■ order ID of a Group Scheduling table
group_id	Serial number identifying the Group Scheduling table. Optional for jobs. Must be empty for tables. For more information on the <code>group_id</code> parameter, see “Job tracking” on page 184 . String.
allow_dup	Allows duplicate jobs in a Group Scheduling table. Optional for jobs. Must be left empty for tables (accepts default). Valid values: <ul style="list-style-type: none"> ■ no - Not allowed ■ yes - Allowed (default)

Figure 28 `autoedit_assignment` XML parametersTable 47 `autoedit_assignment` XML parameters description

Parameter	Description
name	Name of the AutoEdit variable. Mandatory, if the <code>autoedit_assignment</code> element is specified. String. Name sequence.
value	Value of the AutoEdit expression. String.

Response parameters

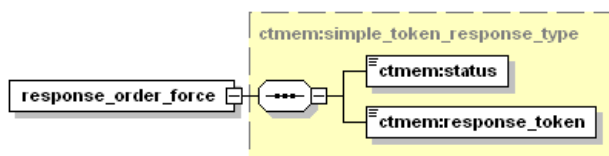
Figure 29 `response_order_force` XML parameters

Table 48 response_order_force XML parameters description

Parameter	Description
status	Description of message content. String.
response_token	Used in the polling request.

Polling request parameters

Figure 30 request_poll_order_force XML parameters

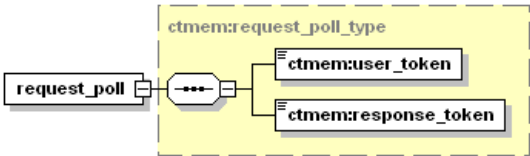


Table 49 request_poll_order_force XML parameters description

Parameter	Description
user_token	Serial identification number supplied to the user during registration. String.
response_token	Used in a polling request. This token is received in the immediate response of a response_order_force.

Polling response parameters

Figure 31 response_poll_order_force XML parameters

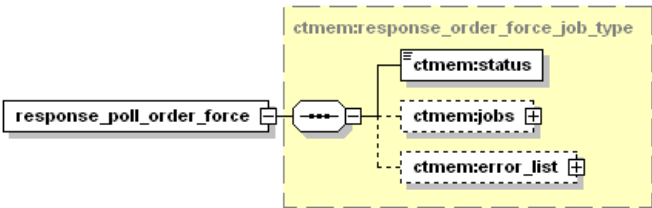


Table 50 response_poll_order_force XML parameters description

Parameter	Description
status	Describes the condition of the element that contains it.(for example, Error). String.
jobs	A sequence of job . See Table 51.
error_list	A sequence of error . See Table 142.
error_list attribute:	highest_severity Indicates the severity level of the most critical error included in the error list. If only one error is included, the severity for that error is displayed. String.

Figure 32 job XML parameters

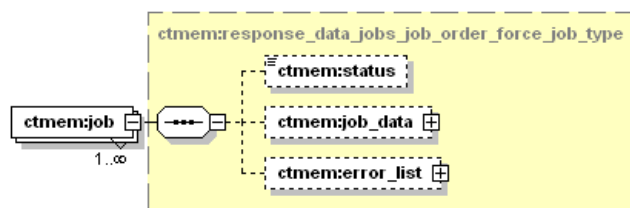


Table 51 job XML parameters description

Parameter	Description	
status	Describes the condition of the element that contains it.(for example, Error). String.	
error_list	A sequence of error . See Table 142 .	
error_list attribute:	highest_severity	Indicates the severity level of the most critical error included in the error list. If only one error is included, the severity for that error is displayed. String.
job_data	An element that contains other parameters that describe the job. A sequence of job_data . See Table 52 .	

Figure 33 job_data XML parameters

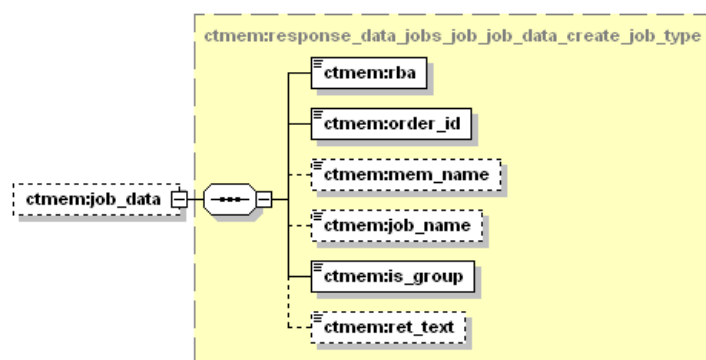


Table 52 job_data XML parameters description

Parameter	Description
rba	Relative block address. String.
order_id	Serial number assigned to the job by the CONTROL-M installation. String.
mem_name	Name of the file that contains the job script. String.
job_name	Name of the job. String.
is_group	Indicates whether the job is a member of a Group Scheduling table. Valid values: <ul style="list-style-type: none"> ■ no (not a member of a Group Scheduling table) ■ yes (member of a Group Scheduling table)
ret_text	Text describing the job run. String.

Fault response parameters

XML parameters for `fault_order_force` and `fault_poll_order_force`, as well as a sample fault response are described in [“Fault Response” on page 213](#).

Errors

See [“Order or Force request errors \(Major code 405\)” on page 259](#).

Examples

Successful example

Request

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_order_force xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>12345630</ctmem:user_token>
      <ctmem:force_it>yes</ctmem:force_it>
      <ctmem:control_m>ctm630</ctmem:control_m>
      <ctmem:table_name>apitest</ctmem:table_name>
      <ctmem:odate>ODAT</ctmem:odate>
    </ctmem:request_order_force>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

```
<?xml version="1.0" encoding="iso-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:response_order_force xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:status>OK</ctmem:status>
      <ctmem:response_token>94</ctmem:response_token>
    </ctmem:response_order_force>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Polling request

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_poll_order_force xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>12345630</ctmem:user_token>
      <ctmem:response_token>94</ctmem:response_token>
    </ctmem:request_poll_order_force>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Polling response

```
<?xml version="1.0" encoding="iso-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:response_poll_order_force xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:status>OK</ctmem:status>
      <ctmem:jobs>
        <ctmem:job>
          <ctmem:status>OK</ctmem:status>
          <ctmem:job_data>
            <ctmem:order_id>00023c</ctmem:order_id>
            <ctmem:mem_name>api MemName1</ctmem:mem_name>
            <ctmem:job_name>api test1</ctmem:job_name>
            <ctmem:ret_text>Job ordered</ctmem:ret_text>
          </ctmem:job_data>
        </ctmem:job>
      </ctmem:jobs>
      <ctmem:status>OK</ctmem:status>
      <ctmem:job_data>
        <ctmem:order_id>00023d</ctmem:order_id>
        <ctmem:mem_name>api MemName2</ctmem:mem_name>
        <ctmem:job_name>api test2</ctmem:job_name>
        <ctmem:ret_text>Job ordered</ctmem:ret_text>
      </ctmem:job_data>
    </ctmem:response_poll_order_force>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Failure example

Request

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_order_force xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>12345630</ctmem:user_token>
      <ctmem:force_it>yes</ctmem:force_it>
      <ctmem:control_m>ctm630</ctmem:control_m>
      <ctmem:job_name>inexistent_job</ctmem:job_name>
      <ctmem:table_name>api test</ctmem:table_name>
      <ctmem:odate>ODAT</ctmem:odate>
    </ctmem:request_order_force>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:response_order_force xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:status>OK</ctmem:status>
      <ctmem:response_token>95</ctmem:response_token>
    </ctmem:response_order_force>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Polling request

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_poll_order_force xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>12345630</ctmem:user_token>
      <ctmem:response_token>95</ctmem:response_token>
    </ctmem:request_poll_order_force>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Polling response

```
<?xml version="1.0" encoding="iso-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Server</faultcode>
      <faultstring>Error response from EM Server.</faultstring>
      <detail>
        <ctmem: fault_poll_order_force xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
          <ctmem: error_list ctmem: highest_severity='Error' >
            <ctmem: error ctmem: major='300' ctmem: minor='70' ctmem: severity='Error' >
              <ctmem: error_message>The requested job does not exist in the given table.
                5745 scheduling table &apos;api test&apos;, job name
                &apos;inexistent_job&apos; not found
              </ctmem: error_message>
            </ctmem: error>
          </ctmem: error_list>
        </ctmem: fault_poll_order_force>
      </detail>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Job creation

Creates a job or group scheduling table and inserts it into the Active Jobs file.

To create a regular job:

- Specify all the parameters required for the job.
- Do not specify a value for the `group_id` parameter.
- Do not specify **scheduling_group** as the value for the `task_type` parameter.

To create a group scheduling table:

- Specify **scheduling_group** as the value for the `task_type` parameter.
- Specify a name for the group scheduling table in the `group` parameter.
- Specify all other parameters required for the table.
- After the group scheduling table has been created, note the value of the `order_id` parameter. This value must be supplied for the **group_id** parameter of jobs that are associated with the group scheduling table.

To create a job in a group scheduling table:

- Indicate to which group scheduling table to associate the job by specifying:
 - The group name for the parent group scheduling table in the group parameter of the job specification.
 - The ID of the group for the parent group scheduling table in the **group_id** parameter of the job specification.

For CONTROL-M/Server (on Non-z/OS Platforms)

The group_id consists of the **order_id** of the existing group scheduling table preceded by a leading zero.

For CONTROL-M for z/OS

The **group_id** consists of the RBA of the existing group scheduling table.

If the group is not already ordered, it can be created using a separate create_aj request before creating the dependent job (see above). The group name is specified when creating the new group, and the necessary information (group name, order_id, and RBA) can be obtained from the response to this request for use when populating the group with jobs.

It is not possible to find out the group name, **order_id**, or RBA of an existing group scheduling table using the CONTROL-M/EM API.

Request parameters

Figure 34 request_create_aj XML parameters

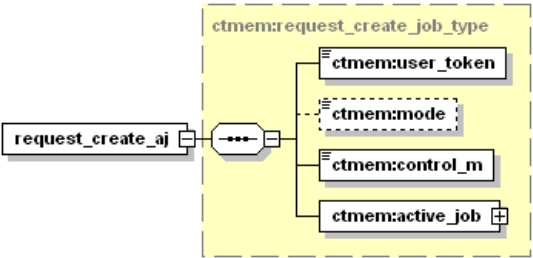


Table 53 request_create_aj XML parameters description

Parameter	Description
user_token	Serial identification number supplied to the user during registration. String.
control-m	Data center name. String. Mandatory.
active_job	Specification for a single job. The parameters of the job are included as elements between the opening and closing active_job tags. Contains no attributes. See Table 54 .

Table 54 active_job XML Parameters Description (part 1 of 7)

Parameter	Description
adjust_condition	Indicates whether to ignore prerequisite conditions normally set by predecessor jobs if the relevant predecessor jobs are not scheduled. This parameter is relevant only for jobs in a Group Scheduling table. Optional. Valid values: <ul style="list-style-type: none"> ■ no – (Do not ignore. Default.) ■ yes – (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 MEMLIB parameter of the job set to DUMMY.
application	Name of the application to which the job's group belongs. Used as a descriptive name for related groups of jobs.
application_cm_version	Indicates the version of external application (for example, SAP or Oracle Applications) Control Module (CM) that is installed in the CONTROL-M installation. Specified together with the application_form , application_type , and application_version elements.
application_form	Specifies a predefined set of external application parameters that are displayed in the External Application panel of the CONTROL-M/EM Job Editing form. Specified together with the application_cm_version , application_type , and application_version elements.
application_type	Indicates the type of external application (for example, SAP or Oracle Applications) on which the external application job runs. Specified together with the application_cm_version , application_form , and application_version elements.
application_version	Indicates the version of the external application (for example, SAP or Oracle Applications) on which the external application job runs. Specified together with the application_cm_version , application_form , and application_type elements.
arch_max_days	Maximum number of days to retain the SYSDATA archive data set for jobs that ended NOTOK . [z/OS only]
arch_max_runs	Maximum number of job runs to retain the SYSDATA archive data set for jobs that ended NOTOK . [z/OS only]
auto_archive	Determines whether or not SYSDATA is to be archived. [z/OS only] Valid values: <ul style="list-style-type: none"> ■ yes ■ no
autoedit_assignments	A sequence of autoedit_assignment . See Table 68 .
command	Command string supplied when the job Task Type (the task_type element) is Command . Optional.
confirm_flag	Specifies whether user confirmation is required before the job is submitted for execution. String. Valid values: <ul style="list-style-type: none"> ■ no - Job needs no confirmation to run. Default. ■ yes - Job must be confirmed to run.

Table 54 active_job XML Parameters Description (part 2 of 7)

Parameter	Description
control_resources	A sequence of control_resource . See Table 71 .
count_cyclic_from	Indicates whether the interval between successive runs of a cyclic job is calculated from the start or the end of the previous job run. Specified only for cyclic jobs (when the cyclic element is specified). Valid values: <ul style="list-style-type: none"> ■ start – Counts interval from the start of the previous job run ■ end – Counts interval from the end of the previous job run ■ target – Counts interval from the scheduling time of the current job run.
critical	Indicates that the job is a critical-path job in CONTROL-M. Valid values: <ul style="list-style-type: none"> ■ yes – a critical path job ■ no – not a critical path job
ctb_steps	A sequence of ctb_step . See Table 69 .
cyclic	Indicates if a job is cyclic. Valid values: <ul style="list-style-type: none"> ■ yes – Cyclic job ■ no – Non-cyclic job
cyclic_type	If job is cyclic (cyclic equal yes), indicates how the intervals for running the job are specified. Valid values are: <ul style="list-style-type: none"> ■ interval: Job is run at fixed interval. See rerun_interval. ■ interval_sequence: Job is run according to a list of time periods. See interval_sequence. ■ specific_times: Job is run according to a list of specific times. See specific_times. Note: This parameter is relevant only for jobs running in CONTROL-M/Server version 6.4.01 and later.
description	Text description of the job.
days_due_out_offset	The number of days that job execution can be extended after the ODAT. Note: days_due_out_offset is relevant only for jobs running in CONTROL-M for z/OS version 6.2.00 and later.
doc_lib	Name of the directory/library containing the job documentation file.
doc_member	Name of the file containing job documentation.
group	Name of the group to which the job belongs. Used as a descriptive name for related groups of jobs.
group_id	Order ID of the parent group scheduling table preceded by a leading zero.
in_conditions	A sequence of in_condition . See Table 70 .
instream_jcl	JCL stream forming part of the job definition. Note: instream_jcl is relevant for jobs running in: <ul style="list-style-type: none"> ■ CONTROL-M for z/OS version 6.2.00 and later ■ CONTROL-M/Server version 6.4.01 and later

Table 54 active_job XML Parameters Description (part 3 of 7)

Parameter	Description
interval_sequence	A sequence of interval_item . See Table 77 Note: This parameter is relevant only for jobs running in CONTROL-M/Server version 6.4.01 and later.
job_name	Name of the job.
max_wait	Number of extra days (beyond the original scheduling date) that the job is allowed to remain in the Active Jobs file awaiting execution. Integer.
mem_lib	Name of the library/directory in which the job script resides. String.
mem_name	Name of the file that contains the job script. String.
multiagent	When selected, broadcasts job submission details to all Agents within a specified Node Group. Not for z/OS. Optional. Valid values: <ul style="list-style-type: none"> ■ yes – run as multi-agent job ■ no – not run as multi-agent job. Default.
node_group	Host name of a node group to which the job is submitted. Not for z/OS.
odate	Original scheduling date of a job.
on_do_statements	A sequence of on_do_statement . See Table 55 .
order_table	Default or dummy Scheduling table to which you indicate the job belongs. A Scheduling table is not necessary because jobs that are created with the CONTROL-M/EM API are inserted directly into the Active Jobs file. However, you may want to include a value for this parameter so that the job can be tracked during statistical analysis that uses Scheduling table as a criterion.
order_library	Default or dummy Scheduling table library in which Scheduling table documentation is said to be stored. A Scheduling table (and, by extension, a Scheduling table library) are not necessary because jobs that are created with the CONTROL-M/EM API are inserted directly into the Active Jobs file. However, you may want to include a value for this parameter so that the job can be tracked during statistical analysis that uses Scheduling table or Scheduling Table Library as criteria. This parameter is specified only for z/OS jobs for which the order_table element was also specified.
out_conditions	A sequence of out_condition . See Table 74 .
over_lib	Name of the alternate job script library/directory.
owner	Owner (username) associated with the job.
pipes	A sequence of pipe . See Table 73 .

Table 54 active_job XML Parameters Description (part 4 of 7)

Parameter	Description
prevent_nct2	Prevents data set cleanup before the original job run. [z/OS only]. Optional. Valid values: <ul style="list-style-type: none"> ■ no – Does not prevent data set cleanup. Default. ■ yes – Prevents data set cleanup. ■ blank - Do not perform data set cleanup before the original job run. ■ 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. ■ flush - Halt processing of the job if any data set cleanup error is detected (even if MVS would not have stopped processing the job).
priority	Indicates CONTROL-M job priority. String.
quantitative_resources	A sequence of quantitative_resource . See Table 72 .
request_nje	Specifies the node in the JES network on which the job is to execute. [z/OS only] String.
rerun_interval	Specifies the length of time to wait between reruns of a job or between cyclic runs of a job. The value is expressed as a number and a letter. The number indicates the amount. The letter indicates the unit of measurement. Valid values: <ul style="list-style-type: none"> ■ 0 - 64800M (minutes) ■ 0 - 1080H (hours) ■ 0 - 45D (days) Default: 0 .
rerun_max	Specifies the maximum number of reruns that can be performed for the job. Integer. Valid values: 0-99 .
rerun_member	Name of the JCL member to use when the job is automatically rerun. [z/OS only] String. Optional.
reten_days	Number of days to retain the job in the History Jobs file. [z/OS, only]. String.
reten_gen	Maximum number of generations of the job to keep in the History Jobs file.[z/OS, only]. String.
schedule_environment	Indicates the JES2 workload management scheduling environment that is to be associated with the job. z/OS, only. String.
shouts	A sequence of shout . See Table 76 .
specific_times	A sequence of specific_time . See Table 78 . Note: This parameter is relevant only for jobs running in CONTROL-M/Server version 6.4.01 and later.
statistic_calendar	Name of the CONTROL-M periodic calendar within which statistics relating to the job are collected.
	Note: statistic_calendar is relevant only for jobs running in CONTROL-M for z/OS version 6.2.00 and later.
step_ranges	A sequence of step_range . See Table 75 .

Table 54 active_job XML Parameters Description (part 5 of 7)

Parameter	Description
sys_db	Indicates that a single data set is used for archiving the SYSDATA of all jobs until it is full, when another data set is started. [z/OS only] Valid values: <ul style="list-style-type: none"> ■ yes – Single data set created for the SYSDATA of each job run. ■ no – Separate data set created for the SYSDATA of each job run.
sysout_from_class	Limits the sysout handling operation to only sysouts from the specified class. [z/OS only]
sysout_option	Sysout Handling options. Optional. Valid values (non-z/OS): <ul style="list-style-type: none"> ■ copy ■ delete ■ move ■ release Valid values (z/OS): <ul style="list-style-type: none"> ■ copy ■ delete ■ move ■ release ■ change_class
sysout_parameter	Certain sysout_option values require that you supply additional information (such as Copy , NewDest): <ul style="list-style-type: none"> ■ If the sysout_option element is change_class, the sysout_parameter value corresponds to the new class name. ■ If the sysout_option element is copy, the sysout_parameter value corresponds to the destination file name. ■ If the sysout_option element is move, the sysout_parameter value corresponds to the new destination for the file.
system_affinity	Indicates the identity of the system in which the job must be initiated and executed (in JES2). Indicates the identity of the processor on which the job must execute (in JES3). Note: For z/OS jobs only.
task_class	CONTROL-D mission. Mandatory for CONTROL-D jobs. Valid values: <ul style="list-style-type: none"> ■ distribution ■ decollation

Table 54 active_job XML Parameters Description (part 6 of 7)

Parameter	Description
task_type	<p>Type of the job (task) to be performed by CONTROL-M.</p> <p>Microsoft Windows and UNIX</p> <ul style="list-style-type: none"> ■ job ■ command ■ dummy ■ detached ■ external ■ scheduling_group <p>CONTROL-M for z/OS</p> <ul style="list-style-type: none"> ■ job ■ task ■ scheduling_group ■ cyclic_job ■ emergency_job ■ emergency_cyclic_job ■ cyclic_task ■ emergency_task ■ emergency_cyclic_task
time_due_out	Time that the job is expected to finish.
time_from	Indicates the earliest time for submitting the job.
time_from_days_offset	<p>Number of days after the original scheduling date of the job during which execution of the job can begin.</p> <p>Note: time_from_days_offset is relevant only for jobs running in CONTROL-M for z/OS version 6.2.00 and later.</p>
time_reference	<p>Valid values:</p> <ul style="list-style-type: none"> ■ server ■ adjust
time_until	Indicates the latest time for submitting the job.
time_until_days_offset	<p>Number of days after the original scheduling date of the job during which execution of the job can end.</p> <p>Note: time_until_days_offset is relevant only for jobs running in CONTROL-M for z/OS version 6.2.00 and later.</p>
time_zone	Indicates the global time zone used to calculate the interval for time-related conditions.
tolerance	<p>Maximum delay in minutes permitted for late submission when selecting a specific time.</p> <p>Note: This parameter is relevant only for jobs running in CONTROL-M/Server version 6.4.01 and later.</p>

Table 54 active_job XML Parameters Description (part 7 of 7)

Parameter	Description
use_instream_jcl	Whether the job submits a JCL stream defined within the job scheduling definition. Valid values are: <ul style="list-style-type: none"> ■ yes ■ no
	Note: use_instream_jcl is relevant for jobs running in: <ul style="list-style-type: none"> ■ CONTROL-M for z/OS version 6.2.00 and later ■ CONTROL-M/Server version 6.4.01 and later

Figure 35 on_do_statement XML parameters

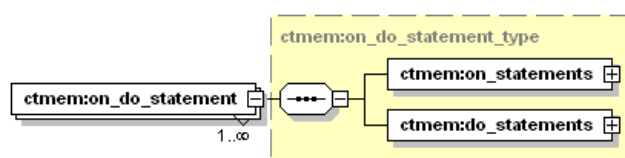


Table 55 on_do_statement XML parameters description

Parameter	Description
on_statements	A sequence of on_statements . See Table 56 .
do_statements	A sequence of do_statements . See Table 59 .

Figure 36 on_statements XML parameters

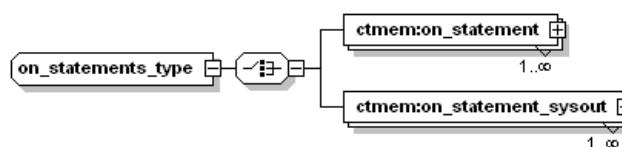


Table 56 on_statements XML parameters description

Parameter	Description
on_statement	A sequence of on_statement . See Table 57
OR	
on_statement_sysout	A sequence of on_statement_sysout . See Table 58

Figure 37 on_statement XML parameters

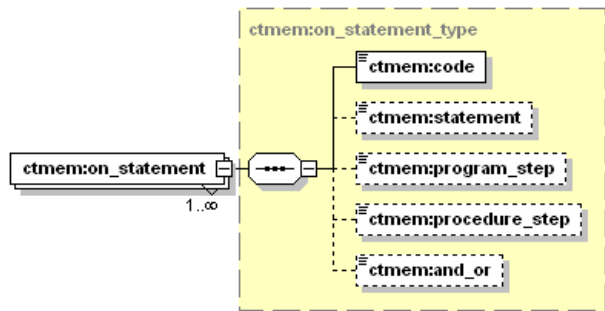


Table 57 on_statement XML parameters description

Parameter	Description
and_or	Specifies the relationship between two successive items in a series. Optional.
code	Code value for the On Statement/Code parameter. Valid values: <ul style="list-style-type: none">■ ok■ not_ok
procedure_step	Step in the procedure that triggers the On statement. String.
program_step	Step in the program that triggers the On statement. String.
statement	statement can be: <ul style="list-style-type: none">■ A character string containing a statement from the job script file (1-132 characters). The specified string can be a portion of the statement.■ An asterisk (*), when code is a completion status for a job.

Figure 38 on_statement_sysout XML parameters

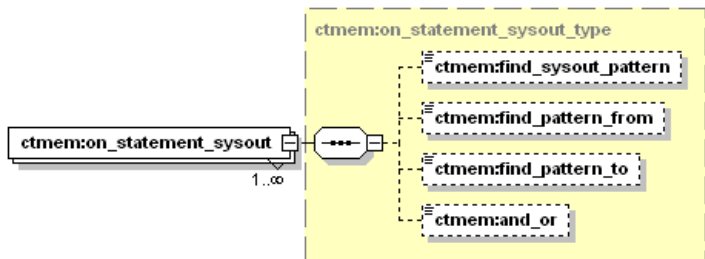


Table 58 on_statement_sysout XML parameters description

Parameter	Description
find_sysout_pattern	A string of up to 40 characters.
find_pattern_from	A number from 001 through 132, indicating the column at which the search should start. If this field is blank, the value 001 is assumed. The value in this field must be lower than that in the To Column field.

Table 58 on_statement_sysout XML parameters description

Parameter	Description
find_pattern_to	A number from 001 through 132, indicating the column at which the search should end. If this field is blank, the value 132 is assumed. The value in this field must be higher than that in the From Column field.
and_or	Option buttons that set the logical relationship between multiple On statements.

Figure 39 do statements type XML parameters

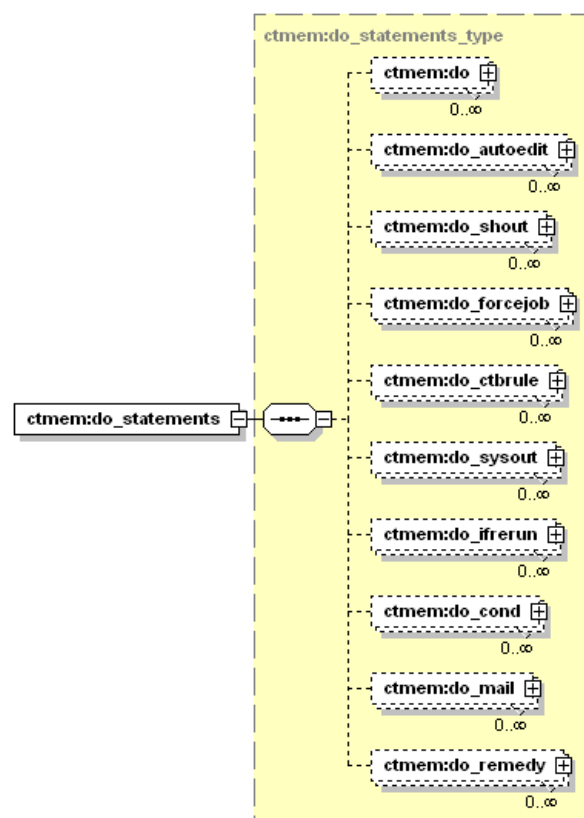


Table 59 do statements type XML parameters description (part 1 of 2)

Parameter	Description
do	A sequence of do_statements .
do_autoedit	Assigns an AutoEdit variable when the On criteria are met. See Table 60 .
do_cond	Assigns an In or Out condition when the On criteria are met. See Table 61 .
do_ctbrule	Invokes a CONTROL-M/Analyzer rule to be executed during the processing of a specific program step when an On condition is met. See Table 62 .
do_forcejob	Forces a specified job when the current job is performed. Note: The dsn element is for z/OS jobs only. See Table 63 .
do_ifrerun	Specifies job steps to be executed during rerun of a job. Only for networks using CONTROL-M/Restart. See Table 64 .
do_mail	Sends e-mail.

Table 59 do statements type XML parameters description (part 2 of 2)

Parameter	Description
do_remedy	Creates a remedy ticket.
do_shout	Sends a shout message when the On criteria are met.
do_sysout	Determines what to do with the sysout documentation when On criteria are met.

Figure 40 do_autoedit XML parameters

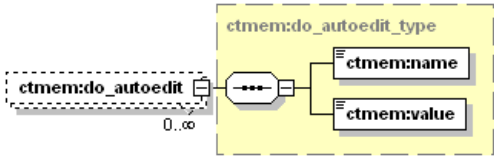


Table 60 do_autoedit XML parameters description

Parameter	Description
name	Name of the item in question (for example, when specified for request , name is the name of the request; when specified for pipe, name is the name of the pipe)
value	Value of the AutoEdit expression.

Figure 41 do_cond XML parameters

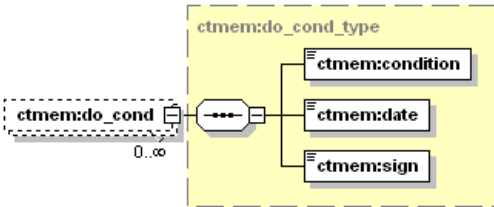


Table 61 do_cond XML parameters description

Parameter	Description
condition	Condition name. When specified, it is be accompanied by the other condition parameter element, date (and, optionally, by sign or and_or). ■ Wrapped in the in_condition and out_condition elements.
date	Specifies an order date for various condition formats.
sign	Indicates whether to add or delete an Out condition Valid values: ■ add ■ delete

Figure 42 do_ctbrule XML parameters

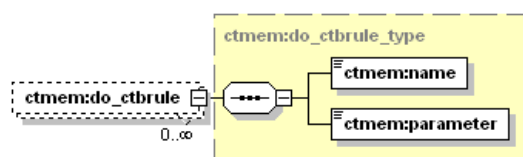


Table 62 do_ctbrule XML parameters description

Parameter	Description
name	Name of the CONTROL-M/Analyzer rule.
parameter	Contains arguments that are passed to the CONTROL-M/Analyzer rule.

Figure 43 do_forcejob XML parameters

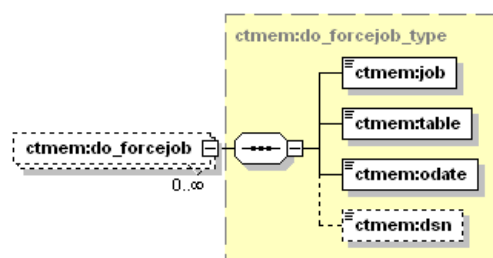


Table 63 do_forcejob XML parameters description

Parameter	Description
dsn	Name of the directory/library containing Scheduling table file. [z/OS only]
job	Specifies the job name of the job that is forced.
odate	Original scheduling date of a job.
table	Name of the Scheduling table with which the job specified in <code>do_forcejob</code> is associated.

Figure 44 do_ifrerun XML parameters

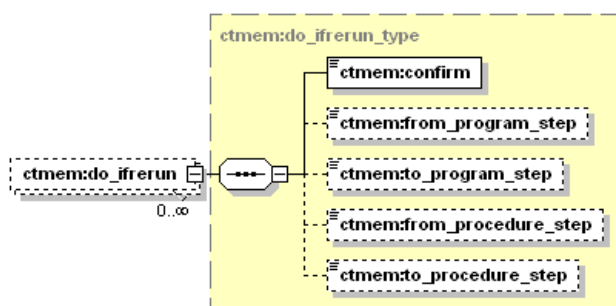


Table 64 do_ifrerun XML parameters description

Parameter	Description
confirm	Indicates that a job rerun specified by the Do If Rerun parameter must be manually confirmed before it is executed. Valid values: <ul style="list-style-type: none"> ■ yes – requires confirmation ■ no – no confirmation required
from_procedure_step	Procedure step (EXEC statement) that invokes a procedure from which the specified program step program is executed.
from_program_step	Job step. The execution results of the program executed by the job step are checked against the specified codes criteria.
to_procedure_step	Last procedure step in a range.
to_program_step	Last program step in a range.

Figure 45 do_mail XML parameters

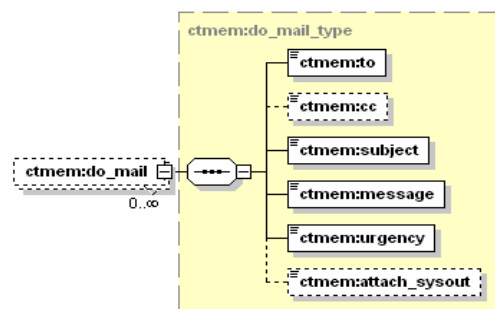


Table 65 do_mail XML parameters description

Parameter	Description
cc	Optional additional address to which a Do Mail can be sent. Optional.
message	Text of the message. String.
to	Recipient of the do_mail message.
subject	Subject of the do_mail message.
urgency	Indicates the severity of a mail or shout message. Valid values: <ul style="list-style-type: none"> ■ regular (Default) ■ urgent ■ very_urgent
attach_sysout	Specifies whether the sysout should be sent as an e-mail attachment. Valid values are: <ul style="list-style-type: none"> ■ yes - Send the job's sysout as an attachment ■ no - Do not send the job's sysout as an attachment ■ default - Use the settings configured for the relevant CONTROL-M server to determine whether the job's sysout should be sent as an attachment. Note: This parameter is relevant only for jobs running in CONTROL-M/Server version 6.4.01 and later.

Figure 46 do_shout XML parameters

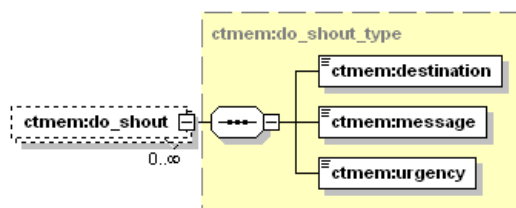


Table 66 do_shout XML parameters description

Parameter	Description
destination	Recipient of a Shout message. Specified in both the Shout or the Do Shout parameters.
message	Text of the message. String.
urgency	Indicates the severity of a mail or shout message. Valid values: <ul style="list-style-type: none"> ■ regular (Default) ■ urgent ■ very_urgent

Figure 47 do_sysout XML parameters

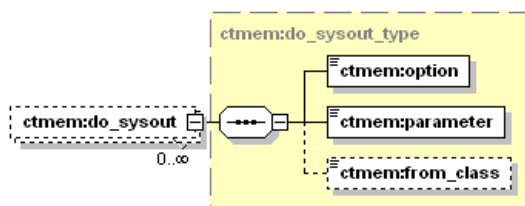


Table 67 do_sysout XML parameters description

Parameter	Description
from_class	Specifies the class of jobs with sysouts that are handled using the Do Sysout specifications of the job.

Table 67 do_sysout XML parameters description

Parameter	Description
option	Do Sysout parameter sysout handling options. Valid values: <ul style="list-style-type: none">■ Release■ Delete■ Copy■ Move■ File■ NewDest■ ChangeClass Note: Copy and Move are not used with z/OS. File, NewDest, and ChangeClass are used only with z/OS.
parameter	Contains additional sysout handling information. The type of information required is dependent on the value of the option element. <ul style="list-style-type: none">■ If the option element is ChangeClass, the parameter value corresponds to the new class name.■ If the option element is Copy, the parameter value corresponds to the destination file name.■ If the option element is Move, the parameter value corresponds to the new destination for the file.

Figure 48 autoedit_assignment XML parameters

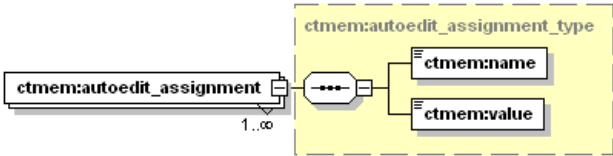


Table 68 autoedit_assignment XML parameters description

Parameter	Description
name	Name of theAutoEdit variable.
value	Value of the AutoEdit expression.

Figure 49 ctb_step XML parameters

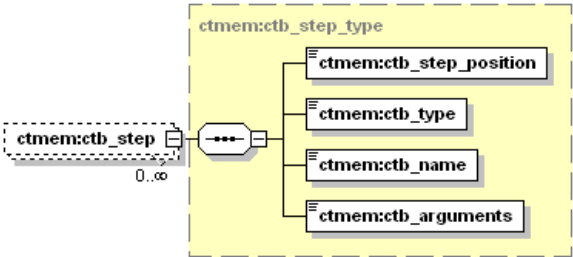


Table 69 ctb_step XML parameters description

Parameter	Description
ctb_arguments	CONTROL-M/Analyzer argument.
ctb_name	Name of the CONTROL-M/Analyzer entity. Must be a valid name of a CONTROL-M/Analyzer rule or mission.
ctb_step_position	Indicates where to place the CONTROL-M/Analyzer step in the job.
ctb_type	Type of CONTROL-M/Analyzer entity.

Figure 50 in_condition XML parameters

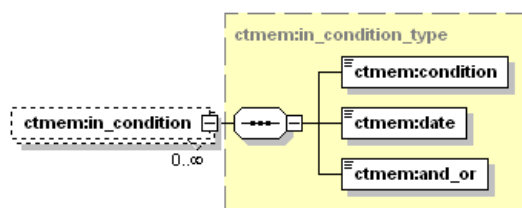


Table 70 in_condition XML parameters description

Parameter	Description
and_or	Specifies the relationship between two successive items in a series. Optional. Valid values: ■ and ■ or
condition	Condition name.
date	Specifies an order date for various condition formats.

Figure 51 control_resources XML parameters

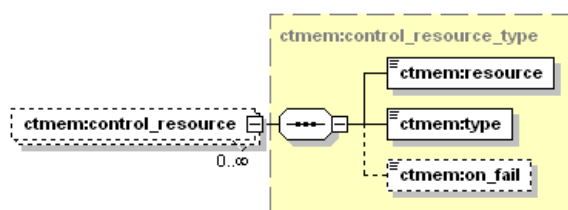


Table 71 control_resources XML parameters description

Parameter	Description
resource	Name of the specified resource.

Table 71 control_resources XML parameters description

Parameter	Description
type	Indicates job access to a Control resource. Valid values are: <ul style="list-style-type: none">■ exclusive - default■ shared
on_fail	Whether to keep a Control resource tied to a job if the job does not end OK. Valid values: <ul style="list-style-type: none">■ keep■ release - default

Figure 52 quantitative_resource XML parameters

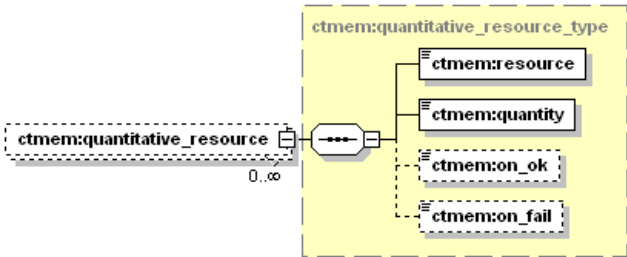


Table 72 quantitative_resource XML parameters description

Parameter	Description
quantity	Amount of the specified quantitative resource.
resource	Name of the specified resource.
on_ok	Whether to keep a Quantitative resource tied to a job if the job ends OK. Valid values are: <ul style="list-style-type: none">■ release■ discard Note: on_ok is relevant only for jobs running in CONTROL-M for z/OS version 6.2.00 and later.
on_fail	Whether to keep a Quantitative resource tied to a job if the job does not end OK. Valid values are: <ul style="list-style-type: none">■ keep■ release Note: on_fail is relevant only for jobs running in CONTROL-M for z/OS version 6.2.00 and later.

Figure 53 pipe XML parameters

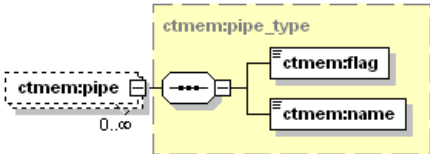


Table 73 pipe XML parameters description

Parameter	Description
flag	Valid values: <ul style="list-style-type: none"> ■ yes – ■ no –
name	Name of the item in question (for example, when specified for request , name is the name of the request; when specified for pipe, name is the name of the pipe)

Figure 54 out_condition XML parameters

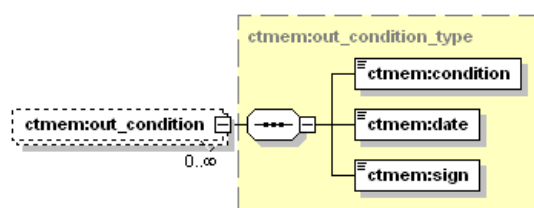


Table 74 out_condition XML parameters description

Parameter	Description
condition	Condition name. When specified, it is be accompanied by the other condition parameter element, date (and, optionally, by sign or and_or).
date	Specifies an order date for various condition formats.
sign	Indicates whether to add or delete an Out condition Valid values: <ul style="list-style-type: none"> ■ add ■ delete

Figure 55 step_range XML parameters

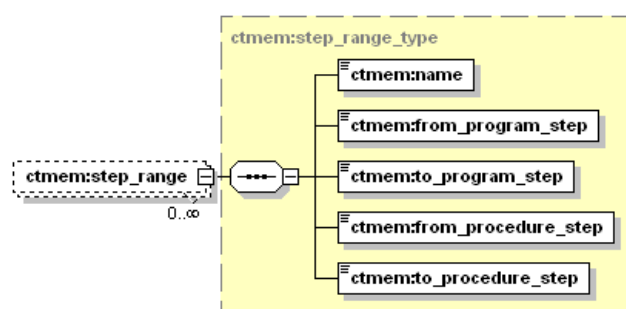


Table 75 step_range XML parameters description

Parameter	Description
name	Name of the item in question.
from_procedure_step	Procedure step (EXEC statement) that invokes a procedure from which the specified program step program is executed.

Table 75 step_range XML parameters description

Parameter	Description
from_program_step	Job step. The execution results of the program executed by the job step are checked against the specified codes criteria.
to_procedure_step	Last procedure step in a range.
to_program_step	Last program step in a range.

Figure 56 shouts XML parameters

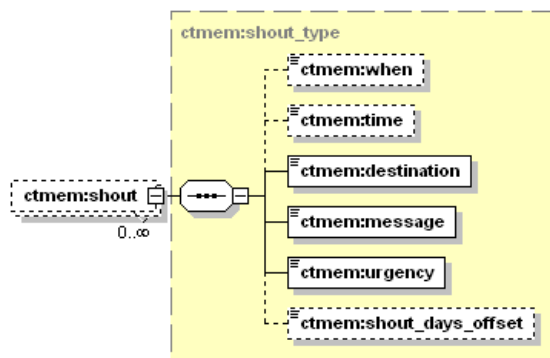


Table 76 shouts XML parameters description

Parameter	Description
destination	Recipient of a Shout message. Specified in both the Shout or the Do Shout parameters.
message	Text of the message. String.
time	Time that the message is sent.
urgency	Indicates the severity of a mail or shout message. Valid values: <ul style="list-style-type: none"> ■ regular (Default) ■ urgent ■ very_urgent
when	Time that the Shout message was sent. Valid values: <ul style="list-style-type: none"> ■ ok ■ not_ok ■ rerun (not valid for scheduling group entities) ■ late_submission ■ late_time ■ execution_time
shout_days_offset	The number of days relative to the ODAT by which the sending of the Shout message is offset. Valid values are: <ul style="list-style-type: none"> ■ a number from 0 through 254 ■ blank – no offset Note: shout_days_offset is relevant only for jobs running in CONTROL-M for z/OS version 6.2.00 and later.

Figure 57 interval_sequence XML parameters

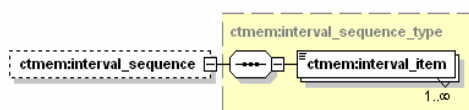


Table 77 interval_sequence XML parameters description

Parameter	Description
<code>interval_item</code>	Time interval to rerun a cyclic job such as +2H, +1D, or +30M. Limited to 4000 characters for all fields.

Figure 58 specific_times XML parameters

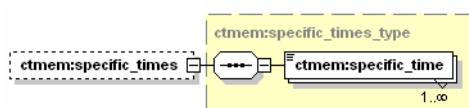


Table 78 specific_times XML parameters description

Parameter	Description
<code>specific_time</code>	Specific time for a cyclic job to run, such as 7:00 or 11:00. Limited to 4000 for all fields.

Response parameters

Figure 59 response_create_aj XML parameters

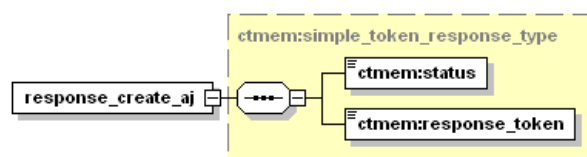


Table 79 response_create_aj XML parameters description

Parameter	Description
<code>status</code>	Description of message content. String.
<code>response_token</code>	Used in the polling request.

Polling request parameters

Figure 60 request_poll_create aj XML parameters

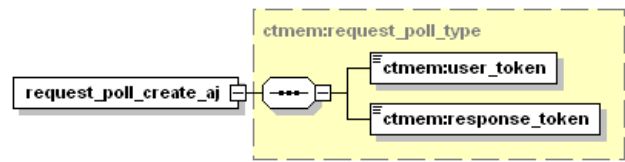


Table 80 request_poll_create aj XML parameters description

Parameter	Description
user_token	Serial identification number supplied to the user during registration. String.
response_token	Used in a polling request. This token is received in the immediate response of a response_create aj.

Polling response parameters

Responses to Job and Group Scheduling table creation requests are sent in a format conforming to the schema.

Figure 61 response_poll_create aj XML parameters

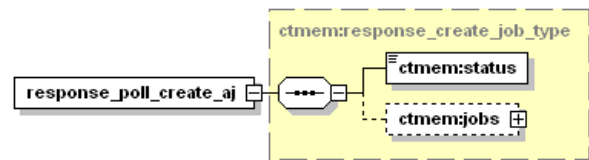


Table 81 response_poll_create aj XML parameters description

Parameter	Description
status	Describes the condition of the element that contains it.(for example, Error). String.
jobs	A sequence of jobs . See Table 82 .

Figure 62 job XML parameters

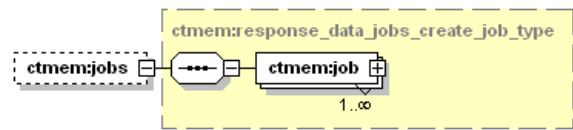


Table 82 jobs XML Parameters Description

Parameter	Description
job	Tags that indicate a single job. A sequence of job . See Table 83

Figure 63 job XML parameters

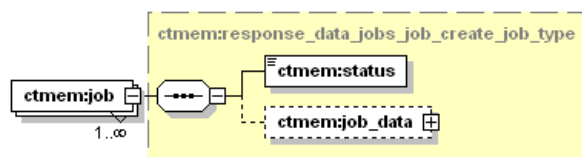


Table 83 job XML parameters description

Parameter	Description
status	Describes the condition of the element that contains it.(for example, Error). String.
job_data	An element that contains other parameters that describe the job. See Table 84 .

Figure 64 job_data XML parameters

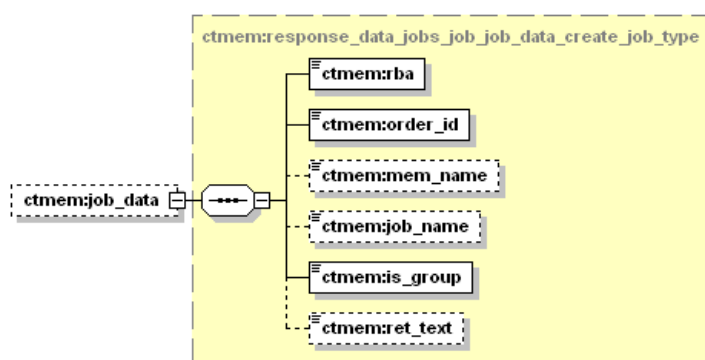


Table 84 job_data XML parameters description

Parameter	Description
rba	Relative block address. String.
order_id	Serial number assigned to the job by the CONTROL-M installation. String.
mem_name	Name of the file that contains the job script. String.
job_name	Name of the job. String.
is_group	Indicates whether the job is a member of a Group Scheduling table. Valid values: <ul style="list-style-type: none"> ■ no (not a member of a Group Scheduling table) ■ yes (member of a Group Scheduling table)
ret_text	Text describing the job run. String.

Fault response parameters

XML parameters for `fault_create_aj` and `fault_poll_create_aj`, as well as a sample fault response are described in [“Fault Response” on page 213](#).

Errors

Job creation errors are described in [“Create active job request errors \(Major code 409\)” on page 261.](#)

Examples

Successful example

Request

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_create_job xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>12345630</ctmem:user_token>
      <ctmem:control_m>ctm630</ctmem:control_m>
      <ctmem:active_job>
        <ctmem:job_name>MYJOB</ctmem:job_name>
        <ctmem:owner>control m</ctmem:owner>
        <ctmem:task_type>command</ctmem:task_type>
        <ctmem:application>MYAPP</ctmem:application>
        <ctmem:group>MYGROUP1</ctmem:group>
        <ctmem:command>ls</ctmem:command>
      </ctmem:active_job>
    </ctmem:request_create_job>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

```
<?xml version="1.0" encoding="iso-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:response_create_job xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:status>OK</ctmem:status>
      <ctmem:response_token>96</ctmem:response_token>
    </ctmem:response_create_job>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Polling request

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_poll_create_aj xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>12345630</ctmem:user_token>
      <ctmem:response_token>96</ctmem:response_token>
    </ctmem:request_poll_create_aj>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Polling response

```
<?xml version="1.0" encoding="iso-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:response_poll_create_aj xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:status>OK</ctmem:status>
      <ctmem:jobs>
        <ctmem:job>
          <ctmem:status>OK</ctmem:status>
          <ctmem:job_data>
            <ctmem:rba>000000</ctmem:rba>
            <ctmem:order_id>0023e</ctmem:order_id>
            <ctmem:mem_name>
            </ctmem:mem_name>
            <ctmem:job_name>MYJOB</ctmem:job_name>
            <ctmem:is_group>no</ctmem:is_group>
          </ctmem:job_data>
        </ctmem:job>
      </ctmem:jobs>
    </ctmem:response_poll_create_aj>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Failure example

Request

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_create_job xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>12345630</ctmem:user_token>
      <ctmem:control_m>Failure_ctm_name</ctmem:control_m>
      <ctmem:active_job>
        <ctmem:job_name>MYJOB</ctmem:job_name>
        <ctmem:owner>control m</ctmem:owner>
        <ctmem:task_type>command</ctmem:task_type>
        <ctmem:application>MYAPP</ctmem:application>
        <ctmem:group>MYGROUP1</ctmem:group>
        <ctmem:command>ls</ctmem:command>
      </ctmem:active_job>
    </ctmem:request_create_job>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

```
<?xml version="1.0" encoding="iso-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Server</faultcode>
      <faultstring>Error response from EM Server.</faultstring>
      <detail>
        <ctmem:fault_create_job xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
          <ctmem:error_list ctmem:highest_severity='Error'>
            <ctmem:error ctmem:major='401' ctmem:minor='2' ctmem:severity='Error'>
              <ctmem:error_message>Invalid Control -M.</ctmem:error_message>
            </ctmem:error>
          </ctmem:error_list>
        </ctmem:fault_create_job>
      </detail>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Add condition

Adds prerequisite conditions.

Request parameters

Figure 65 request_add_condition XML Parameters

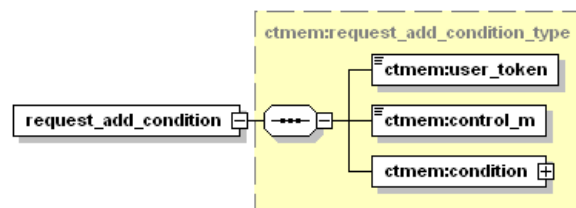


Table 85 request_add_condition XML Parameters Description

Parameter	Description
user_token	Serial identification number supplied to the user during registration. String.
control_m	Name of the CONTROL-M installation that processes the request. String.
condition	Condition description wrapper. Wrapper for the name and odate elements that identify the specific condition being added or deleted. Condition. See Table 86 , below.

Figure 66 condition XML parameters

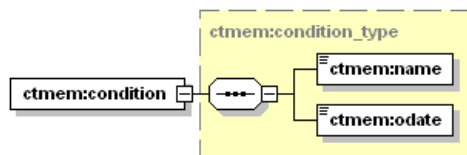


Table 86 condition XML parameters description

Parameter	Description
name	Name of the condition to be added or deleted. String.
odate	Order date of the condition. String. Valid values: <ul style="list-style-type: none"> ■ mmdd ■ STAT

Response parameters

Figure 67 response_add_condition XML parameters

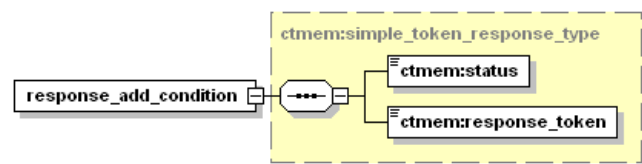


Table 87 response_add_condition XML parameters description

Parameter	Description
status	Description of message content. String.
response_token	Used in the polling request.

Polling request parameters

Figure 68 request_poll_add_condition XML parameters

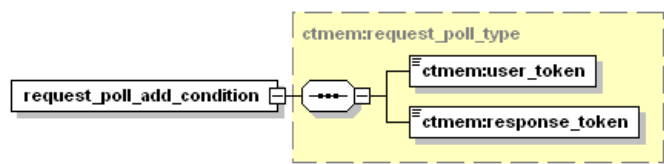


Table 88 request_poll_add_condition XML parameters description

Parameter	Description
user_token	Serial identification number supplied to the user during registration. String.
response_token	Used in a polling request. This token is received in the immediate response of a response_add_condition.

Polling response parameters

Figure 69 response_poll_add_condition XML parameters

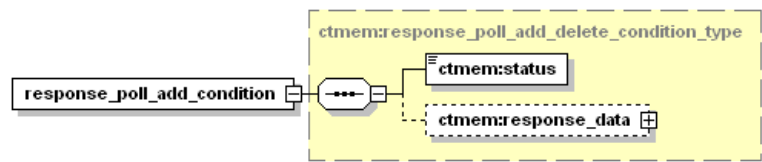


Table 89 response_poll_add_condition XML parameters description

Parameter	Description
status	Description of message content (OK or EXEC). String.
response_data	See Table 96

Figure 70 response_data XML parameters

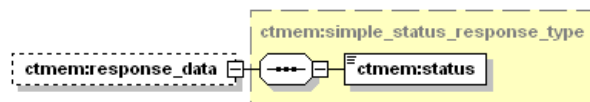


Table 90 response_data XML parameters description

Parameter	Description
status	Description of message content. String.

Fault response parameters

XML parameters for `fault_add_condition` and `fault_poll_add_condition`, as well as a sample fault response are described in [“Fault Response”](#) on page 213.

Errors

See [“Add or Delete Condition request errors \(Major code 404\)”](#) on page 259.

Examples

Successful example

Request

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_add_condition xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>12345630</ctmem:user_token>
      <ctmem:control_m>ctm630</ctmem:control_m>
      <ctmem:condition>
        <ctmem:name>MYJOBOUTCOND1</ctmem:name>
        <ctmem:odate>STAT</ctmem:odate>
      </ctmem:condition>
    </ctmem:request_add_condition>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:response_add_condition xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:status>OK</ctmem:status>
      <ctmem:response_token>97</ctmem:response_token>
    </ctmem:response_add_condition>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Polling request

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_poll_add_condition xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>12345630</ctmem:user_token>
      <ctmem:response_token>97</ctmem:response_token>
    </ctmem:request_poll_add_condition>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Polling response

```
<?xml version="1.0" encoding="iso-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:response_poll_add_condition xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:status>OK</ctmem:status>
    </ctmem:response_poll_add_condition>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Failure example

Request

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_add_condition xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>12345630</ctmem:user_token>
      <ctmem:control_m>ctm630</ctmem:control_m>
      <ctmem:condition>
        <ctmem:name>MYJOBOUTCOND1</ctmem:name>
        <ctmem:odate>STAT</ctmem:odate>
      </ctmem:condition>
    </ctmem:request_add_condition>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

```
<?xml version="1.0" encoding="iso-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:response_add_condition xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:status>OK</ctmem:status>
      <ctmem:response_token>98</ctmem:response_token>
    </ctmem:response_add_condition>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Polling request

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_poll_add_condition xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>12345630</ctmem:user_token>
      <ctmem:response_token>98</ctmem:response_token>
    </ctmem:request_poll_add_condition>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Polling response

```
<?xml version="1.0" encoding="iso-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Server</faultcode>
      <faultstring>Error response from EM Server. </faultstring>
      <detail>
        <ctmem:fault_poll_add_condition xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
          <ctmem:error_list ctmem:highest_severity='Error' >
            <ctmem:error ctmem:major='404' ctmem:minor='14' ctmem:severity='Error' >
              <ctmem:error_message>Connot add condition, already exist.
            </ctmem:error_message>
            </ctmem:error>
          </ctmem:error_list>
        </ctmem:fault_poll_add_condition>
      </detail>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Delete condition

Deletes prerequisite conditions.

Request parameters

Figure 71 request_delete_condition XML parameters

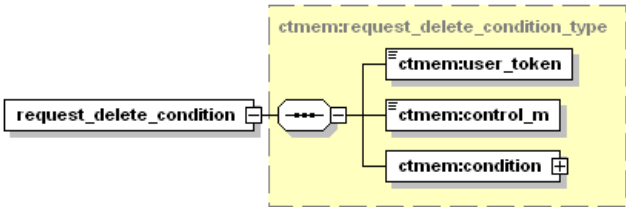


Table 91 request_delete_condition XML parameters description

Parameter	Description
control_m	Name of the CONTROL-M installation that processes the request. String.
condition	Condition description wrapper. See Table 92, below.

Figure 72 condition XML parameters

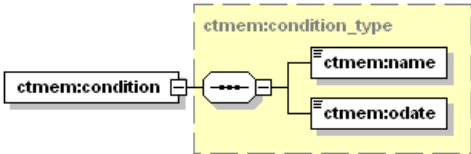


Table 92 condition XML parameters description

Parameter	Description
name	Name of the condition to be added or deleted. String.
odate	Order date of the condition. String. Valid values: <ul style="list-style-type: none">■ mmdd■ STAT

Response parameters

Figure 73 response_delete_condition XML parameters

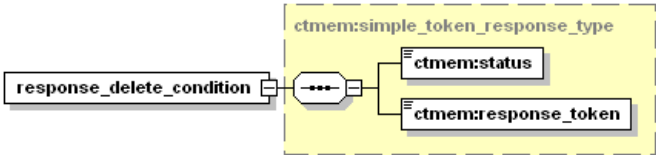


Table 93 response_delete_condition XML parameters description

Parameter	Description
status	Description of message content. String.
response_token	Used in the polling request.

Polling request parameters

Figure 74 request_poll_delete_condition XML parameters

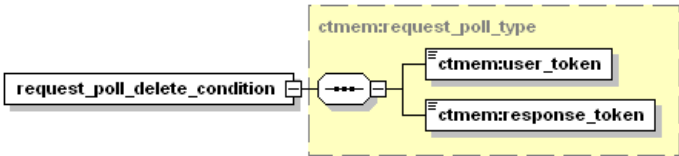


Table 94 request_poll_delete_condition XML parameters description

Parameter	Description
user_token	Serial identification number supplied to the user during registration. String.
response_token	Used in a polling request. This token is received in the immediate response of a response_delete_condition.

Polling response parameters

Figure 75 response_poll_delete_condition XML parameters

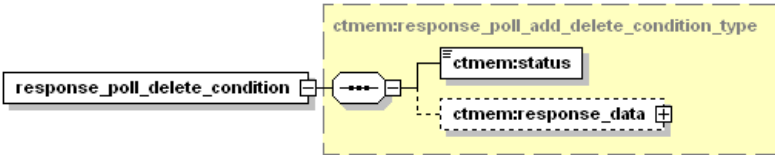


Table 95 response_poll_delete_condition XML parameters description

Parameter	Description
status	Description of message content (OK or EXEC). String.
response_data	See Table 96

Figure 76 response_data_XML parameters

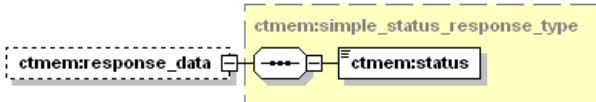


Table 96 response_data XML parameters description

Parameter	Description
status	Description of message content. String.

Fault response parameters

XML parameters for `fault_delete_condition` and `fault_poll_delete_condition`, as well as a sample fault response are described in [“Fault Response” on page 213](#).

Errors

See [“Add or Delete Condition request errors \(Major code 404\)” on page 259](#).

Examples

Successful example

Request

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_delete_condition xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>12345630</ctmem:user_token>
      <ctmem:control_m>ctm630</ctmem:control_m>
      <ctmem:condition>
        <ctmem:name>MYJOBOUTCOND1</ctmem:name>
        <ctmem:odate>STAT</ctmem:odate>
      </ctmem:condition>
    </ctmem:request_delete_condition>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

```
<?xml version="1.0" encoding="iso-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:response_delete_condition xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:status>OK</ctmem:status>
      <ctmem:response_token>99</ctmem:response_token>
    </ctmem:response_delete_condition>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Polling request

```
<?xml version="1.0" encoding="iso-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_poll_delete_condition xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>12345630</ctmem:user_token>
      <ctmem:response_token>99</ctmem:response_token>
    </ctmem:request_poll_delete_condition>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Polling response

```
<?xml version="1.0" encoding="iso-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:response_poll_delete_condition xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:status>OK</ctmem:status>
    </ctmem:response_poll_delete_condition>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```


Failure example

Request

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_delete_condition xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>12345630</ctmem:user_token>
      <ctmem:control_m>ctm630</ctmem:control_m>
      <ctmem:condition>
        <ctmem:name>INEXTENT_COND_NAME</ctmem:name>
        <ctmem:odate>STAT</ctmem:odate>
      </ctmem:condition>
    </ctmem:request_delete_condition>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:response_delete_condition xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:status>OK</ctmem:status>
      <ctmem:response_token>101</ctmem:response_token>
    </ctmem:response_delete_condition>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Polling request

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_poll_delete_condition xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>12345630</ctmem:user_token>
      <ctmem:response_token>101</ctmem:response_token>
    </ctmem:request_poll_delete_condition>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Polling response

```
<?xml version="1.0" encoding="iso-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Server</faultcode>
      <faultstring>Error response from EM Server.</faultstring>
      <detail>
        <ctmem:fault_poll_delete_condition xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
          <ctmem:error_list ctmem:highest_severity='Error'>
            <ctmem:error ctmem:major='404' ctmem:minor='15' ctmem:severity='Error'>
              <ctmem:error_message>Cannot delete condition, does not exist.</ctmem:error_message>
            </ctmem:error>
          </ctmem:error_list>
        </ctmem:fault_poll_delete_condition>
      </detail>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Job actions in active environment

Performs actions on jobs that are currently in the active environment. This request type allows the following job actions in the active environment for a single job:

- **Hold**
- **Free**
- **Confirm**
- **Rerun**
- **Kill**
- **Force OK**

These requests are asynchronous. The immediate response will indicate if the action request is sent (or not sent to CTM). In case the request is successfully sent to CTM, the response will contain a response token. The returned response token can be used to get the final response using the existing polling request.

Hold

Request parameters

Figure 77 request aj_hold XML parameters

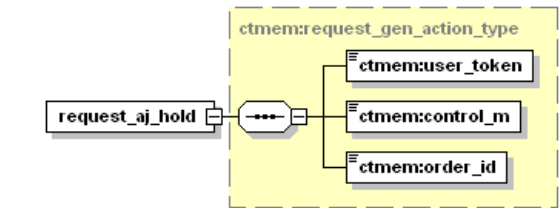


Table 97 request aj_hold XML parameters description

Parameter	Description
user_token	Serial identification number supplied to the user during registration. String.
control_m	CONTROL-M installation to which the job belongs. String. Mandatory.
order_id	Order ID of the job. String. Mandatory.

Response parameters

Figure 78 response aj_hold XML parameters

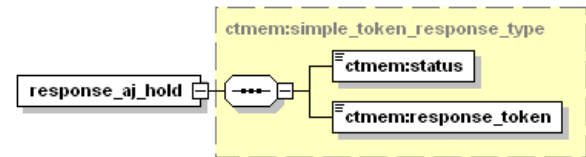


Table 98 response aj_hold XML parameters description

Parameter	Description
status	The status of the response.
response_token	Used in a polling request.

Polling request parameters

Figure 79 request_poll aj_hold XML parameters

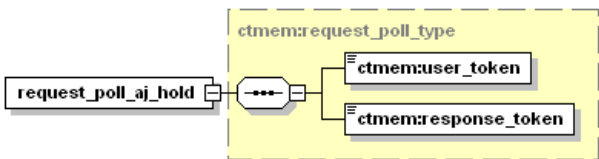


Table 99 request_poll aj_hold XML parameters description

Parameter	Description
user_token	Serial identification number supplied to the user during registration. String.
response_token	Limits the number of returned entities.

Polling response parameters

Figure 80 response_poll aj_hold XML parameters

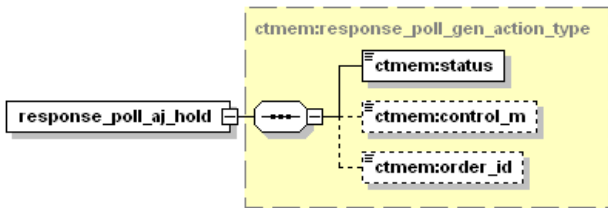


Table 100 response_poll aj_hold XML parameters description

Parameter	Description
status	Status of polling. String. Valid values: ■ OK ■ EXEC
control_m	CONTROL-M installation to which the job belongs. String. Mandatory.
order_id	Order ID of the job. String. Mandatory.

Fault response parameters

XML parameters for fault aj_hold and fault_poll aj_hold, as well as a sample fault response are described in “Fault Response” on page 213.

Free

Request parameters

Figure 81 request aj_free XML parameters

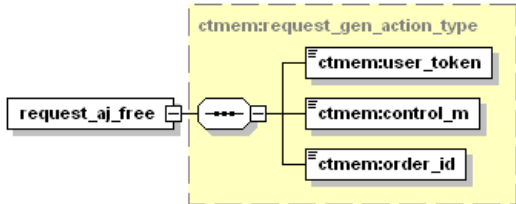


Table 101 request_aj_free XML parameters description

Parameter	Description
user_token	Serial identification number supplied to the user during registration. String.
control_m	CONTROL-M installation to which the job belongs. String. Mandatory.
order_id	Order ID of the job. String. Mandatory.

Response parameters

Figure 82 response_aj_free XML parameters

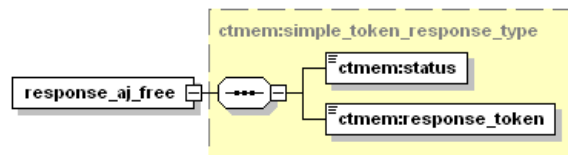


Table 102 response_aj_free XML parameters description

Parameter	Description
status	The status of the response.
response_token	Used in a polling request.

Polling request parameters

Figure 83 request_poll_aj_free XML parameters

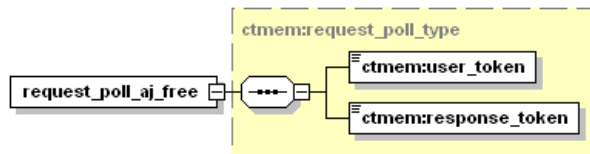


Table 103 request_poll_aj_free XML parameters description

Parameter	Description
user_token	Serial identification number supplied to the user during registration. String.
response_token	Limits the number of returned entities.

Polling response parameters

Figure 84 response_poll_aj_free XML parameters

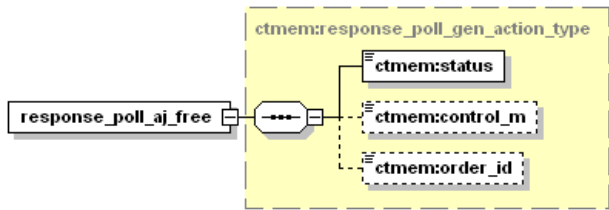


Table 104 response_poll_aj_free XML parameters description

Parameter	Description
status	Status of polling. String. Valid values: <ul style="list-style-type: none">■ OK■ EXEC
control_m	CONTROL-M installation to which the job belongs. String. Mandatory.
order_id	Order ID of the job. String. Mandatory.

Fault response parameters

XML parameters for fault_aj_free and fault_poll_aj_free, as well as a sample fault response are described in “Fault Response” on page 213.

Confirm

Request parameters

Figure 85 request_aj_confirm XML parameters

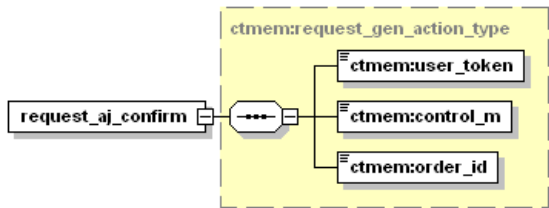


Table 105 request_aj_confirm XML parameters description

Parameter	Description
user_token	Serial identification number supplied to the user during registration. String.
control_m	CONTROL-M installation to which the job belongs. String. Mandatory.
order_id	Order ID of the job. String. Mandatory.

Response parameters

Figure 86 response_aj_confirm XML parameters

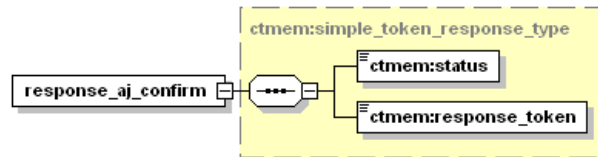


Table 106 response_aj_confirm XML parameters description

Parameter	Description
status	The status of the response.
response_token	Used in a polling request.

Polling request parameters

Figure 87 request_poll_aj_confirm XML parameters

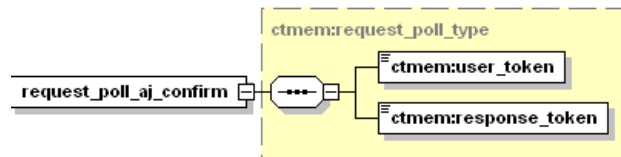


Table 107 request_poll_aj_confirm XML parameters description

Parameter	Description
user_token	Serial identification number supplied to the user during registration. String.
response_token	Limits the number of returned entities.

Polling response parameters

Figure 88 response_poll_aj_confirm XML parameters

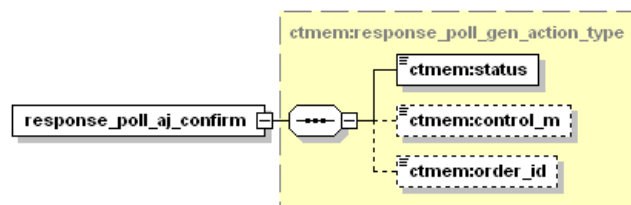


Table 108 response_poll aj_confirm XML parameters description

Parameter	Description
status	Status of polling. String. Valid values: <ul style="list-style-type: none">■ OK■ EXEC
control_m	CONTROL-M installation to which the job belongs. String. Mandatory.
order_id	Order ID of the job. String. Mandatory.

Fault response parameters

XML parameters for fault aj_confirm and fault_poll aj_confirm, as well as a sample fault response are described in “Fault Response” on page 213.

Rerun

Request parameters

Figure 89 request aj_rerun XML parameters

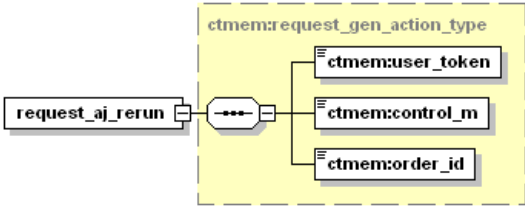


Table 109 request aj_rerun XML parameters description

Parameter	Description
user_token	Serial identification number supplied to the user during registration. String.
control_m	CONTROL-M installation to which the job belongs. String. Mandatory.
order_id	Order ID of the job. String. Mandatory.

Response parameters

Figure 90 response aj_rerun XML parameters

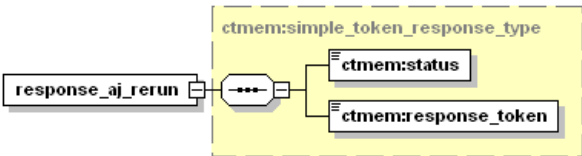


Table 110 response aj rerun XML parameters description

Parameter	Description
status	The status of the response.
response_token	Used in a polling request.

Polling request parameters

Figure 91 request_poll aj rerun XML parameters

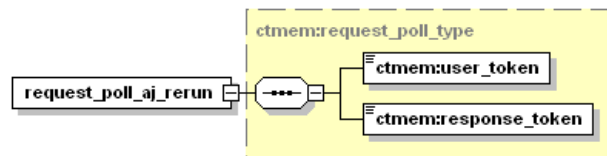


Table 111 request_poll aj rerun XML parameters description

Parameter	Description
user_token	Serial identification number supplied to the user during registration. String.
response_token	Limits the number of returned entities.

Polling response parameters

Figure 92 response_poll aj rerun XML parameters

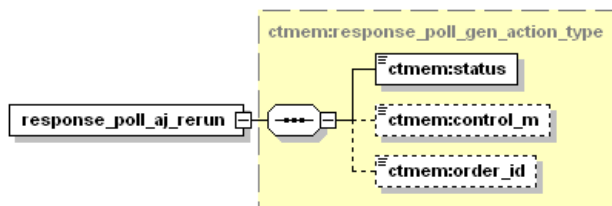


Table 112 response_poll aj rerun XML parameters description

Parameter	Description
status	Status of polling. String. Valid values: <ul style="list-style-type: none"> ■ OK ■ EXEC
control_m	CONTROL-M installation to which the job belongs. String. Mandatory.
order_id	Order ID of the job. String. Mandatory.

Fault response parameters

XML parameters for fault aj rerun and fault_poll aj rerun, as well as a sample fault response are described in [“Fault Response” on page 213](#).

Kill

Request parameters

Figure 93 request_aj_kill XML parameters

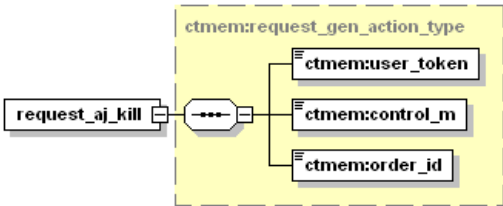


Table 113 request_aj_kill XML parameters description

Parameter	Description
user_token	Serial identification number supplied to the user during registration. String.
control_m	CONTROL-M installation to which the job belongs. String. Mandatory.
order_id	Order ID of the job. String. Mandatory.

Response parameters

Figure 94 response_aj_kill XML parameters

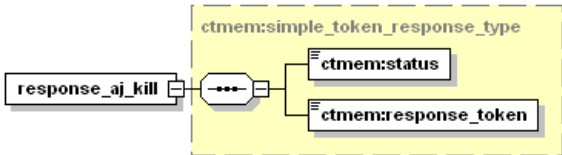


Table 114 response_aj_kill XML parameters description

Parameter	Description
status	The status of the response.
response_token	Used in a polling request.

Polling request parameters

Figure 95 request_poll_aj_kill XML parameters

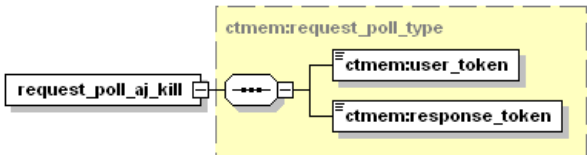


Table 115 request_poll_aj_kill XML parameters description

Parameter	Description
user_token	Serial identification number supplied to the user during registration. String.
response_token	Limits the number of returned entities.

Polling response parameters

Figure 96 response_poll_aj_kill XML parameters

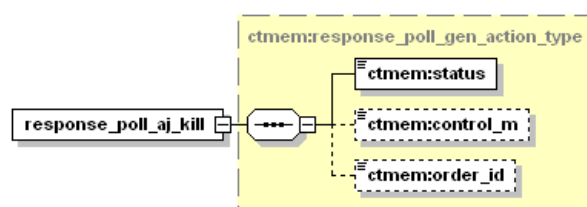


Table 116 response_poll_aj_kill XML parameters description

Parameter	Description
status	Status of polling. String. Valid values: <ul style="list-style-type: none"> ■ OK ■ EXEC
control_m	CONTROL-M installation to which the job belongs. String. Mandatory.
order_id	Order ID of the job. String. Mandatory.

Fault response parameters

XML parameters for fault_aj_kill and fault_poll_aj_kill, as well as a sample fault response are described in “[Fault Response](#)” on page 213.

Force OK

Request parameters

Figure 97 request_aj_force_ok XML parameters

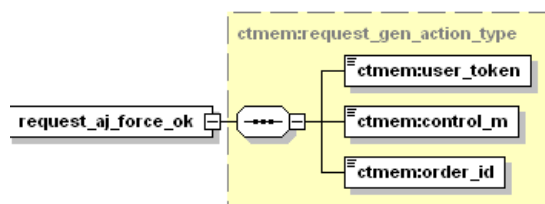


Table 117 request_aj_force_ok XML parameters description

Parameter	Description
user_token	Serial identification number supplied to the user during registration. String.
control_m	CONTROL-M installation to which the job belongs. String. Mandatory.
order_id	Order ID of the job. String. Mandatory.

Response parameters

Figure 98 response_aj_force_ok XML parameters

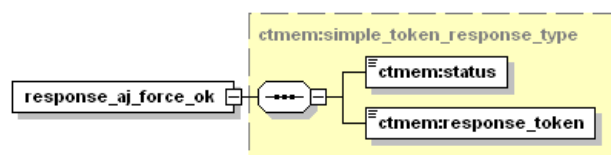


Table 118 response_aj_force_ok XML parameters description

Parameter	Description
status	The status of the response.
response_token	Used in a polling request.

Polling request parameters

Figure 99 request_poll_aj_force_ok XML parameters

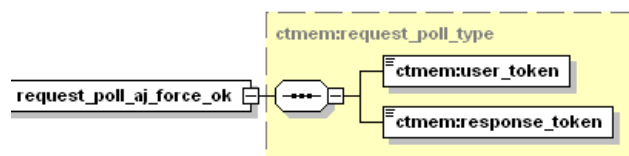


Table 119 request_poll_aj_force_ok XML parameters description

Parameter	Description
user_token	Serial identification number supplied to the user during registration. String.
response_token	Limits the number of returned entities.

Polling response parameters

Figure 100 response_poll_aj_force_ok XML parameters

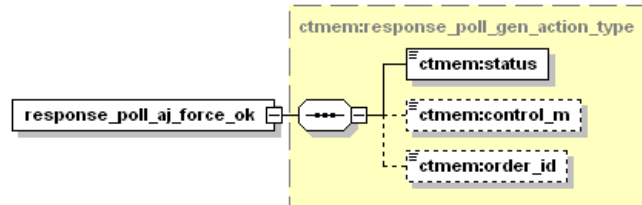


Table 120 response_poll_aj_force_ok XML parameters description

Parameter	Description
status	Status of polling. String. Valid values: <ul style="list-style-type: none"> ■ OK ■ EXEC
control_m	CONTROL-M installation to which the job belongs. String. Mandatory.
order_id	Order ID of the job. String. Mandatory.

Fault response parameters

XML parameters for fault_aj_force_ok and fault_poll_aj_force_ok, as well as a sample fault response are described in [“Fault Response”](#) on page 213.

Errors

See [“Job actions request errors \(Major code 450\)”](#) on page 263.

Examples

Successful example

Request

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body>
<ctmem:request_aj_free xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
  <ctmem:user_token>12345630</ctmem:user_token>
  <ctmem:control_m>ctm630</ctmem:control_m>
  <ctmem:order_id>0023e</ctmem:order_id>
</ctmem:request_aj_free>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body>
  <ctmem:response_aj_free xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
    <ctmem:status>OK</ctmem:status>
    <ctmem:response_token>103</ctmem:response_token>
  </ctmem:response_aj_free>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Polling request

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body>
<ctmem:request_polling_aj_free xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
  <ctmem:user_token>12345630</ctmem:user_token>
  <ctmem:response_token>103</ctmem:response_token>
</ctmem:request_polling_aj_free>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Polling response

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:response_polling_aj_free xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:status>OK</ctmem:status>
      <ctmem:control_m>ctm630</ctmem:control_m>
      <ctmem:order_id>0023e</ctmem:order_id>
    </ctmem:response_polling_aj_free>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Failure example

Request

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_aj_hold xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>12345630</ctmem:user_token>
      <ctmem:control_m>ctm630</ctmem:control_m>
      <ctmem:order_id>unknown</ctmem:order_id>
    </ctmem:request_aj_hold>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

```
<?xml version="1.0" encoding="iso-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Server</faultcode>
      <faultstring>Error response from EM Server.</faultstring>
      <detail>
        <ctmem:fault_aj_hold xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
          <ctmem:error_list ctmem:highest_severity="Error">
            <ctmem:error ctmem:major="401" ctmem:minor="4" ctmem:severity="Error">
              <ctmem:error_message>Internal Error: Order ID not Found...
            </ctmem:error_message>
            </ctmem:error>
          </ctmem:error_list>
        </ctmem:fault_aj_hold>
      </detail>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Job tracking

Tracks the progress of existing jobs in the CONTROL-M installation.

Request parameters

Figure 101 request_job_track XML parameters

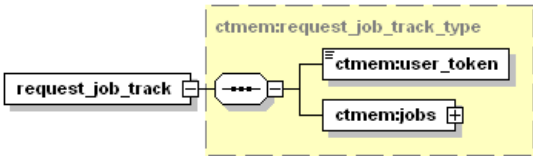


Table 121 request_job_track XML parameters Description

Parameter	Description
user_token	Serial identification number supplied to the user during registration. String.
jobs	A sequence of job. See Table 122.

Figure 102 job XML parameters

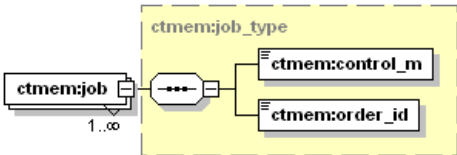


Table 122 job XML parameters description

Parameter	Description
control_m	CONTROL-M installation to which the job belongs. String. Mandatory.
order_id	Order ID of the job. String. Mandatory.

Response parameters

Figure 103 response_job_track XML parameters

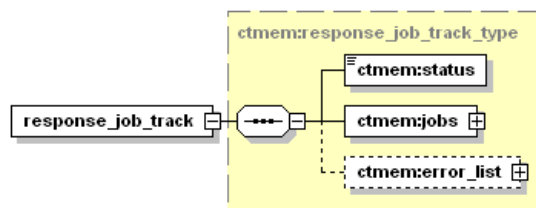


Table 123 response_job_track XML parameters description

Parameter	Description	Valid values
status	Describes the condition of the element that contains it. String. Note: status is a descriptive element in the response and job elements.	
jobs	A sequence of job . See Table 124 .	
error_list	A sequence of error . See Table 142 .	

Figure 104 job XML parameters

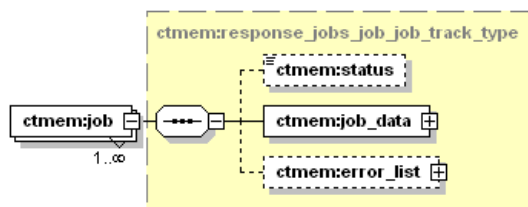


Table 124 job XML parameters description

Parameter	Description	
status	Describes the condition of the element that contains it. (for example, Error). String.	
job_data	An element that contains other parameters that describe the job. A sequence of job_data . See Table 125 .	
error_list	A sequence of error . See Table 142 .	
error_list attribute:	highest_severity	Indicates the severity level of the most critical error included in the error list. If only one error is included, the severity for that error is displayed. String.

Figure 105 job_data XML parameters

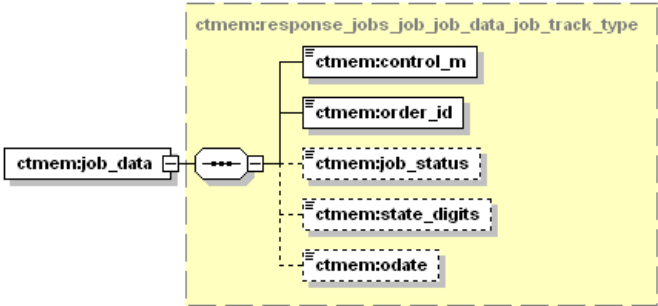


Table 125 job_data XML parameters description

Parameter	Description
control_m	String.
order_id	Serial number assigned to the job by the CONTROL-M installation. String.
job_status	Execution status of the job. String. Valid values are: <ul style="list-style-type: none">■ Ended OK■ Ended not OK■ Executing■ Wait Condition■ Wait Resource■ Wait User■ Not in AJF■ Unknown
state_digits	Serial number identifying the current job state.
odate	Original Scheduling date of the job. String.

Fault response parameters

Figure 106 fault_job_track

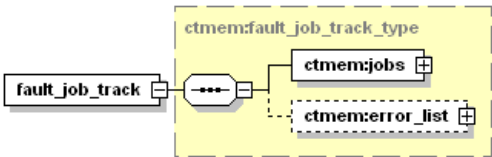


Table 126 fault_job_track XML parameters description

Parameter	Description
jobs	A sequence of job.

Table 126 fault_job_track XML parameters description

Parameter	Description	
error_list	A sequence of error . See Table 142 .	
error_list attribute:	highest_severity	Indicates the severity level of the most critical error included in the error list. If only one error is included, the severity for that error is displayed. String.

Figure 107 job XML parameters

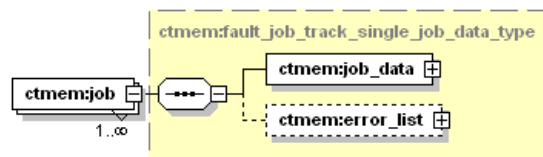


Table 127 job XML parameters description

Parameter	Description	
job_data	A sequence of job. For more information, refer to	
error_list	A sequence of error . See Table 142 .	
error_list attribute:	highest_severity	Indicates the severity level of the most critical error included in the error list. If only one error is included, the severity for that error is displayed. String.

Errors

See “Job tracking request errors (Major code 406)” on page 260.

Examples

Successful Example (tracking single job)

Request

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_job_track xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>12345630</ctmem:user_token>
      <ctmem:jobs>
        <ctmem:job>
          <ctmem:control_m>ctm620</ctmem:control_m>
          <ctmem:order_id>013ic</ctmem:order_id>
        </ctmem:job>
      </ctmem:jobs>
    </ctmem:request_job_track>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:response_job_track xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:status>OK</ctmem:status>
      <ctmem:jobs>
        <ctmem:job>
          <ctmem:status>OK</ctmem:status>
          <ctmem:job_data>
            <ctmem:control_m>ctm620</ctmem:control_m>
            <ctmem:order_id>013ic</ctmem:order_id>
            <ctmem:job_status>Ended OK</ctmem:job_status>
            <ctmem:state_digits>000030000000</ctmem:state_digits>
            <ctmem:odate>050711</ctmem:odate>
          </ctmem:job_data>
        </ctmem:job>
      </ctmem:jobs>
    </ctmem:response_job_track>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Failure Example (tracking single job)

Request

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_job_track xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>12345630</ctmem:user_token>
      <ctmem:jobs>
        <ctmem:job>
          <ctmem:control_m>ctm620</ctmem:control_m>
          <ctmem:order_id>013xc</ctmem:order_id>
        </ctmem:job>
      </ctmem:jobs>
    </ctmem:request_job_track>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP:Server</faultcode>
      <faultstring>Error response from EM Server</faultstring>
      <detail>
        <ctmem:fault_job_track xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
          <ctmem:jobs>
            <ctmem:job>
              <ctmem:job_data>
                <ctmem:control_m>ctm620</ctmem:control_m>
                <ctmem:order_id>013xc</ctmem:order_id>
              </ctmem:job_data>
              <ctmem:error_list ctmem:highest_severity="Error">
                <ctmem:error ctmem:major="406" ctmem:minor="1" ctmem:severity="Error">
                  <ctmem:error_message>
                    Job was not found in the last AJF.
                  </ctmem:error_message>
                </ctmem:error>
              </ctmem:error_list>
            </ctmem:job>
          </ctmem:jobs>
          <ctmem:error_list ctmem:highest_severity="Error">
            <ctmem:error ctmem:major="401" ctmem:minor="5" ctmem:severity="Error">
              <ctmem:error_message>Errors in request.</ctmem:error_message>
            </ctmem:error>
          </ctmem:error_list>
        </ctmem:fault_job_track>
      </detail>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Successful Example (tracking multiple jobs)

Request

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_job_track xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>12345630</ctmem:user_token>
      <ctmem:jobs>
        <ctmem:job>
          <ctmem:control_m>ctm620</ctmem:control_m>
          <ctmem:order_id>013ic</ctmem:order_id>
        </ctmem:job>
        <ctmem:job>
          <ctmem:control_m>ctm613</ctmem:control_m>
          <ctmem:order_id>006b2</ctmem:order_id>
        </ctmem:job>
        <ctmem:job>
          <ctmem:control_m>ctm620</ctmem:control_m>
          <ctmem:order_id>013xc</ctmem:order_id>
        </ctmem:job>
      </ctmem:jobs>
    </ctmem:request_job_track>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:response_job_track xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:status>OK</ctmem:status>
      <ctmem:jobs>
        <ctmem:job>
          <ctmem:status>OK</ctmem:status>
          <ctmem:job_data>
            <ctmem:control_m>ctm620</ctmem:control_m>
            <ctmem:order_id>013ic</ctmem:order_id>
            <ctmem:job_status>Ended OK</ctmem:job_status>
            <ctmem:state_digits>000030000000</ctmem:state_digits>
            <ctmem:odate>050711</ctmem:odate>
          </ctmem:job_data>
        </ctmem:job>
        <ctmem:job>
          <ctmem:status>OK</ctmem:status>
          <ctmem:job_data>
            <ctmem:control_m>ctm613</ctmem:control_m>
            <ctmem:order_id>006b2</ctmem:order_id>
            <ctmem:job_status>Wait Resource</ctmem:job_status>
            <ctmem:state_digits>220000004000</ctmem:state_digits>
            <ctmem:odate>050704</ctmem:odate>
          </ctmem:job_data>
        </ctmem:job>
        <ctmem:job>
          <ctmem:status>OK</ctmem:status>
          <ctmem:job_data>
            <ctmem:control_m>ctm620</ctmem:control_m>
            <ctmem:order_id>012qh</ctmem:order_id>
            <ctmem:job_status>Ended OK</ctmem:job_status>
            <ctmem:state_digits>000030000000</ctmem:state_digits>
            <ctmem:odate>050711</ctmem:odate>
          </ctmem:job_data>
        </ctmem:job>
      </ctmem:jobs>
    </ctmem:response_job_track>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Failure Example (tracking multiple jobs)

Request

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_job_track xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>12345630</ctmem:user_token>
      <ctmem:jobs>
        <ctmem:job>
          <ctmem:control_m>ctm620</ctmem:control_m>
          <ctmem:order_id>013ic</ctmem:order_id>
        </ctmem:job>
        <ctmem:job>
          <ctmem:control_m>ctm613</ctmem:control_m>
          <ctmem:order_id>006b2</ctmem:order_id>
        </ctmem:job>
        <ctmem:job>
          <ctmem:control_m>ctm620</ctmem:control_m>
          <ctmem:order_id>013xc</ctmem:order_id>
        </ctmem:job>
      </ctmem:jobs>
    </ctmem:request_job_track>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```


Response

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:response_job_track xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:status>PARTIAL_SUCCESS</ctmem:status>
      <ctmem:jobs>
        <ctmem:job>
          <ctmem:status>OK</ctmem:status>
          <ctmem:job_data>
            <ctmem:control_m>ctm620</ctmem:control_m>
            <ctmem:order_id>013ic</ctmem:order_id>
            <ctmem:job_status>Ended OK</ctmem:job_status>
            <ctmem:state_digits>000030000000</ctmem:state_digits>
            <ctmem:odate>050711</ctmem:odate>
          </ctmem:job_data>
        </ctmem:job>
        <ctmem:job>
          <ctmem:status>OK</ctmem:status>
          <ctmem:job_data>
            <ctmem:control_m>ctm613</ctmem:control_m>
            <ctmem:order_id>006b2</ctmem:order_id>
            <ctmem:job_status>Wait Resource</ctmem:job_status>
            <ctmem:state_digits>220000004000</ctmem:state_digits>
            <ctmem:odate>050704</ctmem:odate>
          </ctmem:job_data>
        </ctmem:job>
        <ctmem:job>
          <ctmem:job_data>
            <ctmem:control_m>ctm620</ctmem:control_m>
            <ctmem:order_id>013xc</ctmem:order_id>
          </ctmem:job_data>
          <ctmem:error_list ctmem:highest_severity="Error">
            <ctmem:error ctmem:major="406" ctmem:minor="1" ctmem:severity="Error">
              <ctmem:error_message>
                Job was not found in the last AJF.
              </ctmem:error_message>
            </ctmem:error>
          </ctmem:error_list>
        </ctmem:job>
      </ctmem:jobs>
    </ctmem:response_job_track>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Retrieve jobs in active environment

Retrieves jobs in the active environment.

- Jobs are searched in the "All Jobs" collection in GUI Server memory. If the "PinAllJobsCollection" system parameter is turned on, the "All Jobs" collection is loaded into memory at the start up of the GUI Server. If the "PinAllJobsCollection" is turned off, the request causes the "All Jobs" collection to be loaded into GUI Server memory, which may affect the performance and response time.
- The EMAPIActiveJobsLoadLimit system parameter controls the number of jobs in the active environment that are checked by the GUI Server when processing the request_act_retrieve_jobs request, and included in the request response.

It is possible to limit the number of returned jobs on the client side by specifying the max_returned_nodes parameter in the request. The value of the max_returned_nodes parameter should not exceed the value of the EMAPIActiveJobsLoadLimit system parameter.

If the value of the max_returned_nodes parameter exceeds the value of the EMAPIActiveJobsLoadLimit system parameter, a fault response is returned.

For more information about the PinAllJobsCollection and EMAPIActiveJobsLoadLimit system parameters, refer to the chapter discussing system parameters in the *CONTROL-M Administrator Guide*.

Request parameters

Figure 108 request_act_retrieve_jobs XML Parameters

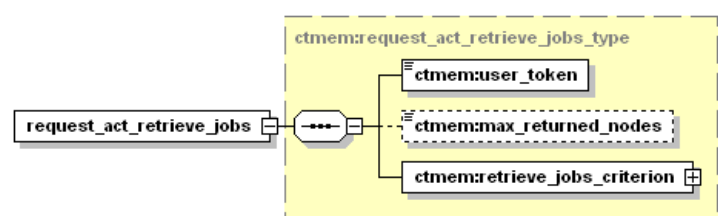


Table 128 request_act_retrieve_jobs XML Parameters Description(part 1 of 2)

Parameter	Description
user_token	Serial identification number supplied to the user during registration. String.

Table 128 request_act_retrieve_jobs XML Parameters Description(part 2 of 2)

Parameter	Description
max_returned_nodes	Limits the number of returned entities. Optional. Note: Should not exceed the value of the EMAPIActiveJobsLoadLimit system parameter.
retrieve_jobs_criterion	Retrieve jobs criteria wrapper. String. Consists of include and exclude filters that allow specifying items to include with or exclude from the retrieve criteria, as described in Table 129 .

Figure 109 retrieve_jobs_criterion

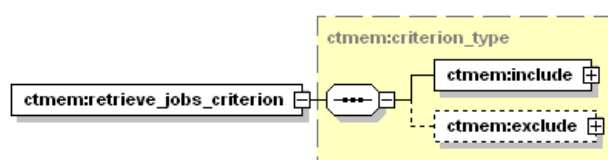


Table 129 retrieve_jobs_criterion XML parameters description

Parameter	Description
include	Mandatory. Include filter definitions criteria wrapper. For more information, refer Table 130 .
exclude	Optional. Exclude filter definitions criteria wrapper. For more information, refer Table 130 .

Figure 110 include and exclude XML parameters

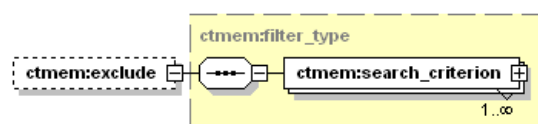


Table 130 include and exclude XML parameters description

Parameter	Description
search_criterion	String. Search criteria wrapper that consists of a sequence of param elements. At least one search_criterion element must appear under the include element. The amount of search_criterion elements is unbounded. The relationship between search_criterion elements in one filter is OR. For a description of param, refer to Table 131 .

Figure 111 search_criterion XML parameter

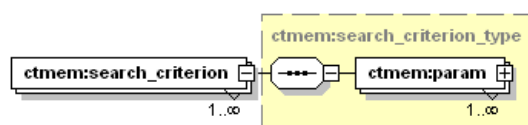


Table 131 search_criterion XML parameter description

Parameter	Description
param	Search criteria parameter wrapper. String. Parameters used to build the search criteria. At least one param element should appear under a search_criterion element. The amount of param elements is unbounded. The relationship between param elements in the same search_criterion is AND.

Figure 112 param XML parameters

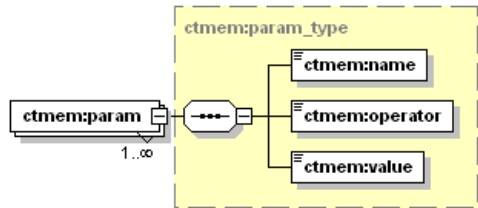


Table 132 param XML parameters description

Parameter	Description
name	Mandatory. Name of the retrieve active job parameter used as a search criteria. String. For a list of valid values for the name parameter, refer to Table 133 .
operator	Operator used in search criteria. String. Valid values: <ul style="list-style-type: none">■ EQ■ NE■ LT■ GT■ LIKE
value	Value used in search criteria. Any valid value of a job parameter. Wildcards and search patterns can be used in combination with LIKE operator. String. Mandatory.

Table 133 Valid name parameter values(part 1 of 3)

Parameter	Description
ORDER_ID	Order ID of the job to be retrieved. String. Mandatory
DATA_CENTER	The data center to which the job belongs.
APPLICATION	Name of the application to which the job's group belongs.
APPL_TYPE	The external application on which the job runs.
GROUP_NAME	Name of the group to which the job belongs.
MEMNAME	Name of the file that contains the job script.
JOB_NAME	Name of the job.

Table 133 Valid name parameter values(part 2 of 3)

Parameter	Description
TASK_TYPE	Type of the job (task) to be performed by CONTROL-M. Valid values: Microsoft Windows and UNIX <ul style="list-style-type: none"> ■ Job ■ Command ■ Dummy ■ Detached ■ External ■ Scheduling Group CONTROL-M for z/OS <ul style="list-style-type: none"> ■ Job ■ Started Task ■ Scheduling Group ■ Cyclic Job ■ Emergency Job ■ Emergency Cyclic Job ■ Cyclic Task ■ Emergency Task ■ Emergency Cyclic Task
CRITICAL	When selected, resources for the job are reserved exclusively for that job as they become available. When all necessary resources are available, the job is executed.
CYCLIC	If selected, indicates that the current job is cyclic (it should be rerun at specified intervals).
Emergency	CONTROL-M for z/OS: Indicates that the current job or started task is an emergency job or started task.
Part_of_BIM_service	Indicates if the job is included in a Business Service.
STATUS	The job execution status.
Ended	Job ended. Valid values are: <ul style="list-style-type: none"> ■ True ■ False
Ended_Not_OK	Job ended unsuccessfully. Valid values are: <ul style="list-style-type: none"> ■ True ■ False
Ended_OK	Job ended successfully. Valid values are: <ul style="list-style-type: none"> ■ True ■ False
Late	Job ended late. Valid values are: <ul style="list-style-type: none"> ■ True ■ False
Held	Job was held. Valid values are: <ul style="list-style-type: none"> ■ True ■ False

Table 133 Valid name parameter values(part 3 of 3)

Parameter	Description
DELETE_FLAG	Job was deleted. Valid values are: <ul style="list-style-type: none"> ■ True ■ False
Requested	Job was requested. Valid values are: <ul style="list-style-type: none"> ■ True ■ False
OWNER	Owner (user ID) on whose behalf the job is executed. This parameter is used by the CONTROL-M security mechanism.
NODEGROUP	Name of the node or node group on which following iterations of a job is run. This parameter does not apply to MVS jobs.
DESCRIPTION	A description of the job.
ODATE	The order date of the job.
AVG_RUNTIME	The average runtime of the job.
START_TIME	The start time of the job.
END_TIME	The end time of the job.
Incond Name	The name of the in-condition.
Incond Date	The date of the in-condition.
Outcond Name	The name of the out-condition.
Outcond Date	The date of the out-condition.
Quant Res	The name of the quantitative resource.
Control Res	The name of the control resource.
RBA	Relative block address (RBA). String.

Response parameters

Figure 113 response_act_retrieve_jobs XML parameters

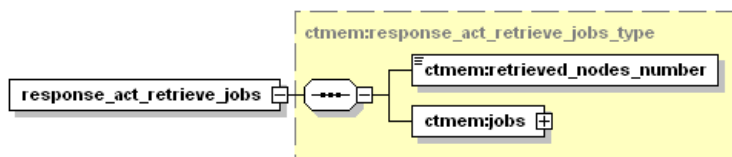


Table 134 response_act_retrieve_jobs XML parameters description

Parameter	Description
retrieved_nodes_number	Number of retrieved nodes.
jobs	A sequence of job_data. For more information, refer to Table 135 .

Table 135 job_data XML Parameters Description(part 1 of 3)

Parameter	Description
order_id	Order ID of the job. String. Mandatory.
control_m	CONTROL-M installation to which the job belongs. String. Mandatory.
application	Name of the application to which the job's group belongs.
application_type	Indicates the external application on which the job to be retrieved runs.
group	The group to which the retrieved job belongs.
job_name	The name of the job.
mem_name	The name of the file that contains the job script.
mem_lib	The name of the path that contains the retrieved job file.
order_table	Default or dummy Scheduling table to which you indicate the job belongs. A Scheduling table is not necessary because jobs that are created with the CONTROL-M/EM Web Services API are inserted directly into the Active Jobs file. However, you may want to include a value for this parameter so that the job can be tracked during statistical analysis that uses Scheduling table as a criterion.
order_library	Default or dummy Scheduling table library in which Scheduling table documentation is said to be stored. A Scheduling table (and, by extension, a Scheduling table library) are not necessary because jobs that are created with the CONTROL-M/EM Web Services API are inserted directly into the Active Jobs file. However, you may want to include a value for this parameter so that the job can be tracked during statistical analysis that uses Scheduling table or Scheduling Table Library as criteria. This parameter is specified only for z/OS jobs for which the order_table element was also specified.
owner	Owner (user ID) on whose behalf the job is executed. This parameter is used by the CONTROL-M security mechanism.
description	A description of the job.

Table 135 job_data XML Parameters Description(part 2 of 3)

Parameter	Description
task_type	Type of the job (task) to be performed by CONTROL-M. Valid values: Microsoft Windows and UNIX <ul style="list-style-type: none"> ■ Job ■ Command ■ Dummy ■ Detached ■ External ■ Scheduling Group CONTROL-M for z/OS <ul style="list-style-type: none"> ■ Job ■ Started Task ■ Scheduling Group ■ Cyclic Job ■ Emergency Job ■ Emergency Cyclic Job ■ Cyclic Task ■ Emergency Task ■ Emergency Cyclic Task
time_zone	Indicates the time zone according to which the job was scheduled.
in_BIM_service	Indicates if the job is included in Business Service.
job_status	Name of the job. String. Valid values are: <ul style="list-style-type: none"> ■ Ended OK ■ Ended not OK ■ Executing ■ Wait Condition ■ Wait Resource ■ Wait User ■ Not in AJF ■ Unknown
job_state	Indicates the job state, such as Held, Deleted, or Restarted.
state_digits	Serial number identifying the current job state.
odate	Original scheduling date of a job.
otime	Order time of the job. String. This parameter only applies to jobs run on CONTROL-M/Server for Distributed Systems 6.3.0x or CONTROL-M for z/OS 6.2.xx and above.
next_time	Indicates the next expected submission time for the job. For reruns or cyclic jobs that use the Interval option. For CONTROL-M for z/OS jobs, only the time can be specified. For all other jobs, the time and the date can be specified.
rerun_counter	Number of times the job has been rerun.

Table 135 job_data XML Parameters Description(part 3 of 3)

Parameter	Description
average_runtime	Average time (in minutes) for the job to run. The field displays the runtime statistic generated by the latest run of the CTMJSA utility (which generates a statistic based on the last successful runs of the job).
start_time	Date and time the job began executing.
end_time	Date and time the job finished executing.
critical	When selected, resources for the job are reserved exclusively for that job as they become available. When all necessary resources are available, the job is executed.
cyclic	If selected, indicates that the current job is cyclic (it should be rerun at specified intervals).
emergency	CONTROL-M for z/OS only: Indicates that the current job or started task is an emergency job or started task.
rba	Relative block address (RBA). String.
group_rba	RBA of job's scheduling group entity.

Fault response parameters

XML parameters for `fault_act_retrieve_jobs`, as well as a sample fault response are described in [“Fault Response” on page 213](#).

Errors

See [“Retrieve active jobs request errors \(Major code 440\)” on page 263](#).

Examples

Successful example

Request

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_act_retrieve_jobs xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>12345630</ctmem:user_token>
      <ctmem:max_returned_nodes>1000</ctmem:max_returned_nodes>
      <ctmem:retrieve_jobs_criterion>
        <ctmem:include>
          <ctmem:search_criterion>
            <ctmem:param>
              <ctmem:name>JOB_NAME</ctmem:name>
              <ctmem:operator>LIKE</ctmem:operator>
              <ctmem:val ue>*</ctmem:val ue>
            </ctmem:param>
            <ctmem:param>
              <ctmem:name>ODATE</ctmem:name>
              <ctmem:operator>EQ</ctmem:operator>
              <ctmem:val ue>060625</ctmem:val ue>
            </ctmem:param>
          </ctmem:search_criterion>
        </ctmem:include>
        <ctmem:exclude>
          <ctmem:search_criterion>
            <ctmem:param>
              <ctmem:name>DATA_CENTER</ctmem:name>
              <ctmem:operator>NE</ctmem:operator>
              <ctmem:val ue>ctm630</ctmem:val ue>
            </ctmem:param>
          </ctmem:search_criterion>
          <ctmem:search_criterion>
            <ctmem:param>
              <ctmem:name>APPLICATION</ctmem:name>
              <ctmem:operator>EQ</ctmem:operator>
              <ctmem:val ue>AJMN</ctmem:val ue>
            </ctmem:param>
          </ctmem:search_criterion>
          <ctmem:search_criterion>
            <ctmem:param>
              <ctmem:name>APPLICATION</ctmem:name>
              <ctmem:operator>EQ</ctmem:operator>
              <ctmem:val ue>Full Jobs</ctmem:val ue>
            </ctmem:param>
          </ctmem:search_criterion>
        </ctmem:exclude>
      </ctmem:retrieve_jobs_criterion>
    </ctmem:request_act_retrieve_jobs>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

```
<?xml version="1.0" encoding="iso-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:response_act_retrieve_jobs xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:retrieved_nodes_number>3
      </ctmem:retrieved_nodes_number>
      <ctmem:jobs>
        <ctmem:job>
          <ctmem:job_data>
            <ctmem:control_m>ctm630</ctmem:control_m>
            <ctmem:order_id>0023e</ctmem:order_id>
            <ctmem:rba>00023e</ctmem:rba>
            <ctmem:group_rba>000000</ctmem:group_rba>
            <ctmem:application>MYAPP</ctmem:application>
            <ctmem:application_type>OS</ctmem:application_type>
            <ctmem:group>MYGROUP1</ctmem:group>
            <ctmem:job_name>MYJOB</ctmem:job_name>
            <ctmem:mem_name> </ctmem:mem_name>
            <ctmem:mem_lib> </ctmem:mem_lib>
            <ctmem:order_table> </ctmem:order_table>
            <ctmem:owner>control m</ctmem:owner>
            <ctmem:description> </ctmem:description>
            <ctmem:task_type>Command</ctmem:task_type>
            <ctmem:time_zone> </ctmem:time_zone>
            <ctmem:in_BIM_service>0</ctmem:in_BIM_service>
            <ctmem:job_status>Wait Resource </ctmem:job_status>
            <ctmem:job_state> </ctmem:job_state>
            <ctmem:state_digits>020000004000</ctmem:state_digits>
            <ctmem:odate>060625</ctmem:odate>
            <ctmem:otime>20060625115417</ctmem:otime>
            <ctmem:next_time> </ctmem:next_time>
            <ctmem:rerun_counter>00000</ctmem:rerun_counter>
            <ctmem:average_runtime>000000</ctmem:average_runtime>
            <ctmem:start_time> </ctmem:start_time>
            <ctmem:end_time> </ctmem:end_time>
            <ctmem:critical>0</ctmem:critical>
            <ctmem:cyclic>False</ctmem:cyclic>
            <ctmem:emergency>False</ctmem:emergency>
          </ctmem:job_data>
        </ctmem:job>
        <ctmem:job>
          <ctmem:job_data>
            <ctmem:control_m>ctm630</ctmem:control_m>
            <ctmem:order_id>0023d</ctmem:order_id>
            <ctmem:rba>00023d</ctmem:rba>
            <ctmem:group_rba>000000</ctmem:group_rba>
            <ctmem:application>api WinApp2</ctmem:application>
            <ctmem:application_type> </ctmem:application_type>
            <ctmem:group>api Group2</ctmem:group>
            <ctmem:job_name>api test2</ctmem:job_name>
            <ctmem:mem_name>api MemName2</ctmem:mem_name>
            <ctmem:mem_lib>d: &lt; OWNER> control m</ctmem:mem_lib>
            <ctmem:order_table>api test</ctmem:order_table>
            <ctmem:owner> </ctmem:owner>
            <ctmem:description> </ctmem:description>
            <ctmem:task_type>Job </ctmem:task_type>
            <ctmem:time_zone> </ctmem:time_zone>
            <ctmem:in_BIM_service>0 </ctmem:in_BIM_service>
            <ctmem:job_status>Wait Resource </ctmem:job_status>
            <ctmem:job_state> </ctmem:job_state>
            <ctmem:state_digits>020000004000</ctmem:state_digits>
            <ctmem:odate>060625</ctmem:odate>
            <ctmem:otime>20060625113440</ctmem:otime>
            <ctmem:next_time> </ctmem:next_time>
            <ctmem:rerun_counter>00000</ctmem:rerun_counter>
            <ctmem:average_runtime>000000</ctmem:average_runtime>
            <ctmem:start_time> </ctmem:start_time>
            <ctmem:end_time> </ctmem:end_time>
            <ctmem:critical>0</ctmem:critical>
            <ctmem:cyclic>False</ctmem:cyclic>
            <ctmem:emergency>False</ctmem:emergency>
          </ctmem:job_data>
        </ctmem:job>
      </ctmem:jobs>
    </ctmem:response_act_retrieve_jobs>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```

</ctmem:job>
<ctmem:job>
  <ctmem:job_data>
    <ctmem:control_m>ctm630</ctmem:control_m>
    <ctmem:order_id>0023c</ctmem:order_id>
    <ctmem:rba>00023c</ctmem:rba>
    <ctmem:group_rba>000000</ctmem:group_rba>
    <ctmem:application>apiWinApp1</ctmem:application>
    <ctmem:application_type></ctmem:application_type>
    <ctmem:group>apiGroup1</ctmem:group>
    <ctmem:job_name>api test1</ctmem:job_name>
    <ctmem:mem_name>api MemName1</ctmem:mem_name>
    <ctmem:mem_lib></ctmem:mem_lib>
    <ctmem:order_table>api test</ctmem:order_table>
    <ctmem:owner>controlm</ctmem:owner>
    <ctmem:description></ctmem:description>
    <ctmem:task_type>Dummy</ctmem:task_type>
    <ctmem:time_zone></ctmem:time_zone>
    <ctmem:in_BIM_service>0</ctmem:in_BIM_service>
    <ctmem:job_status>Ended OK</ctmem:job_status>
    <ctmem:job_state>Run (2)</ctmem:job_state>
    <ctmem:state_digits>000030000000</ctmem:state_digits>
    <ctmem:odate>060625</ctmem:odate>
    <ctmem:otime>20060625113439</ctmem:otime>
    <ctmem:next_time></ctmem:next_time>
    <ctmem:rerun_counter>00001</ctmem:rerun_counter>
    <ctmem:average_runtime>000000</ctmem:average_runtime>
    <ctmem:start_time>20060625113441</ctmem:start_time>
    <ctmem:end_time>20060625113441</ctmem:end_time>
    <ctmem:critical>0</ctmem:critical>
    <ctmem:cyclic>False</ctmem:cyclic>
    <ctmem:emergency>False</ctmem:emergency>
  </ctmem:job_data>
</ctmem:job>
</ctmem:jobs>
</ctmem:response_act_retrieve_jobs>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Failure example

Request

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body>
  <ctmem:request_act_retrieve_jobs xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
    <ctmem:user_token>12345630</ctmem:user_token>
    <ctmem:max_returned_nodes>1000</ctmem:max_returned_nodes>
    <ctmem:retrieve_jobs_criterion>
      <ctmem:include>
        <ctmem:search_criterion>
          <ctmem:param>
            <ctmem:name>JOB_NAME</ctmem:name>
            <ctmem:operator>LIKE</ctmem:operator>
            <ctmem:val ue>*</ctmem:val ue>
          </ctmem:param>
        </ctmem:search_criterion>
      </ctmem:include>
    </ctmem:retrieve_jobs_criterion>
  </ctmem:request_act_retrieve_jobs>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Response

```
<?xml version="1.0" encoding="iso-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Server</faultcode>
      <faultstring>Error response from EM Server.</faultstring>
      <detail>
        <ctmem:fault_act_retrieve_jobs xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
          <ctmem:error_list ctmem:highest_severity='Error' >
            <ctmem:error ctmem:major='440' ctmem:minor='4' ctmem:severity='Error' >
              <ctmem:error_message>
                Partial result.
              </ctmem:error_message>
            </ctmem:error>
          </ctmem:error_list>
        </ctmem:fault_act_retrieve_jobs>
      </detail>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Change alert status

Changes the status (for example, from **not_noticed** to **handled**) of an alert.

Request parameters

Figure 114 request_change_alert_status XML parameters

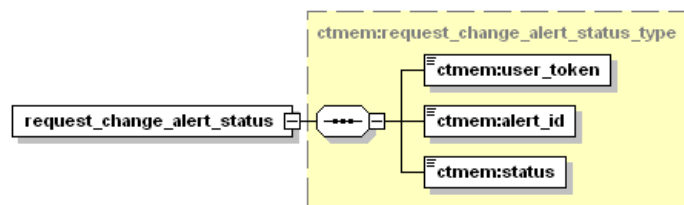


Table 136 request_change_alert_status XML parameters description

Parameter	Description	Valid Values
alert_id	The ID number of the alert. String. For more information on monitoring and handling alerts, see the <i>CONTROL-M User Guide</i> .	
status	Required status for the specified alert.	Valid values: <ul style="list-style-type: none">■ notice■ unnotice■ handle■ unhandle

Response parameters

Figure 115 response_change_alert_status XML parameters

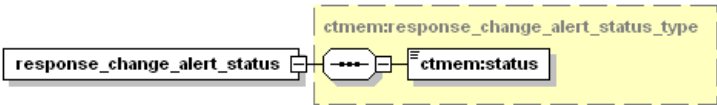


Table 137 response_change_alert_status XML parameters description

Parameter	Description
status	Description of message content (for example, Error). String.

Fault response parameters

XML parameters for the fault_change_alert_status, as well as a sample fault response are described in “[Fault Response](#)” on page 213.

Errors

See “[Alerts request errors \(Major code 408\)](#)” on page 261.

Examples

Successful Example

Request

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" >
  <SOAP-ENV:Body>
    <ctmem:request_change_alert_status xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>12345630</ctmem:user_token>
      <ctmem:alert_id>359</ctmem:alert_id>
      <ctmem:status>notice</ctmem:status>
    </ctmem:request_change_alert_status>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" >
  <SOAP-ENV:Body>
    <ctmem:response_change_alert_status xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:status>OK</ctmem:status>
    </ctmem:response_change_alert_status>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Failure Example 1

Request

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" >
  <SOAP-ENV:Body>
    <ctmem:request_change_alert_status xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>12345630</ctmem:user_token>
      <ctmem:alert_id>9879</ctmem:alert_id>
      <ctmem:status>notice</ctmem:status>
    </ctmem:request_change_alert_status>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP:Server</faultcode>
      <faultstring>Error response from EM Server</faultstring>
      <detail>
        <ctmem:fault_change_alert_status xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
          <ctmem:error_list ctmem:highest_severity="Error">
            <ctmem:error ctmem:major="408" ctmem:minor="1" ctmem:severity="Error">
              <ctmem:error_message>
                Alert id is not valid.
              </ctmem:error_message>
            </ctmem:error>
          </ctmem:error_list>
        </ctmem:fault_change_alert_status>
      </detail>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Failure Example 2

Request

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_change_alert_status xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>12345630</ctmem:user_token>
      <ctmem:alert_id>359</ctmem:alert_id>
      <ctmem:status>notice</ctmem:status>
    </ctmem:request_change_alert_status>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```


Response

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP:Server</faultcode>
      <faultstring>Error response from EM Server</faultstring>
      <detail>
        <ctmem: fault_change_alert_status xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
          <ctmem: error_list ctmem: highest_severity="Warning">
            <ctmem: error ctmem: major="408" ctmem: minor="7" ctmem: severity="Information">
              <ctmem: error_message>
                Alert already in desired state.
              </ctmem: error_message>
            </ctmem: error>
          </ctmem: error_list>
        </ctmem: fault_change_alert_status>
      </detail>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Retrieve BIM Services list

Retrieves the list of services active in the Batch Impact Manager Server.

Request parameters

Figure 116 request_get_bim_services_info XML parameters

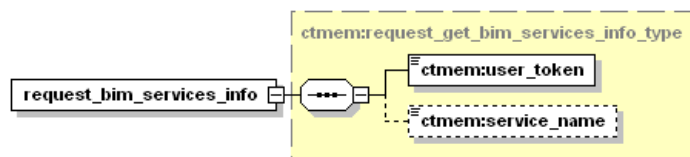


Table 138 request_get_bim_services_info XML parameters description

Parameter	Description
user_token	Serial identification number supplied to the user during registration. String.
service_name	The name of the service.

Response parameters

Figure 117 response_get_bim_services_info XML parameters

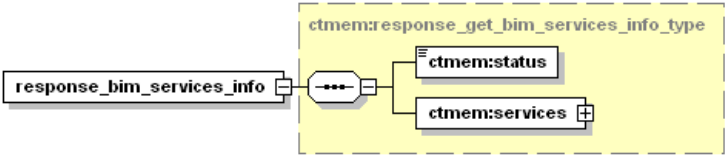


Table 139 response_get_bim_services_info XML parameters description

Parameter	Description
status	The status of the BIM service.
services	Information regarding the BIM service.

Figure 118 bim_services XML parameters

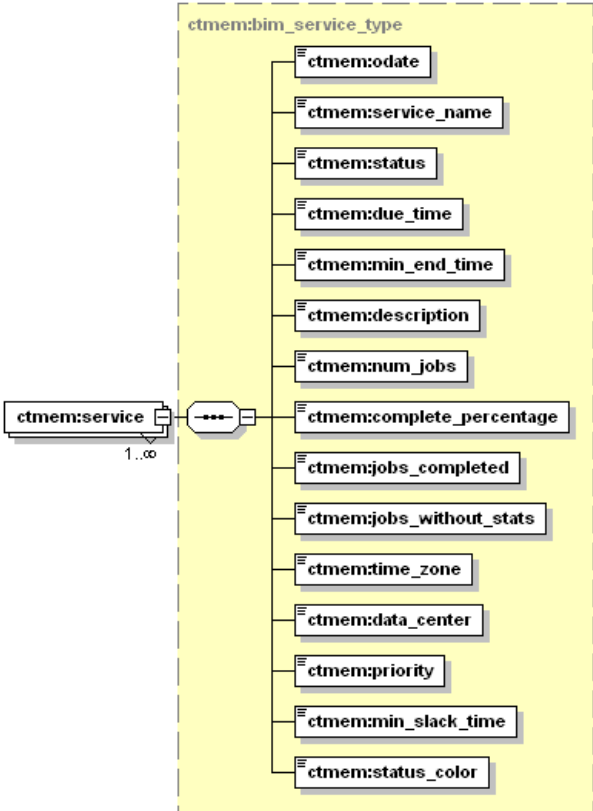


Table 140 bim_services XML parameters description

Parameter	Description
odate	The order date of the service.
service_name	The name of the service.
status	The status of the service.

Table 140 bim_services XML parameters description

Parameter	Description
due_time	The due time of the service.
min_end_time	The minimum time required for the service to complete.
description	The description of the service
num_jobs	The number of jobs contained in the service.
complete_percentage	The percentage of jobs completed.
jobs_completed	The number of jobs completed.
jobs_without_stats	The number of jobs without statistic information.
time_zone	Indicates the time zone according to which the job should be scheduled. When the value of this parameter is not zero, all dates reported use this time zone.
data_center	The data center where the service was ordered.
priority	The priority of the service.
min_slack_time	In case of a failure, this field indicates how long before the service is late.
status_color	Indicates the color of the status. Valid values are: <ul style="list-style-type: none"> ■ Service in process or completed ■ Job delay or Job finished early ■ Service delay or will not complete ■ Service completed late

Fault response parameters

XML parameters for the `get_bim_services_info` parameter, as well as a sample fault response are described in [“Fault Response” on page 213](#).

Examples

Successful example

Request

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" >
  <SOAP-ENV:Body>
    <ctmem:request_bim_services_info xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>12345630</ctmem:user_token>
    </ctmem:request_bim_services_info>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

```
<?xml version="1.0" encoding="iso-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:response_bim_services_info xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:status>OK</ctmem:status>
      <ctmem:services>
        <ctmem:service>
          <ctmem:odate>12: 26: 07 01/12/2006</ctmem:odate>
          <ctmem:service_name>Bim Service</ctmem:service_name>
          <ctmem:status>Service completed</ctmem:status>
          <ctmem:due_time>12: 29: 07 01/12/2006</ctmem:due_time>
          <ctmem:min_end_time>12: 26: 07 01/12/2006</ctmem:min_end_time>
          <ctmem:description/>
          <ctmem:num_jobs>3</ctmem:num_jobs>
          <ctmem:complete_percentage>100</ctmem:complete_percentage>
          <ctmem:jobs_completed>3</ctmem:jobs_completed>
          <ctmem:jobs_without_stats>2</ctmem:jobs_without_stats>
          <ctmem:time_zone> GMT +2</ctmem:time_zone>
          <ctmem:data_center>pal ace620</ctmem:data_center>
          <ctmem:priority>5</ctmem:priority>
          <ctmem:min_slack_time>0</ctmem:min_slack_time>
          <ctmem:status_color>Service in process or completed</ctmem:status_color>
        </ctmem:service>
        <ctmem:service>
          <ctmem:odate>12: 22: 00 01/12/2006</ctmem:odate>
          <ctmem:service_name>Bim Service</ctmem:service_name>
          <ctmem:status>Service is late</ctmem:status>
          <ctmem:due_time>12: 25: 00 01/12/2006</ctmem:due_time>
          <ctmem:min_end_time>13: 52: 08 01/12/2006</ctmem:min_end_time>
          <ctmem:description>Job "Service4" should have started by
            12: 25: 00 GMT+02: 00 and will not start on time. The reason
            is: Job is held.</ctmem:description>
          <ctmem:num_jobs>3</ctmem:num_jobs>
          <ctmem:complete_percentage>33</ctmem:complete_percentage>
          <ctmem:jobs_completed>1</ctmem:jobs_completed>
          <ctmem:jobs_without_stats>2</ctmem:jobs_without_stats>
          <ctmem:time_zone> GMT +2</ctmem:time_zone>
          <ctmem:data_center>pal ace620</ctmem:data_center>
          <ctmem:priority>5</ctmem:priority>
          <ctmem:min_slack_time>0</ctmem:min_slack_time>
          <ctmem:status_color>Service delay or will not complete</ctmem:status_color>
        </ctmem:service>
      </ctmem:services>
    </ctmem:response_bim_services_info>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Fault Response

This section shows the parameters for fault responses of all requests.

Table 141 fault XML parameter description

Parameter	Description
error_list	A sequence of error . For a list of the error_list sub-parameters, see Table 142 .

Table 142 error XML Parameters Description

Parameter	Description	
error attributes:	major	The error Major Code. An integer describing the family of errors to which the error belongs. For more information, see Appendix B, “Error codes and exceptions.”
	minor	The error minor Code. An integer that is a unique ID for an error of this type. For more information, see Appendix B, “Error codes and exceptions.”
	severity	The priority level assigned to the error.
error_message	Text description of the error. For more information, see Appendix B, “Error codes and exceptions.” String.	

Fault Example

```
<?xml version="1.0" encoding="iso-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Server</faultcode>
      <faultstring>Error response from EM Server.</faultstring>
      <detail>
        <ctmem:fault_def_create_jobs xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
          <ctmem:error_list ctmem:highest_severity='Error' >
            <ctmem:error ctmem:major='412' ctmem:minor='1' ctmem:severity='Error' >
              <ctmem:error_message>
                Create jobs definitions failed, invalid params.
              </ctmem:error_message>
            </ctmem:error>
            <ctmem:error ctmem:major='412' ctmem:minor='14' ctmem:severity='Error' >
              <ctmem:error_message>
                Create jobs definitions validation error: 'Field: AuthorError: The
                field must have a value'.
              </ctmem:error_message>
            </ctmem:error>
          </ctmem:error_list>
        </ctmem:fault_def_create_jobs>
      </detail>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Advanced features and optimization

This chapter presents the following topics:

Modifying initialization properties	215
Prototype 1	216
Prototype 2	216
Prototype 3	217
Getting and setting CONTROL-M/EM API properties	219
getProperties	219
setProperties	220
Polling interval timeout configuration	221
setPollRequestIntervalMilli	222
setPollRequestTimeoutMilli	222

Modifying initialization properties

Before using the CONTROL-M/EM API in your project, the API must be initialized (started). The API services are initialized using the **EMXMLInvoker** static method, `init`. This section shows you how to initialize the CONTROL-M/EM API using non-default CORBA ORB properties.

NOTE



If you intend to use the default initialization properties, you do not need to read “[Modifying initialization properties](#)”.

CORBA is an architecture and specification for creating, distributing, and managing distributed program objects in a network. In the CONTROL-M/EM API, CORBA is implemented through the use of JacORB, a third-party product that brokers requests among the components of distributed applications (acts as middleware).

The `CONTROL-M/EM API init` method initializes the CORBA Object Request Broker (ORB) in accordance with instructions that you (the programmer) supply in the `init` call.

- The default `init` prototype, Prototype 1, is called without the need to supply additional parameters or properties.
- The two additional `init` prototypes, Prototypes 2 and 3, enable you to specify alternate parameters and properties for initializing CORBA.

Prototype 1

```
public static void init ()
```

Prototype 1 (the default prototype) initializes the CORBA services using a `CONTROL-M/EM API`-specific CORBA configuration that was created during the API post-installation configuration (using **emapi-configure**).

The default `init` prototype is described in [“Initializing and stopping the `CONTROL-M/EM API` services” on page 61](#).

This CORBA configuration information is contained in the `jacorb.properties` file located under the `emapi-640\etc` directory.

Prototype 2

```
public static void init(String[] args)
```

This `init` prototype enables you to include an array of strings representing a list of arguments.

Arguments

The `args` parameter generally contains your command line arguments for the application’s `main`. This enables you to control ORB initialization from outside the program.

NOTE

For a list of CORBA parameters suitable for use in `args`, see the manufacturer’s documentation.



Code example

```
public class HelloWorld {
    public static void main(String[] args) {
        EMXMLInvoker.init(args);
        ...
    }
}
```

Run example

```
java HelloWorld -j acorb.impl name StandardNS
```

Prototype 3

```
public static void init(String[] args, Properties props)
```

This `init` prototype enables you to include an array of strings representing a list of arguments.

Arguments

The `args` parameter generally contains your command line arguments for the application's `main`. This enables you to control ORB initialization from outside the program.

NOTE



For a list of CORBA parameters suitable for use in `args`, see the manufacturer's documentation.

Properties

The `props` parameter (`Properties`) can contain CORBA parameters, using the same options as in the command line that was passed as the first parameter (`args`).

NOTE



The `Properties` class is part of the `java.util` package.

Code example

```
public class HelloWorld {
    public static void main(String[] args) {

        Properties props = new Properties();
        props.setProperty("jacob.ImplName", "StandardNS");

        EMXMLInvoker.init(args, props);
        ...
    }
}
```

Run example

```
j ava Hel l oWorl d
```



NOTE

This run example is used for illustrative purposes. To run it, you must add Java virtual machine (JVM) parameters and the CONTROL-M/EM API CLASSPATH. These concepts are discussed in [Chapter 2, "Installation."](#)

This CORBA configuration information is contained in the **jacorb.properties** file located in the **emapi-640\etc** directory.

By using both values for props and args, you can supply values for props that can be overridden by values that are supplied for args.

Getting and setting CONTROL-M/EM API properties

The CONTROL-M/EM API properties generally contain the host names of the CONTROL-M/EM GUI Server and the Global Alert Server (GAS).

Using the `getProperties` and `setProperties` methods, you can supply alternate parameters for these properties. These alternate parameters include, but are not limited to the host names of the CONTROL-M/EM GUI Server and the GAS.

getProperties

The `getProperties` method returns the properties object that you specified with the `setProperties` method.

If you have not set the CONTROL-M/EM API properties using `setProperties`, using `getProperties` returns the properties object containing the values in the `ctmem.properties` file under your *project\application* working directory.

Each call to `getProperties` reads the `ctmem.properties` file. Any modifications made to this file while the application is running affect subsequent calls to `getProperties`.

Prototype

```
public Properties getProperties();
```

The CONTROL-M/EM API default parameter properties are described in [Table 143](#). These parameters were specified in the `ctmemapi.properties` file during configuration of the CONTROL-M/EM API.

Table 143 CONTROL-M/EM API properties parameters

Parameter	Description
<code>com.bmc.ctmem.emapi.GSR.hostname</code>	Set to the hostname of the CONTROL-M/EM GUI Server.
<code>com.bmc.ctmem.emapi.GAS.hostname</code>	Set to the hostname of the Global Alerts Server.
<code>com.bmc.ctmem.emapi.XMLDATAPATH</code>	Set to the location of the XML schema.

setProperties

Use `setProperties` to modify the values of the CONTROL-M/EM API properties. All subsequent calls to `getProperties` return the `Properties` object that you specified with `setProperties`.

Prototype

```
public void setProperties(Properties props);
```

If the CONTROL-M/EM API receives `Properties` that do not contain the parameters described in [Table 143](#), the `localhost` value is used in their place. It is recommended that you supply parameter values for the CONTROL-M/EM servers that are being used.

Example

In this example, `setProperties` is called before the CONTROL-M/EM API is used. As a result, the CONTROL-M/EM API does not read the properties from the `ctmem.properties` file.

```
{
Properties props = new Properties();
props.setProperty("com.bmc.ctmem.emapi.GSR.hostname", "comp1");
props.setProperty("com.bmc.ctmem.emapi.GAS.hostname", "comp2");
EMXMLInvoker.setProperties(props);
...
// From this point forward, any new instance of EMXMLInvoker will
work with comp1 for the GUI Server and with comp2 for GAS
}
```

Polling interval timeout configuration

Requests that are sent using the **EMBasicXMLInvoker** class are polled automatically with default polling values. Poll requests are sent every five seconds until a reply is received.

It may be desirable to tailor these polling values to your network capabilities.

You can modify the number and frequency of poll requests that are sent using the `setPollRequestIntervalMilli` and `setPollRequestTimeoutMilli` methods.



NOTE

`setPollRequestIntervalMilli` and `setPollRequestTimeoutMilli` are methods of the **EMBasicXMLInvoker** class. They are not included in the **EMXMLInvoker** class.

If you do not intend to use the **EMBasicXMLInvoker** class, you do not need to read this section.

The total number of times that polling is conducted is a function of the values determined by the `setPollRequestIntervalMilli` and `setPollRequestTimeoutMilli` methods and the amount of time that each poll request takes.

Example

The total amount of time for polling is set at 10,000 milliseconds (10 seconds), using the `setPollRequestTimeoutMilli` method.

The time between poll requests is set at 2000 milliseconds (2 seconds), using the `setPollRequestIntervalMilli` method.

Each polling request takes about 500 milliseconds (0.5 seconds).

$$10,000 / (2000 + 500) = 4 \text{ poll requests}$$

setPollRequestIntervalMilli

The `setPollRequestIntervalMilli` method determines the time, in milliseconds, between poll requests. This time is measured from the **end** of the current poll request. The actual time that it takes to carry out the polling request is taken into account.

Prototype

```
public void setPollRequestIntervalMilli (final long timeout)
```

Default: **5000** milliseconds (5 seconds)

setPollRequestTimeoutMilli

The `setPollRequestTimeoutMilli` method determines the total time, in milliseconds, that is allotted for polling for a response to a request.

Prototype

```
public void setPollRequestTimeoutMilli (final long timeout)
```

Default: **-1** milliseconds. The **-1** value indicates that polling is carried on until a response is received. There is no timeout value.

Diagnostics and troubleshooting

This chapter presents the following topics:

CONTROL-M/EM API logging	223
Default logging behavior	224
Modifying logging behavior	224
Environment configuration troubleshooting.....	226
CLASSPATH: missing libraries or directories	226
Java virtual machine parameters.....	227
Incompatible object argument for a function call.....	229
Application runtime and communication troubleshooting.....	229
An exception is thrown by the invoke method.....	230
An error occurs when an XML file is submitted.....	231
Application cannot be started	231

This chapter helps you to analyze the CONTROL-M/EM API product's performance and prevent conflicts before they arise.

CONTROL-M/EM API logging

The CONTROL-M/EM API logs actions, exceptions, and warnings. The logging mechanism uses the Log4J library from the Jakarta Java Solutions project. The manifest file located at **emapi-version\classes\log4j-1.2.8.jar** indicates which version of Log4J is in use.

You can modify the logging procedures if you want to enable greater control over logging behavior. For more information, see [“Modifying logging behavior” on page 224](#) and the Log4J documentation.

Default logging behavior

The CONTROL-M/EM API logging default behavior has the following characteristics:

- The CONTROL-M/EM API logs only errors (priority level: ERROR).
- The maximum log file size is 50 KB.
- The CONTROL-M/EM API saves the last two completed logs.
- Log files are rolling, not cyclical.

Modifying logging behavior

Log default settings are coded into the CONTROL-M/EM API product. These settings can be overridden by values supplied from an outside source. You can supply additional parameters to extend logging capabilities. You can modify the logging procedure in either of two ways:

- Create a file called `emapi.log.cfg` in the working directory of your application or project. When this file is present, the CONTROL-M/EM API automatically passes the information contained in the file to Log4J. [Figure 119 on page 225](#) shows a sample configuration file.
- Use your own code to pass information to Log4J. The CONTROL-M/EM API uses the `com.bmc.ctmem.emapi` logging category to log data.

Logging parameters

Any logging configuration information that you enter in the `emapi_log.cfg` file or in your project code must conform to Log4J specifications (as described in the Log4J documentation).

Log message priorities

If you are modifying logging behavior, you need to indicate the priority of messages recorded in the log in your configuration file. [Table 144 on page 225](#) describes the message priority levels supported by Log4J. The table orders the levels from the most critical to the least critical.

NOTE

Each setting includes messages of that level plus the more critical levels.



Table 144 Priority levels for log messages

Level	Description
FATAL	displays fatal error messages
ERROR	displays messages for fatal and non-fatal errors
WARN	displays warning messages and all error messages
INFO	displays system information, warnings, and all errors
DEBUG	displays debugging information and all other priority messages

Sample log configuration file

Figure 119 shows a sample log configuration file that contains common attributes.

Figure 119 `emapi_log.cfg` file example with default parameters

```
log4j.rootLogger=ERROR
log4j.category.com.bmc.ctmem.emapi=ERROR, EMAPI_Appender

# set appender for EM/API
log4j.appender.EMAPI_Appender=org.apache.log4j.RollingFileAppender
log4j.appender.EMAPI_Appender.file=emapi_log_file
log4j.appender.EMAPI_Appender.append=true
log4j.appender.EMAPI_Appender.maxFileSize=10kb
log4j.appender.EMAPI_Appender.maxBackupIndex=2
log4j.appender.EMAPI_Appender.layout=org.apache.log4j.SimpleLayout
```

The first line in Figure 119 (`log4j.rootLogger=ERROR`) sets the priority of the root category to ERROR. If your project code uses the Log4J library, the root category determines the highest overall priority of log messages.

The second line defines an appender for a specified category. The category can have multiple appenders. The appender that is defined here is assigned a logging priority.

- The appender is `EMAPI_Appender`.
- The category is `com.bmc.ctmem.emapi`.
- The logging priority is `ERROR`.

The remaining lines of the example define the properties of the appender, `EMAPI_Appender`. The valid properties depend on the type of appender that is being defined.

This example illustrates a `RollingFileAppender` that has the properties discussed in Table 145 on page 226.

Table 145 RollingFileAppender example properties

Code	Description
file=emapi.log	indicates that the log output file is named emapi.log
append=true	indicates that new information is added to the end of the log file
maxFileSize=50kb	indicates that the maximum file of the log file is 50 KB
maxBackupIndex=2	indicates the number of backups made (that is, the number of old log files saved)
layout=org.apache.log4j.SimpleLayout	indicates the format for entries to the log file

Environment configuration troubleshooting

Modifications or omissions in the environment configuration can cause the CONTROL-M/EM API to malfunction.

This section discusses the following problems and their solutions:

- [“CLASSPATH: missing libraries or directories”](#)
- [“Java virtual machine parameters” on page 227](#)
- [“Incompatible object argument for a function call” on page 229](#)

CLASSPATH: missing libraries or directories

Problem

Null exception at some point during runtime.

Possible cause

Some CONTROL-M/EM API directories, **.jar** libraries, or library locations might be missing from your application class path (CLASSPATH).

Solution

Ensure that the following files and directories are in the class path:

- *fullPath/emapi-640/classes/log4j-1.2.8.jar*
- *fullPath/emapi-640/classes/emapi.jar*
- *fullPath/emapi-640/classes/jbcl.jar*
- *fullPath/emapi-640/classes/jacorb.jar*
- *fullPath/emapi-640/classes/concurrent-1.3.2.jar*
- *fullPath/emapi-640/classes/logkit-1.2.jar*
- *fullPath/emapi-640/classes/commons-codec-1.3.jar*
- *fullPath/emapi-640/classes/avalon-framework-4.1.5.jar*
- *fullPath/emapi-640/classes/antlr-2.7.2.jar*
- *fullPath/emapi-640/classes/xercesImpl.jar*
- *fullPath/emapi-640/classes/xml-apis.jar*
- *fullPath/emapi-640/etc*
- *fullPath/emapi-640/etc/jacorb.properties*

Java virtual machine parameters

Problem

Your application fails during **init** initialization or when the **invoke** method is first used.

Possible cause

There is an error in the virtual machine's configuration parameters.

Solution

Check the log file for a message similar to the following message:

```
ERROR - resolve naming service failed during initial reference:
org.omg.CORBA.COMM_FAILURE: minor code: 1398079490 completed: No
at com.sun.corba.se.internal.iiop.IIOPConnection.writeLock(Unknown Source)
```

If you received an **org.omg.CORBA.COMM_FAILURE** exception, and the exception was thrown from the **com.sun.corba.se.internal.iiop** package, you probably did not specify to the virtual machine that it must use the JacORB (CORBA) implementation (in place of the Sun default implementation).

You must specify to the Java virtual machine that it must use the CORBA implementation.

Use one of the following methods:

Method 1

Run the `java` command from the project working directory, using the following parameters:

```
j java -Dorg.omg.CORBA.ORBClass=org.jacorb.orb.ORB  
-Dorg.omg.CORBA.ORBSingletonClass=org.jacorb.orb.ORBSingleton projectAppName
```

projectAppName is the executable file of your project.

Method 2

1 Create a file containing the following text:

```
org.omg.CORBA.ORBClass=org.jacorb.orb.ORB  
org.omg.CORBA.ORBSingletonClass=org.jacorb.orb.ORBSingleton
```

2 Save the file with the name **orb.properties** in one of the following directories:

- If you are using JDK, use `JDK_HOME/jre/lib`
- If you are using JRE, use `JRE_HOME/lib`

Method 3

Call the **init** method by using the properties illustrated in the following code example (assuming that **args** is your application command-line arguments).

```
Properties props = new Properties();  
props.setProperty("org.omg.CORBA.ORBClass",  
"org.jacorb.orb.ORB");  
props.setProperty("org.omg.CORBA.ORBSingletonClass",  
"org.jacorb.orb.ORBSingleton");  
EMXMLInvoker.init(args, props);
```

Alternatively, call the **init** method without any parameters.

Incompatible object argument for a function call

Problem

Calls fail when using `EMBasicXMLInvoker`, but succeed when using `EMXMLInvoker`.

Possible cause

Irrelevant libraries in the application class path (`CLASSPATH`) conflict with libraries that are required by the CONTROL-M/EM API. For example, a newer or older version of the APACHE Xerces Parser might be found in the `CLASSPATH`.

Solution

Use one of the following solutions:

- Use the smallest set of classes possible when running the API. Doing so prevents your project from finding irrelevant products and libraries in the `CLASSPATH`.
- Ensure that the you have listed the required libraries first in the `CLASSPATH` so that they are found first.

Application runtime and communication troubleshooting

Communication with CONTROL-M/EM server components is essential to the operation of the CONTROL-M/EM API. A disruption in communication can cause an exception to be thrown.

This section discusses the following problems and their solutions:

- “An exception is thrown by the invoke method.” on page 230
- “An error occurs when an XML file is submitted.” on page 231
- “Application cannot be started” on page 231

An exception is thrown by the invoke method.

Problem

An exception is thrown when you use the invoke method.

Possible cause 1

The CONTROL-M/EM API cannot communicate with the CORBA Naming Service for one of the following reasons:

- The CORBA services are down.
- The Naming Service host name and port settings in the **jacorb.properties** file are incorrect. You can find the corbaloc reference by searching for the following string: **ORBInitRef.NameService**.

Solution

Ensure that the following requirements are met:

- The CORBA Naming Service is running (started) on the CORBA Server host computer.
- The Naming Service host name and port settings in the **jacorb.properties** file point to the computer that hosts the CORBA Server that you are using.

Possible cause 2

Communication cannot be established with the CONTROL-M/EM GUI Server or Global Alerts Server for one of the following reasons:

- Host names of the GUI Server or the Global Alerts Server are incorrect in the **ctmemapi.properties** file.
- Host names of the GUI Server or the Global Alerts Server were specified with incorrect values in the code.

Solution

Ensure that the following requirements are met:

- The CONTROL-M/EM GUI Server is running.

- The host names and ports of the CONTROL-M/EM server components (GUI Server and Global Alerts Server) are registered correctly in the CORBA Naming Service.
- If you are using more than one CORBA Server, you are currently using the CORBA Server that serves the CONTROL-M/EM GUI Server and Global Alerts Server to which you are sending CONTROL-M/EM API requests.

An error occurs when an XML file is submitted.

Problem

An error occurs when an XML file is submitted.

Possible cause

The XML file contains characters that the XML standard considers invalid. The following characters are translated in the XML file as indicated in the following table:

Character	Meaning	Translate to
<	less than	<
>	greater than	>
&	ampersand	&
“	quotation marks	"
‘	apostrophe	'
ASCII 10	line feed	

ASCII 13	carriage return	

Application cannot be started

Problem

Applications based on CONTROL-M/EM API cannot be started.

Possible cause

Because of Java limitations, applications that use the CONTROL-M/EM API cannot be started when the path of the directory in which the CONTROL-M/EM API is installed contains spaces or other special characters.

Solution 1

- 1 Move the CONTROL-M/EM API directory structure to a path that does not contain spaces or special characters.
- 2 Reconfigure the CONTROL-M/EM API.
- 3 If needed, modify the **JAVA_HOME** and CLASSPATH environment variables.

NOTE



JAVA_HOME refers to the directory where the Java 2 Runtime Environment (JRE) was installed. The Java 2 SDK (also called the JDK) contains the JRE, but at a different level in the file hierarchy. For example, if the Java 2 SDK or JRE was installed in **/home/user1**, **JAVA_HOME** would be as follows:

- If you are using JRE, **/home/user1/jre1.4.x**
- If you are using SDK, **/home/user1/jdk1.4.x/jre**

Solution 2: A workaround

This solution provides an alternate method (one for UNIX and one for Windows) for use with the CONTROL-M/EM API. If you implement this solution, use this method at all times.

UNIX

1. Define an alias to bypass the path that contains spaces. The alias should point to the CONTROL-M/EM API **emapi-640** root directory. By using this alias when calling the CONTROL-M/EM API, you can avoid referring to spaces and special characters that exist in the actual path.
2. After defining the alias, reconfigure the CONTROL-M/EM API so that it uses the new path.

Example

If the CONTROL-M/EM API root directory is **\$HOME/path with space/emapi-640**, use the following commands to create the alias.

```
cd $HOME
ln -s "path with space" pathwithoutspace
cd $HOME/pathwithoutspace/emapi-640
```

Reconfigure the CONTROL-M/EM API to reflect the new path.

Microsoft Windows

To avoid referring to spaces and special characters that exist in the actual path of the CONTROL-M/EM API, call the CONTROL-M/EM API by using the MS-DOS-compatible (short) 8.3-format version of its name. Ensure that you reconfigure the CONTROL-M/EM API so that it uses the 8.3-format version of its path.

Example

1. Open a new command prompt window.
2. Change the directory to the CONTROL-M/EM API root directory, referring to the directory by using the 8.3-format version of its name.

If the CONTROL-M/EM API root directory is
d:\Program Files\BMC Software\emapi-640, use the following command:

```
cd progra~1\bmcsof~1\emapi-640
```

3. Reconfigure the CONTROL-M/EM API to reflect the new path.

Uninstalling CONTROL-M/EM API

The CONTROL-M/EM API is a self-contained product provided in compressed formats (zip and tar) that can be extracted to any directory. This chapter explains how to remove previous versions of the CONTROL-M/EM API from your system.

To uninstall the CONTROL-M/EM API:



WARNING

Every CONTROL-M/EM installation has a directory named **emapi** under its home directory. This directory contains dictionary and XML schema files necessary for the functioning of server components. Do not delete this directory. If you remove this directory by mistake, restore it from a backup.

Delete the CONTROL-M/EM API directory:

- For version 6.3.01: **emapi-630**
- For version 6.2.01: **emapi-620**
- For version 6.1.03: **emapi-613**
- For versions earlier than 6.1.03: **emapi-610**

Request format examples

This appendix includes one or more examples for each of the more complex requests:

Add Condition or Delete Condition request	238
Example 1: Add a condition to a CONTROL-M	238
Example 2: Delete a condition from a CONTROL-M	238
Job Creation request	239
Example 1: Create a job requiring confirmation	239
Example 2: Create a cyclic Job	240
Example 3: Create a job including In and Out conditions	241
Example 4: Create a job that requires resources	242
Example 5: Create a job that includes On-Do statements	243
Example 6: Create a job that includes On-Do statements	244
Example 7: Create an active group scheduling table	245
Example 8: Create an active job in an existing group scheduling table	246
Order or Force request	247
Example 1: Order a UNIX job	247
Example 2: Force a UNIX job	247
Example 3: Force a UNIX job into a 'recent' group scheduling table	248
Example 4: Force a UNIX job into a 'recent' group scheduling table allowing duplication	249
Example 5: Force a scheduling table that contains a group scheduling table	250

Requests are submitted to the CONTROL-M/EM API as XML-formatted text strings. The text strings can be saved as plain-text files.

For instructions for how to create request strings and examples of the less complex requests, see [Chapter 7, "Request reference."](#)

Add Condition or Delete Condition request

Example 1: Add a condition to a CONTROL-M

An Add Condition request that adds the Cond01 prerequisite condition to the CTM01 CONTROL-M installation.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_add_condition xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>$USER_TOKEN$</ctmem:user_token>
      <ctmem:control_m>CTM01</ctmem:control_m>
      <ctmem:condition>
        <ctmem:name>Cond01</ctmem:name>
        <ctmem:odate>ODATE</ctmem:odate>
      </ctmem:condition>
    </ctmem:request_add_condition>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Example 2: Delete a condition from a CONTROL-M

A Delete Condition request that removes the Cond01 prerequisite condition from the CTM01 CONTROL-M installation.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_delete_condition xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>$USER_TOKEN$</ctmem:user_token>
      <ctmem:control_m>CTM01</ctmem:control_m>
      <ctmem:condition>
        <ctmem:name>Cond01</ctmem:name>
        <ctmem:odate>0101</ctmem:odate>
      </ctmem:condition>
    </ctmem:request_delete_condition>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Job Creation request

Example 1: Create a job requiring confirmation

A Job Creation request that creates a single job that performs a command and requires confirmation to run.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_create_job xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>$USER_TOKEN$</ctmem:user_token>
      <ctmem:control_m>ctm3-omega</ctmem:control_m>
      <ctmem:active_job>
        <ctmem:job_name>Conf0</ctmem:job_name>
        <ctmem:mem_name>Conf0</ctmem:mem_name>
        <ctmem:task_type>command</ctmem:task_type>
        <ctmem:application>Confirm</ctmem:application>
        <ctmem:group>Confirm</ctmem:group>
        <ctmem:confirm_flag>yes</ctmem:confirm_flag>
        <ctmem:max_wait>0</ctmem:max_wait>
        <ctmem:prevent_nct2>no</ctmem:prevent_nct2>
        <ctmem:time_from>0900</ctmem:time_from>
        <ctmem:time_until>1100</ctmem:time_until>
        <ctmem:critical>no</ctmem:critical>
        <ctmem:cyclic>no</ctmem:cyclic>
        <ctmem:auto_archive>no</ctmem:auto_archive>
        <ctmem:sys_db>no</ctmem:sys_db>
        <ctmem:arch_max_days>0</ctmem:arch_max_days>
        <ctmem:arch_max_runs>0</ctmem:arch_max_runs>
        <ctmem:rerun_max>0</ctmem:rerun_max>
        <ctmem:command>ls -l</ctmem:command>
      </ctmem:active_job>
    </ctmem:request_create_job>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Example 2: Create a cyclic Job

A Job Creation request that creates a single cyclic job that performs a command.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_create_job xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>$USER_TOKEN$</ctmem:user_token>
      <ctmem:control_m>ctm3-omega</ctmem:control_m>
      <ctmem:active_job>
        <ctmem:job_name>Cyc0</ctmem:job_name>
        <ctmem:mem_name>Cyc0</ctmem:mem_name>
        <ctmem:task_type>command</ctmem:task_type>
        <ctmem:application>Cycli c</ctmem:application>
        <ctmem:group>Cycli c</ctmem:group>
        <ctmem:confirm_flag>no</ctmem:confirm_flag>
        <ctmem:max_wait>0</ctmem:max_wait>
        <ctmem:prevent_nct2>no</ctmem:prevent_nct2>
        <ctmem:time_from>1100</ctmem:time_from>
        <ctmem:time_until>2300</ctmem:time_until>
        <ctmem:rerun_interval>00060M</ctmem:rerun_interval>
        <ctmem:critical>no</ctmem:critical>
        <ctmem:cyclic>yes</ctmem:cyclic>
        <ctmem:auto_archive>no</ctmem:auto_archive>
        <ctmem:sys_db>no</ctmem:sys_db>
        <ctmem:arch_max_days>0</ctmem:arch_max_days>
        <ctmem:arch_max_runs>0</ctmem:arch_max_runs>
        <ctmem:rerun_max>0</ctmem:rerun_max>
        <ctmem:command>ls -l</ctmem:command>
      </ctmem:active_job>
    </ctmem:request_create_job>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```


Example 3: Create a job including In and Out conditions

A Job Creation request that creates a single job including In and Out conditions.

```
<?xml version="1.0"?>
<!DOCTYPE ctmem: message_package SYSTEM "EMAPI_create_aj.dtd">
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV: Envelope xmlns: SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV: Body>
    <ctmem: request_create_aj xmlns: ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem: user_token>$USER_TOKEN$</ctmem: user_token>
      <ctmem: control_m>ctm3-omega</ctmem: control_m>
      <ctmem: active_job>
        <ctmem: job_name>InCond0</ctmem: job_name>
        <ctmem: mem_name>InCond0</ctmem: mem_name>
        <ctmem: task_type>command</ctmem: task_type>
        <ctmem: application>InCond</ctmem: application>
        <ctmem: group>InCond</ctmem: group>
        <ctmem: config_flag>no</ctmem: config_flag>
        <ctmem: max_wait>0</ctmem: max_wait>
        <ctmem: prevent_nct2>no</ctmem: prevent_nct2>
        <ctmem: time_from>0800</ctmem: time_from>
        <ctmem: critical>no</ctmem: critical>
        <ctmem: cyclic>no</ctmem: cyclic>
        <ctmem: auto_archive>no</ctmem: auto_archive>
        <ctmem: sys_db>no</ctmem: sys_db>
        <ctmem: arch_max_days>0</ctmem: arch_max_days>
        <ctmem: arch_max_runs>0</ctmem: arch_max_runs>
        <ctmem: rerun_max>0</ctmem: rerun_max>
        <ctmem: command>ls -l</ctmem: command>
        <ctmem: in_conditions>
          <ctmem: in_condition>
            <ctmem: condition>ctm600a0</ctmem: condition>
            <ctmem: date>ODAT</ctmem: date>
            <ctmem: and_or>and</ctmem: and_or>
          </ctmem: in_condition>
        </ctmem: in_conditions>
        <ctmem: out_conditions>
          <ctmem: out_condition>
            <ctmem: condition>ctm600a0</ctmem: condition>
            <ctmem: date>ODAT</ctmem: date>
            <ctmem: sign>delete</ctmem: sign>
          </ctmem: out_condition>
        </ctmem: out_conditions>
      </ctmem: active_job>
    </ctmem: request_create_aj>
  </SOAP-ENV: Body>
</SOAP-ENV: Envelope>
```

Example 4: Create a job that requires resources

A Job Creation request that creates a single job requiring Control and Quantitative resources.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_create_job xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>$USER_TOKEN$</ctmem:user_token>
      <ctmem:control_m>ctm3-omega</ctmem:control_m>
      <ctmem:active_job>
        <ctmem:job_name>Rsc0</ctmem:job_name>
        <ctmem:mem_name>Rsc0</ctmem:mem_name>
        <ctmem:task_type>dummy</ctmem:task_type>
        <ctmem:application>Resource</ctmem:application>
        <ctmem:group>Resource</ctmem:group>
        <ctmem:confirm_flag>no</ctmem:confirm_flag>
        <ctmem:max_wait>0</ctmem:max_wait>
        <ctmem:description>Job requiring resources.</ctmem:description>
        <ctmem:cyclic>no</ctmem:cyclic>
        <ctmem:sys_db>no</ctmem:sys_db>
        <ctmem:arch_max_days>0</ctmem:arch_max_days>
        <ctmem:arch_max_runs>0</ctmem:arch_max_runs>
        <ctmem:overlib>asds</ctmem:overlib>
        <ctmem:count_cyclic_from>end</ctmem:count_cyclic_from>
        <ctmem:control_resources>
          <ctmem:control_resource>
            <ctmem:resource>Disks</ctmem:resource>
            <ctmem:type>shared</ctmem:type>
          </ctmem:control_resource>
          <ctmem:control_resource>
            <ctmem:resource>Time</ctmem:resource>
            <ctmem:type>shared</ctmem:type>
          </ctmem:control_resource>
          <ctmem:control_resource>
            <ctmem:resource>JJ4</ctmem:resource>
            <ctmem:type>shared</ctmem:type>
          </ctmem:control_resource>
          <ctmem:control_resource>
            <ctmem:resource>DE1a34</ctmem:resource>
            <ctmem:type>exclusive</ctmem:type>
          </ctmem:control_resource>
          <ctmem:control_resource>
            <ctmem:resource>CPU</ctmem:resource>
            <ctmem:type>shared</ctmem:type>
          </ctmem:control_resource>
        </ctmem:control_resources>
      </ctmem:active_job>
    </ctmem:request_create_job>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```

        <ctmem: quanti tati ve_resources>
            <ctmem: quanti tati ve_resource>
                <ctmem: resource>DL</ctmem: resource>
                <ctmem: quanti ty>30</ctmem: quanti ty>
            </ctmem: quanti tati ve_resource>
        </ctmem: quanti tati ve_resources>
    </ctmem: acti ve_j ob>
</ctmem: request_create_aj >
</SOAP-ENV: Body>
</SOAP-ENV: Envel ope>

```

Example 5: Create a job that includes On-Do statements

A Job Creation request that creates a single job including multiple On-Do statements and an AutoEdit variable.

```

<?xml versi on="1. 0" encodi ng="I SO-8859-1" ?>
<SOAP-ENV: Envel ope xml ns: SOAP-ENV="http: //schemas. xml soap. org/soap/envel ope/" >
    <SOAP-ENV: Body>
        <ctmem: request_create_aj xml ns: ctmem="http: //www. bmc. com/ctmem/schema640" >
            <ctmem: user_token>$USER_TOKEN$</ctmem: user_token>
            <ctmem: control _m>ctm3-omega</ctmem: control _m>
            <ctmem: acti ve_j ob>
                <ctmem: j ob_name>DoSys0</ctmem: j ob_name>
                <ctmem: mem_name>DoSys0</ctmem: mem_name>
                <ctmem: task_type>command</ctmem: task_type>
                <ctmem: appl i cati on>OnDo</ctmem: appl i cati on>
                <ctmem: group>DoSys</ctmem: group>
                <ctmem: confi rm_fl ag>no</ctmem: confi rm_fl ag>
                <ctmem: max_wai t>0</ctmem: max_wai t>
                <ctmem: descri pti on>Job with On-Do statements. </ctmem: descri pti on>
                <ctmem: ti me_from>0900</ctmem: ti me_from>
                <ctmem: ti me_until >1100</ctmem: ti me_until >
                <ctmem: pri ori ty>Ab</ctmem: pri ori ty>
                <ctmem: cri ti cal >no</ctmem: cri ti cal >
                <ctmem: cycl i c>no</ctmem: cycl i c>
                <ctmem: auto_archi ve>no</ctmem: auto_archi ve>
                <ctmem: sys_db>no</ctmem: sys_db>
                <ctmem: arch_max_days>0</ctmem: arch_max_days>
                <ctmem: arch_max_runs>0</ctmem: arch_max_runs>
                <ctmem: rerun_max>0</ctmem: rerun_max>
                <ctmem: count_cycl i c_from>end</ctmem: count_cycl i c_from>
                <ctmem: command>I s -I </ctmem: command>
                <ctmem: autoedi t_assi gnments>
                    <ctmem: autoedi t_assi gnment>
                        <ctmem: name>COPYTO</ctmem: name>
                        <ctmem: val ue>%%HOME. %%FI LE</ctmem: val ue>
                    </ctmem: autoedi t_assi gnment>
                </ctmem: autoedi t_assi gnments>
                <ctmem: on_do_statements>
                    <ctmem: on_do_statement>

```

```
<ctmem: on_statements>
  <ctmem: on_statement>
    <ctmem: code>*</ctmem: code>
    <ctmem: statement>*</ctmem: statement>
  </ctmem: on_statement>
</ctmem: on_statements>
<ctmem: do_statements>
  <ctmem: do_sysout>
    <ctmem: option>copy</ctmem: option>
    <ctmem: parameter>%%COPYTO</ctmem: parameter>
  </ctmem: do_sysout>
</ctmem: do_statements>
</ctmem: on_do_statement>
</ctmem: on_do_statements>
</ctmem: active_job>
</ctmem: request_create_job>
</SOAP-ENV: Body>
</SOAP-ENV: Envelope>
```

Example 6: Create a job that includes On-Do statements

A Job Creation request that creates a single job including multiple On-Do statements.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV: Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" >
  <SOAP-ENV: Body>
    <ctmem: request_create_job xmlns:ctmem="http://www.bmc.com/ctmem/schema640" >
      <ctmem: user_token>$USER_TOKEN$</ctmem: user_token>
      <ctmem: control_m>ctm3-omega</ctmem: control_m>
      <ctmem: active_job>
        <ctmem: job_name>DoVar0</ctmem: job_name>
        <ctmem: mem_name>DoVar0</ctmem: mem_name>
        <ctmem: task_type>command</ctmem: task_type>
        <ctmem: application>OnDo</ctmem: application>
        <ctmem: group>DoSetVar</ctmem: group>
        <ctmem: confirm_flag>no</ctmem: confirm_flag>
        <ctmem: max_wait>0</ctmem: max_wait>
        <ctmem: description>Multiple On-Dos</ctmem: description>
        <ctmem: priority>Ab</ctmem: priority>
        <ctmem: critical>no</ctmem: critical>
        <ctmem: cyclic>no</ctmem: cyclic>
        <ctmem: auto_archive>no</ctmem: auto_archive>
        <ctmem: sys_db>no</ctmem: sys_db>
        <ctmem: arch_max_days>0</ctmem: arch_max_days>
        <ctmem: arch_max_runs>0</ctmem: arch_max_runs>
        <ctmem: rerun_max>0</ctmem: rerun_max>
        <ctmem: overlib>JOBDOC</ctmem: overlib>
        <ctmem: count_cyclic_from>end</ctmem: count_cyclic_from>
        <ctmem: command>echo %%ABC</ctmem: command>
      </ctmem: active_job>
    </ctmem: request_create_job>
  </SOAP-ENV: Body>
</SOAP-ENV: Envelope>
```

```

        <ctmem: on_do_statements>
          <ctmem: on_do_statement>
            <ctmem: on_statements>
              <ctmem: on_statement>
                <ctmem: code>*</ctmem: code>
                <ctmem: statement>*</ctmem: statement>
              </ctmem: on_statement>
            </ctmem: on_statements>
          <ctmem: do_statements>
            <ctmem: do_autoedit>
              <ctmem: name>Auto1</ctmem: name>
              <ctmem: value>1234567890</ctmem: value>
            </ctmem: do_autoedit>
          </ctmem: do_statements>
        </ctmem: on_do_statement>
      </ctmem: on_do_statements>
    </ctmem: active_job>
  </ctmem: request_create_adj>
</SOAP-ENV: Body>
</SOAP-ENV: Envelope>

```

Example 7: Create an active group scheduling table

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV: Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV: Body>
    <ctmem: request_create_adj xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem: user_token>$USER_TOKEN$</ctmem: user_token>
      <ctmem: control_m>omega</ctmem: control_m>
      <ctmem: active_job>
        <ctmem: task_type>scheduling_group</ctmem: task_type>
        <ctmem: application>appl</ctmem: application>
        <ctmem: group>grp1</ctmem: group>
        <ctmem: odate>ODAT</ctmem: odate>
      </ctmem: active_job>
    </ctmem: request_create_adj>
  </SOAP-ENV: Body>
</SOAP-ENV: Envelope>

```

Example 8: Create an active job in an existing group scheduling table

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_create_job xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>$USER_TOKEN$</ctmem:user_token>
      <ctmem:control_m>omega-ctm3</ctmem:control_m>
      <ctmem:active_job>
        <ctmem:job_name>pwJob3</ctmem:job_name>
        <ctmem:mem_name>ajJob3</ctmem:mem_name>
        <ctmem:owner>emuser</ctmem:owner>
        <ctmem:task_type>Command</ctmem:task_type>
        <ctmem:application>UniXAppl</ctmem:application>
        <ctmem:group>UniXJobs</ctmem:group>
        <ctmem:odate>ODAT</ctmem:odate>
        <ctmem:group_id>000115</ctmem:group_id>
        <ctmem:command>c:</ctmem:command>
      </ctmem:active_job>
    </ctmem:request_create_job>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

NOTE



In the group ID element, you need to specify:

- the RBA of the group scheduling table if the CONTROL-M is running on z/OS.
 - the Order ID of the group scheduling table (with the addition of a leading zero) if the CONTROL-M is running on another operating system.
-

Order or Force request

Example 1: Order a UNIX job

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_order_force xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>$USER_TOKEN$</ctmem:user_token>
      <ctmem:force_it>no</ctmem:force_it>
      <ctmem:control_m>UnixDc</ctmem:control_m><ctmem:job_id>2</ctmem:job_id>
      <ctmem:job_name>OrdSimJobU</ctmem:job_name>
      <ctmem:table_name>OrdSimJobU</ctmem:table_name>
      <ctmem:odate>20020522</ctmem:odate>
    </ctmem:request_order_force>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Example 2: Force a UNIX job

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_order_force xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>$USER_TOKEN$</ctmem:user_token>
      <ctmem:force_it>yes</ctmem:force_it>
      <ctmem:control_m>UnixDc</ctmem:control_m>
      <ctmem:job_id>2</ctmem:job_id>
      <ctmem:job_name>ForSimJobU</ctmem:job_name>
      <ctmem:table_name>ForSimJobU</ctmem:table_name>
      <ctmem:odate>20010522</ctmem:odate>
    </ctmem:request_order_force>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Example 3: Force a UNIX job into a 'recent' group scheduling table

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_order_force xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>${USER_TOKEN}</ctmem:user_token>
      <ctmem:force_it>yes</ctmem:force_it>
      <ctmem:control_m>UnixDc</ctmem:control_m>
      <ctmem:job_id>2</ctmem:job_id>
      <ctmem:job_name>Fol nRecent</ctmem:job_name>
      <ctmem:table_name>Fol nSGJobU</ctmem:table_name>
      <ctmem:odate>20010522</ctmem:odate>
      <ctmem:autoedit_assignments>
        <ctmem:autoedit_assignment>
          <ctmem:name>Recent</ctmem:name>
          <ctmem:value>1</ctmem:value>
        </ctmem:autoedit_assignment>
        <ctmem:autoedit_assignment>
          <ctmem:name>A</ctmem:name>
          <ctmem:value>1</ctmem:value>
        </ctmem:autoedit_assignment>
        <ctmem:autoedit_assignment>
          <ctmem:name>A</ctmem:name>
          <ctmem:value>2</ctmem:value>
        </ctmem:autoedit_assignment>
      </ctmem:autoedit_assignments>
      <ctmem:scheduling_group_info>
        <ctmem:into_group>recent</ctmem:into_group>
        <ctmem:allow_dup>no</ctmem:allow_dup>
      </ctmem:scheduling_group_info>
    </ctmem:request_order_force>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```


Example 4: Force a UNIX job into a 'recent' group scheduling table allowing duplication

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_order_force xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>$USER_TOKEN$</ctmem:user_token>
      <ctmem:force_it>yes</ctmem:force_it>
      <ctmem:control_m>UnixDc</ctmem:control_m>
      <ctmem:job_id>6</ctmem:job_id>
      <ctmem:job_name>Fol nRecDup</ctmem:job_name>
      <ctmem:table_name>Fol nSGJobU</ctmem:table_name>
      <ctmem:odate>20010522</ctmem:odate>
      <ctmem:autoedit_assignments>
        <ctmem:autoedit_assignment>
          <ctmem:name>RecentDup</ctmem:name>
          <ctmem:value>2</ctmem:value>
        </ctmem:autoedit_assignment>
        <ctmem:autoedit_assignment>
          <ctmem:name>A</ctmem:name>
          <ctmem:value>2</ctmem:value>
        </ctmem:autoedit_assignment>
        <ctmem:autoedit_assignment>
          <ctmem:name>B</ctmem:name>
          <ctmem:value>3</ctmem:value>
        </ctmem:autoedit_assignment>
      </ctmem:autoedit_assignments>
      <ctmem:scheduling_group_info>
        <ctmem:into_group>recent</ctmem:into_group>
        <ctmem:allow_dup>yes</ctmem:allow_dup>
      </ctmem:scheduling_group_info>
    </ctmem:request_order_force>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Example 5: Force a scheduling table that contains a group scheduling table

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <ctmem:request_order_force xmlns:ctmem="http://www.bmc.com/ctmem/schema640">
      <ctmem:user_token>$USER_TOKEN$</ctmem:user_token>
      <ctmem:force_it>yes</ctmem:force_it>
      <ctmem:control_m>UnixDc</ctmem:control_m>
      <ctmem:table_name>ForSGTblU</ctmem:table_name>
      <ctmem:odate>20010522</ctmem:odate>
      <ctmem:autoedit_assignments>
        <ctmem:autoedit_assignment>
          <ctmem:name>ForceUnixTblWithGST</ctmem:name>
          <ctmem:value>8</ctmem:value>
        </ctmem:autoedit_assignment>
        <ctmem:autoedit_assignment>
          <ctmem:name>A</ctmem:name>
          <ctmem:value>8</ctmem:value>
        </ctmem:autoedit_assignment>
        <ctmem:autoedit_assignment>
          <ctmem:name>H</ctmem:name>
          <ctmem:value>9</ctmem:value>
        </ctmem:autoedit_assignment>
      </ctmem:autoedit_assignments>
    </ctmem:request_order_force>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Error codes and exceptions

This appendix presents the following topics:

Severity	252
Error code reference.....	253
NULL exception errors (Major code 000).....	254
Low-level API exceptions (Major code 100)	254
Parser exceptions (Major code 200).....	255
CONTROL-M/Server errors: Group 1 (Major code 300).....	255
CONTROL-M/Server errors: Group 2 (Major code 301).....	257
CONTROL-M/Server errors: Group 3 (Major code 302).....	258
Generic request exceptions (Major code 401)	258
Poll request errors (Major code 403).....	259
Add or Delete Condition request errors (Major code 404)	259
Order or Force request errors (Major code 405)	259
Job tracking request errors (Major code 406).....	260
Authorization request errors (Major code 407).....	260
Alerts request errors (Major code 408)	261
Create active job request errors (Major code 409).....	261
Upload scheduling table request errors (Major code 411).....	261
Create job/scheduling group definitions request errors (Major code 412)	262
Delete job definitions request errors (Major code 413)	262
Retrieve active jobs request errors (Major code 440)	263
Job actions request errors (Major code 450).....	263
CONTROL-M/EM API Java client errors (Major code 500).....	264
Gateway messages (Major code 600)	265

Errors and exceptions may occur when the CONTROL-M/Enterprise Manager API is being used.

Errors and exceptions can be divided into the following groups:

- errors and exceptions that are caused by the CONTROL-M/EM API
- errors and exceptions that are caused by faulty formatting in the XML request file

- errors and exceptions that are caused by the presence of incorrect data in the XML request file
- errors and exceptions that are caused by CONTROL-M/EM or the CONTROL-M installation

NOTE



In addition to being described in this chapter, CONTROL-M/EM API errors are listed in the **EMAPIMessages.txt** file located in the home directory of the CONTROL-M/EM installation.

Each error is composed of the following information:

- **Major code**
Integer that represents a family of related errors or exceptions.
- **Minor code**
ID code unique to each error.
- **Severity**
Indicates how critical the error is. Severity also determines the way that the CONTROL-M/EM API handles the error.
- **Description**
Text description of the error, returned as a string when an exception is thrown.

Severity

Every error that the CONTROL-M/EM API displays has a severity level. This level indicates the priority of the error and indicates how much information is written to the log file.

When using the CONTROL-M/EM API, the most common severity level is **ERROR**. Using the default logging configuration, the **DEBUG** error is never used. For information on how to modify the logging configuration, see “[CONTROL-M/EM API logging](#)” on page 223.

The severity levels that are supported by the CONTROL-M/EM API are listed in [Table 146](#).

Table 146 Log message priority levels

Level	Description
FATAL	Displays fatal error messages. Warning: When a FATAL error is generated, it is recommended that you stop using the CONTROL-M/EM API immediately, and that you investigate the cause of the error. If you continue to use the CONTROL-M/EM API despite receiving a FATAL error, stop and restart the CONTROL-M/EM GUI Server and Global Alerts Server before performing any more API actions.
ERROR	Displays messages for fatal and non-fatal errors.
WARN	Displays warning messages and all error messages.
INFO	Displays system information, warnings, and all errors.
DEBUG	Displays debugging information and all other priority messages.

Error code reference

All CONTROL-M/EM API errors are arranged in logical groups. Each group is identified by a unique 3 - 5-digit Major Code. Each error in the group is identified by a unique 1 - 3-digit Minor Code.

[Table 147](#) lists all of the Major Code groups and includes a reference to the pages where the errors in each group are described.

Individual errors are described in tables arranged by Major Code. The Minor Code, severity, and description are listed for each error in the tables.

Table 147 Error and exception major codes(part 1 of 2)

Code	Title	For error descriptions, see:
000	NULL Exception	“NULL exception errors (Major code 000)” on page 254
100	Low-level API Exceptions	“Low-level API exceptions (Major code 100)” on page 254
200	Parser Exceptions	“Parser exceptions (Major code 200)” on page 255
300	CONTROL-M Server Errors	Note: Category contains no individual errors.
300	CONTROL-M Server Errors: Group 1	“CONTROL-M/Server errors: Group 1 (Major code 300)” on page 255
301	CONTROL-M Server Errors: Group 2	“CONTROL-M/Server errors: Group 2 (Major code 301)” on page 257
302	CONTROL-M Server Errors: Group 3	“CONTROL-M/Server errors: Group 3 (Major code 302)” on page 258
400	CONTROL-M/EM API Request Exceptions	Note: Category contains no individual errors.

Table 147 Error and exception major codes(part 2 of 2)

Code	Title	For error descriptions, see:
401	Generic Request Exceptions	"Generic request exceptions (Major code 401)" on page 258
403	Poll Request	"Poll request errors (Major code 403)" on page 259
404	Add/Delete Condition Request	"Add or Delete Condition request errors (Major code 404)" on page 259
405	Order/Force Request	"Order or Force request errors (Major code 405)" on page 259
406	Job Tracking Request	"Job tracking request errors (Major code 406)" on page 260
407	Authorization Request	"Authorization request errors (Major code 407)" on page 260
408	Alerts Request	"Alerts request errors (Major code 408)" on page 261
409	Create Active Job Request	"Create active job request errors (Major code 409)" on page 261
411	Upload scheduling tables	"Upload scheduling table request errors (Major code 411)" on page 261
412	Create jobs or Scheduling Group definitions	"Create job/scheduling group definitions request errors (Major code 412)" on page 262
413	Delete job definitions	"Delete job definitions request errors (Major code 413)" on page 262
440	Retrieve active jobs	"Retrieve active jobs request errors (Major code 440)" on page 263
450	Active job actions	"Job actions request errors (Major code 450)" on page 263
500	CONTROL-M/EM API Java Client	"CONTROL-M/EM API Java client errors (Major code 500)" on page 264
600	Gateway Messages	"Gateway messages (Major code 600)" on page 265

NULL exception errors (Major code 000)

CONTROL-M/EM API has generated an undefined error.

Table 148 NULL exceptions

Minor code	Severity	Description
0	ERROR	CONTROL-M/EM API has generated an undefined error.

Low-level API exceptions (Major code 100)

Errors generated due to a CONTROL-M/EM API initialization or request initialization failure.

Table 149 Low level API exceptions

Minor code	Severity	Description
1	ERROR	Error on preliminary parsing of the XML
2	FATAL	Catastrophic exception on server.
3	ERROR	Error: <i>value</i> Tag missing.
4	ERROR	Invalid Request. Request name: <i>value</i>
5	FATAL	Could not obtain response from repository.
6	ERROR	Undefined exception on server.

Parser exceptions (Major code 200)

Error occurred when parsing an XML file.

Table 150 Parser exceptions

Minor code	Severity	Description
1	FATAL	Parser Initialization Failure.
2	ERROR	Error while parsing XML: <i>value</i>
3	FATAL	Internal Parser Error: <i>value</i>
4	FATAL	Internal Parser Error: Undefined Exception

CONTROL-M/Server errors: Group 1 (Major code 300)

Errors generated by the CONTROL-M installation.

Table 151 CONTROL-M/Server errors: Group 1(part 1 of 3)

Minor code	Severity	Description
0	ERROR	Internal Server Error on Control/M.
3	ERROR	Control/M Error: <i>value</i> .
5	ERROR	Storage allocation failed for CTM/EM Gateway.
7	ERROR	Cannot open conditions file.
9	ERROR	Loading of Control-M Parameters failed.
18	ERROR	Internal error on GETMEM.
19	ERROR	DSN not in catalog.
20	ERROR	DSN - dynamic allocation failed.

Table 151 CONTROL-M/Server errors: Group 1(part 2 of 3)

Minor code	Severity	Description
21	ERROR	Internal error. Invalid request to CTMMEM.
22	ERROR	Maximum number of members or lines in member exceeded.
23	ERROR	Invalid return code from CTMMEM.
24	ERROR	Error while processing directory of library.
25	ERROR	Library operation failed.
40	ERROR	Cannot open print file.
42	ERROR	Open of IOA log file failed .
45	ERROR	Internal Error on IOACND.
46	ERROR	Invalid date format.
47	ERROR	Cannot open synchronization file.
48	ERROR	Internal Server Error on CONTROL-M.
49	ERROR	Internal Server Error on CONTROL-M.
51	ERROR	Cannot add the condition because it already exists.
52	ERROR	Cannot add the condition because the condition file is full.
53	ERROR	Cannot add the control resource because it already exists.
54	ERROR	Cannot add the control resource because the resource file is full.
55	ERROR	Cannot delete the condition because it does not exist.
56	ERROR	Cannot delete the quantitative resource because it is in use.
57	ERROR	Cannot add the quantitative resource because the resource file is full.
58	ERROR	Cannot add the quantitative resource because it already exists.
59	ERROR	Cannot delete the quantitative resource because it does not exist.
60	ERROR	Change command can only be issued against quantitative resources
61	ERROR	Invalid value in sign field.
62	ERROR	The value in a change command must be between 1-9999.
63	ERROR	Cannot load module (internal error).
64	ERROR	The requested table does not exist in the library.
70	ERROR	The requested job does not exist in the given table.
87	ERROR	Invalid date.
90	ERROR	Table or Member does not exist.
91	ERROR	Table or Member already exists.
93	ERROR	Invalid Group ID.
94	ERROR	The 'NEWG' order option is not supported in this version.
95	ERROR	The group entity specified was not found in the AJF.
96	ERROR	Failed to extract data from CTM/EM message.
97	ERROR	Failed to delete records from table.
98	ERROR	Insert into table failed.
99	ERROR	Update table failed.

Table 151 CONTROL-M/Server errors: Group 1(part 3 of 3)

Minor code	Severity	Description
100	ERROR	Failed to commit transaction.
101	ERROR	The filed GROUP is mandatory for a group entity.
102	ERROR	GROUP and JOBNAME should be the same for group entity.
103	ERROR	GROUP and SCHEDTAB should be the same for group entity.
104	ERROR	GROUP and SCHEDTAB should be specified.
105	ERROR	The group specified does not exist in the database.
106	ERROR	MEMLIB, CMDLINE, MEMNAME AND OVERLIB should not be specified.
107	ERROR	Failed to allocate ISN.
108	ERROR	No nodes found in node group.
109	ERROR	Failed to get next node in node group.
110	ERROR	Failed to setup application type for NODEGRP, MEMNAME and OVERLIB should not be specified.
111	ERROR	Invalid ODATE.

CONTROL-M/Server errors: Group 2 (Major code 301)

Errors generated by the CONTROL-M installation.

Table 152 CONTROL-M Server errors: Group 2 (part 1 of 2)

Minor code	Severity	Description
501	INFO	The job has been successfully ordered
502	ERROR	Unable to open the specified scheduling library.
506	ERROR	Scheduling failed for member.
510	ERROR	Scheduling member contains invalid data.
512	WARN	Library should be compressed.
514	ERROR	Job was not ordered. Reason: Insufficient storage.
515	ERROR	Job contains too many cards.
516	ERROR	Scheduling table error: First card is not a D statement.
517	ERROR	The specified library is empty.
524	ERROR	Ordering process has entered with errors.
525	ERROR	Ordering process ended successfully.
526	ERROR	Invalid data format.
528	INFO	Job has been successfully ordered.
531	ERROR	Ordering process was canceled by an user exit.
532	ERROR	Cannot open the CTM/EM active jobs file.

Table 152 CONTROL-M Server errors: Group 2 (part 2 of 2)

Minor code	Severity	Description
534	ERROR	Cannot open the AJF - AJF is corrupted, I/O error, or file is not really AJF.
535	ERROR	Cannot order a job while AJF is being formatted.
536	ERROR	Severe error in scheduling definition.
537	ERROR	The job order contains more information than what the CTM/EM can handle.
548	ERROR	The calendar specified in the job is either corrupted or invalid.
549	WARN	CONTROL-R is not installed. IFRERUN statement ignored.
550	WARN	CONTROL-R is not installed. SET statement ignored.
166	ERROR	The job contains a condition with a PREV/NEXT date that cannot be interpreted by CTM/EM.
169	INFO	The CTM/EM has finished handling the request. No jobs were scheduled.

CONTROL-M/Server errors: Group 3 (Major code 302)

Errors generated by the CONTROL-M installation.

Table 153 CONTROL-M Server errors: Group 3

Minor code	Severity	Description
129	ERROR	User exit not loaded.
540	ERROR	Security exit not loaded. Security checking is bypassed.
863	WARN	The AJF is nearly full.

Generic request exceptions (Major code 401)

Errors generated when CONTROL-M/EM cannot communicate with the CONTROL-M installation.

Table 154 Generic request exceptions

Minor code	Severity	Description
1	ERROR	Could not connect to Control-M.
2	ERROR	Invalid Control-M.
3	ERROR	Invalid response from Control-M.
4	ERROR	Internal Error: <i>value</i> .

Poll request errors (Major code 403)

Errors generated when a Polling request fails.

Table 155 Poll request errors

Minor code	Severity	Description
1	ERROR	Error: Invalid token supplied.

Add or Delete Condition request errors (Major code 404)

Errors generated when an Add Condition or Delete Condition request fails.

Table 156 Add or Delete Condition request

Minor code	Severity	Description
1	ERROR	Add or Delete condition failed (code %d).
2	ERROR	Add or Delete condition failed.
3	ERROR	Add or Delete condition aborted.
4	ERROR	Add or Delete condition timed out.
5	ERROR	Add condition failed.
6	ERROR	Add condition failed (code %d).
7	ERROR	Add condition failed, invalid option.
8	ERROR	Delete condition failed.
9	ERROR	Delete condition failed (code %d).
10	ERROR	Delete condition failed, invalid option.
11	ERROR	Condition's name is not valid.
12	ERROR	Condition's order date is not valid.
13	ERROR	Cannot Add or Delete condition, already in wanted state.

Order or Force request errors (Major code 405)

Errors generated when an Order request or Force request fails.

Table 157 Order/Force request errors

Minor code	Severity	Description
1	ERROR	Order request didn't pass validity checks. Error: <i>value</i>
2	ERROR	Group RBA not found for Group ID <i>value</i> .
3	ERROR	Order request failed in the server.
4	ERROR	No jobs were ordered.

Job tracking request errors (Major code 406)

Errors generated when a Job tracking request fails.

Table 158 Job tracking request errors

Minor code	Severity	Description
1	ERROR	Job was not found in the last AJF.
2	ERROR	Required Job information does not exist in the database
3	ERROR	Tracked Job failed in validity checks.

Authorization request errors (Major code 407)

Errors generated when a registration or unregistration request fails.

Table 159 Authorization request errors

Minor code	Severity	Description
1	ERROR	Invalid user token.
2	ERROR	Invalid user name.
3	ERROR	Register failed.
4	ERROR	Unregister failed.
5	ERROR	User not authorized.
6	ERROR	Account is locked.
7	ERROR	Password has expired.

Alerts request errors (Major code 408)

Errors generated when a Change Alert Status request fails.

Table 160 Alerts request errors

Minor code	Severity	Description
1	ERROR	Alert id is not valid.
2	ERROR	Invalid alert operation.

Create active job request errors (Major code 409)

Errors generated when a Job Creation request fails.

Table 161 Create active job request errors

Minor code	Severity	Description
1	ERROR	Create active job failed (code: %d).
2	ERROR	Create active job failed.
3	ERROR	Create active job failed, object is in use.
4	ERROR	Create active job failed, object not found.
5	ERROR	Create active job failed, invalid option.
6	ERROR	Create active job validation error: 'value '.
7	ERROR	Scheduling group is not valid for this data-center version
8	ERROR	Failed to initialize job descriptor
9	ERROR	Create active job, invalid order date.

Upload scheduling table request errors (Major code 411)

Errors generated when a Upload Scheduling Table request fails.

Table 162 Upload scheduling table request errors(part 1 of 2)

Minor code	Severity	Description
1	ERROR	Upload table failed, invalid parameters.
2	ERROR	Upload table failed.
3	ERROR	Cannot get Scheduling Table from database.

Table 162 Upload scheduling table request errors(part 2 of 2)

Minor code	Severity	Description
4	ERROR	Upload table aborted.
5	ERROR	Upload table timed-out.
6	ERROR	Upload table failed with status.

Create job/scheduling group definitions request errors (Major code 412)

Errors generated when a Create job/scheduling group definitions request fails.

Table 163 Create job/scheduling group definitions request errors

Minor code	Severity	Description
1	ERROR	Create job definitions failed, invalid parameters.
2	ERROR	Create scheduling table failed.
3	ERROR	Create job definitions failed.
4	ERROR	Failed to add jobs to scheduling table.
5	ERROR	Create jobs definitions failed, unknown reason.
6	ERROR	<TBD: Message text>
7	ERROR	Scheduling table already exists.
8	ERROR	Scheduling group is not valid for this data-center version.
9	ERROR	Failed to initialize job descriptor.
10	ERROR	Create definitions, API is not supported for CTM version '%s'.
11	ERROR	Scheduling table already contains a scheduling group.
12	ERROR	Illegal author in new job.
13	ERROR	Group name of a job or a scheduling group differs from a scheduling group or table name.
14	ERROR	Create jobs definitions validation error: '%s'.

Delete job definitions request errors (Major code 413)

Errors generated when a Delete job definitions request fails.

Table 164 Delete job definitions request errors

Minor code	Severity	Description
1	ERROR	Scheduling table does not exist.
2	ERROR	Many scheduling tables are found.
3	ERROR	Failed to delete jobs from scheduling table.
4	ERROR	No jobs were deleted according to the specified criterion.
5	ERROR	Failed to update number of jobs in scheduling group.
6	ERROR	Delete jobs definitions validation error: [<i><filter name></i>] is not valid filter field.
7	ERROR	Delete jobs definitions, invalid parameters.

Retrieve active jobs request errors (Major code 440)

Errors generated when a Retrieve active jobs request fails.

Table 165 Retrieve active jobs request errors

Minor code	Severity	Description
1	ERROR	Retrieve active jobs failed.
2	ERROR	Request failed - internal error.
3	ERROR	No jobs were found according to the specified criterion.
4	ERROR	Partial result.
5	ERROR	The maximum of returned nodes, which is specified in request, exceeds CONTROL-M/EM server limit, which is specified by EMAPIActiveJobsLoadLimit system parameter.
6	ERROR	Retrieve active jobs validation error: '%s' is not valid filter field.

Job actions request errors (Major code 450)

Errors generated when a Job actions request fails.

Table 166 Job actions request errors(part 1 of 2)

Minor code	Severity	Description
1	ERROR	Unknown error occurred.
2	ERROR	Action request failed (code %d).
3	ERROR	Action request failed.
4	ERROR	Action request aborted.

Table 166 Job actions request errors(part 2 of 2)

Minor code	Severity	Description
5	ERROR	Action request timed-out.
304	ERROR	Cannot hold job - not held.
305	ERROR	Active jobs file is locked, try again later.
306	ERROR	Job does not exist.
309	ERROR	Cannot hold job - already held.
310	ERROR	Job is not waiting for confirmation.
323	ERROR	Security protection violation.
328	ERROR	Job is not in execution state.
361	ERROR	The selected job/group cannot be deleted due to its current state.
506	ERROR	Operation not supported by agent.
305	ERROR	Job does not exist.
305	ERROR	Job actions, invalid order date.

CONTROL-M/EM API Java client errors (Major code 500)

Errors generated by the Java API.

Table 167 CONTROL-M/EM API Java client errors

Minor code	Severity	Description
1	ERROR	Fatal error.
2	ERROR	Null XML document - nothing to do.
3	ERROR	Failed to establish connection with server - no server registered under this hostname.
4	ERROR	Failed to resolve server name - please check your hostname.
5	ERROR	Failed to establish connection with server - not a valid component type, check your hostname.
6	ERROR	failed resolving XMLInvoker interface.
7	ERROR	Failed to establish connection with server - check your connection with the CORBA Naming Service.
8	ERROR	Failed to establish connection with - check your CORBA configuration.
9	ERROR	Properties file not found.
10	ERROR	Failed to read properties file.
101	ERROR	Invoke timeout - no response after.
102	ERROR	Invoke request exited because a InterruptedException occurred.
111	ERROR	Response format is not valid.
112	ERROR	Response format is not valid. Cannot find tag.

Table 167 CONTROL-M/EM API Java client errors

Minor code	Severity	Description
131	ERROR	XML format is not valid.
132	ERROR	Request format is not valid.
133	ERROR	Can't find user token.
140	ERROR	Error parsing the XML.
141	ERROR	Undefined Exception on Parser.

Gateway messages (Major code 600)

Errors generated by the gateway.

Table 168 Gateway messages (part 1 of 7)

Minor code	Severity	Description
1	ERROR	User Request timed out
2	ERROR	No network currently loaded
3	ERROR	No nodes match the Show Net parameters
4	ERROR	Net too large to be fully viewed. Some nodes are missing
5	ERROR	Database login failed three times
6	ERROR	At least one state must be chosen
7	ERROR	At least one status must be chosen
8	ERROR	At least one task type must be chosen
9	ERROR	No events found for current net
10	ERROR	Field <i>value</i> Wrong Format: <i>value</i>
11	ERROR	Cannot load Active Network: No data center
12	ERROR	Field <i>value</i> is required
13	ERROR	Net Load aborted, probably not enough memory
14	ERROR	No nodes match the Load Net Parameters
15	ERROR	Error while reading job-record from database
16	ERROR	Nothing changed since last save
17	ERROR	Nothing changed
18	ERROR	Cannot save file <i>value</i>
19	ERROR	Confirm <i>value</i> for <i>value</i>
20	ERROR	Do you really want to quit <i>value</i> ?
21	ERROR	In a critical job, parameter 'Priority' must begin with
22	ERROR	Field <i>value</i> contains an invalid value
23	ERROR	Field <i>value</i> must be between %d and %d

Table 168 Gateway messages (part 2 of 7)

Minor code	Severity	Description
24	ERROR	Field Rerun Mem Cannot be used if field Task Type is Cyclic
25	ERROR	Field <i>value</i> has an invalid value starting at position: <i>value</i>
26	ERROR	Field <i>value</i> may not contain embedded spaces
27	ERROR	Cannot clear Demo Net <i>value</i>
28	ERROR	Cannot create Demo Net <i>value</i>
29	ERROR	Cannot copy Demo Net <i>value</i>
30	ERROR	Couldn't open user view
31	ERROR	Do you really want to erase the entire <i>value</i> Net <i>value</i>
32	ERROR	Do you really want to clear the entire <i>value</i> Net <i>value</i>
33	ERROR	Cannot append, Owner of <i>value</i> is <i>value</i>
34	ERROR	Cannot open <i>value</i>
35	ERROR	Missing file <i>value</i>
36	ERROR	Unable un-mount diskette in <i>value</i>
37	ERROR	Unable to mount diskette in <i>value</i>
38	ERROR	Incorrect path <i>value</i>
39	ERROR	Files downloaded from incompatible version <i>value</i> (should be <i>value</i>)
40	ERROR	Cannot open file <i>value</i>
41	ERROR	Missing <i>value</i> line
42	ERROR	Unknown line: <i>value</i>
43	ERROR	Cannot write to file <i>value</i>
44	ERROR	File <i>value</i> exists. Ok to overwrite?
45	ERROR	Net <i>value</i> does not exist
46	ERROR	Copy from file to file not allowed
47	ERROR	Database job update failed
48	ERROR	Database select failed for resource events
49	ERROR	Database select failed for override table, simulation stopped
50	ERROR	No current net
51	ERROR	No nodes were found
52	ERROR	No more nodes were found Go Back?
53	ERROR	No more nodes were found
54	ERROR	You are not authorized to perform this action
55	ERROR	If field <i>value</i> (Days/Week days Calendar) begins with ALL then the rest of the parameter must be blank
56	ERROR	Field Days Calender is required if field <i>value</i> contains the value: <i>value</i>
57	ERROR	Field Week Calender is required if field <i>value</i> contains the value: <i>value</i>
58	ERROR	Field Days has an invalid value starting at position: <i>value</i> ; There should be a comma there
59	ERROR	Discard changes made in form?

Table 168 Gateway messages (part 3 of 7)

Minor code	Severity	Description
60	ERROR	Change of calendar type destroys existing data. Proceed?
61	ERROR	Change of data center platform destroys existing data. Proceed?
62	ERROR	Data center unknown, please re-enter value
63	ERROR	Reading calendar failed - database problems
64	ERROR	Calendar is already in use by user <i>value</i> , try later
65	ERROR	Calendar write failed - database problems
66	ERROR	Unknown platform. Verification of data defaults to MVS. Do you want to continue?
67	ERROR	Calendar was modified. Proceed with download?
68	ERROR	Calendar <i>value</i> was downloaded successfully from data center <i>value</i>
69	ERROR	Calendar <i>value</i> was deleted successfully from data center <i>value</i>
70	ERROR	Calendar is in use by user <i>value</i> , and cannot be deleted"
71	ERROR	Confirm Calendar Delete"
72	ERROR	Confirm Calendar Upload"
73	ERROR	Forced Calendar Upload overrides calendar content in data center Proceed?
74	ERROR	You are not authorized to access the Calendar List window
75	ERROR	Field <i>value</i> must be specified if field <i>value</i> is File, New Dest or Change Class
76	ERROR	Attempt to enter conflicting conditions
77	ERROR	Field <i>value</i> may only be used with the Change Class option
78	ERROR	Field <i>value</i> is too long
79	ERROR	Field Sign may not be + if the ODATE field value is **** or \$\$\$\$
80	ERROR	Unknown or disabled Platform! Form is opened as MVS!
81	ERROR	In a critical job, field Priority must begin with *
82	ERROR	Field Command Line may be specified only if Task Type is Command
83	ERROR	Field should be %d char(s) long for this Sysout option
84	ERROR	Field <i>value</i> cannot be used if Task Type is <i>value</i>
85	ERROR	Field <i>value</i> cannot specify GENREAL or USER=
86	ERROR	Fields Parm and From Class must be specified if field Option is Change Class
87	ERROR	Field Parm must be specified if field Option is New Dest or File
88	ERROR	Field Confirmation Cal cannot be used with fields PDS Name or Minimum
89	ERROR	Field Months cannot be used with fields PDS Name, Minimum or Dates
90	ERROR	Field Dates cannot be used with parameters PDS Name, Minimum, Months, Days, or Days Calendar
91	ERROR	Field Dates cannot be used with fields Week Days, Weeks Days Calender Confirmation Cal
92	ERROR	If field PDS Name is specified then Minimum must be specified and vice versa
93	ERROR	Field Priority must begin with * or an alphanumeric character
94	ERROR	Field Priority must begin with an alphanumeric character

Table 168 Gateway messages (part 4 of 7)

Minor code	Severity	Description
95	ERROR	Field <i>value</i> is required if Task Type is: <i>value</i>
96	ERROR	A job with the same name already exists in this table
97	ERROR	If field Retro is specified then PDS Name and Minimum cannot be used
98	ERROR	The fields Days and Week Days Calendar cannot both be specified
99	ERROR	The field <i>value</i> must contain a numeric value or 0
100	ERROR	Fields Rerun Mem and Max Rerun cannot be used if field Task Type is Cyclic
101	ERROR	Field Dates must be specified in format nnnn or nnnn,nnnn,...
102	ERROR	Scheduling table is already in use by user <i>value</i> , try later
103	ERROR	Scheduling table write failed - database problems
104	ERROR	Scheduling table should be uploaded before order/force
105	ERROR	Field <i>value</i> is required
106	ERROR	If field <i>value</i> is *???????, then field <i>value</i> must be blank
107	ERROR	Table was modified. Proceed with download?
108	ERROR	Table was modified. Proceed with order?
109	ERROR	Table was modified. Proceed with force?
110	ERROR	Definition table <i>value</i> was downloaded successfully from data center <i>value</i>
111	ERROR	Definition table <i>value</i> was deleted successfully from data center <i>value</i>
112	ERROR	Confirm Table Delete
113	ERROR	Confirm Table Upload
114	ERROR	Forced Table Upload overrides table content in data center Proceed?
115	ERROR	You are not authorized to access the Scheduling Tables window
116	ERROR	Scheduling table is in use by user <i>value</i> , and cannot be deleted
117	ERROR	Field <i>value</i> must be blank if field <i>value</i> is OK or NOTOK or RERUN
118	ERROR	Communication with WS-Gateway stopped, all requests were canceled
119	ERROR	System Error in <i>value</i> : Command <i>value</i> failed; <i>value</i>
120	ERROR	No communication with WS-Gateway. Request cancelled
121	ERROR	<i>value</i> not possible, no communication with Gateway
122	ERROR	<i>value</i> not possible, Unknown data center <i>value</i>
123	ERROR	<i>value</i> not possible, data center <i>value</i> not available
124	ERROR	Communication with data center <i>value</i> down, all requests canceled
125	ERROR	Data center <i>value</i> is suspended (AJF is being formatted)
126	ERROR	Data center <i>value</i> is suspended (Download in Netgroup)
127	ERROR	Data center <i>value</i> is suspended (AJF is being formatted), all requests canceled
128	ERROR	Invalid event %c received from data center <i>value</i> . To ensure database integrity, WS-GTW stop/restart is recommended
129	ERROR	The host name be specified only for the TCP protocol
130	ERROR	The host name may contain only Alphanumeric characters and periods

Table 168 Gateway messages (part 5 of 7)

Minor code	Severity	Description
131	ERROR	You must specify a communication protocol
132	ERROR	Specified communication protocol not supported on platform
133	ERROR	Control-R not supported on platform
134	ERROR	The Net Group field must contain two alpha numeric characters
135	ERROR	CONTROL-M must be 300 or 400 for MVS
136	ERROR	Incorrect CONTROL-M version specified for this platform
137	ERROR	Scheduling table read failed - database problems
138	ERROR	Please verify that all users have quit CONTROL-M/EM and that the WS-Gateway is down.
139	ERROR	Do you really want to quit CONTROL-M/Enterprise Manager Administration?
140	ERROR	No Net
141	ERROR	Number of nodes in a row exceeds maximum of %d
142	ERROR	Cannot generate report due to insufficient memory
143	ERROR	Cannot print links: <i>value</i>
144	ERROR	Cannot print index: <i>value</i>
145	ERROR	Cannot do poster for routed net
146	ERROR	Job Report fields saved. Please close and open the job report window
147	ERROR	New field is empty
148	ERROR	Confirm delete
149	ERROR	No Loaded Networks Available
150	ERROR	Control Resources with <i>value value</i> cannot be <i>value</i>
151	ERROR	Quantitative Resources with <i>value value</i> cannot be <i>value</i>
152	ERROR	Field Author must be entered
153	ERROR	Order ID <i>value</i> not found
154	ERROR	Pattern <i>value</i> not found in text
155	ERROR	Pattern <i>value</i> appears only once in the text
156	ERROR	Problems searching for pattern <i>value</i>
157	ERROR	Alarm handling command ignored. User <i>value</i> is handling this alarm now
158	ERROR	Alarm handling command ignored. Internal error
159	ERROR	Download of calendar <i>value</i> canceled in gateway - calendar in use
160	ERROR	Download of calendar <i>value</i> canceled in gateway - Another download is currently in progress - Try later
161	ERROR	Upload of calendar <i>value</i> canceled in gateway
162	ERROR	Upload of calendar <i>value</i> canceled in gateway - calendar in use
163	ERROR	Upload of definitions table <i>value</i> canceled in gateway
164	ERROR	Upload of definitions table <i>value</i> canceled in gateway - table in use
165	ERROR	Download of definition table <i>value</i> canceled in gateway - table in use

Table 168 Gateway messages (part 6 of 7)

Minor code	Severity	Description
166	ERROR	Download of definition table <i>value</i> canceled in gateway - Another download is currently in progress - Try later
167	ERROR	Definitions table deleted on data center but delete failed on workstation (rc 1)
168	ERROR	Definitions table deleted on data center but delete failed on workstation (rc 2)
169	ERROR	Calendar deleted on data center but delete failed on workstation (rc 2)
170	ERROR	Delete of definitions table <i>value</i> canceled in gateway - table in use
171	ERROR	Delete of calendar <i>value</i> canceled in gateway - calendar in use
172	ERROR	User request canceled in gateway, no answer from data center (time-out)
173	ERROR	Download of <i>value</i> canceled, no answer from data center (time-out)
174	ERROR	Download of <i>value</i> canceled in gateway
173	ERROR	Download canceled, no answer from data center (time-out)
174	ERROR	Download canceled in gateway
175	ERROR	Upload canceled in gateway
176	ERROR	Upload canceled (file transfer error (rcp) in gateway), consult system administrator
177	ERROR	Information request canceled in gateway, system error (fopen failed), consult system administrator
178	ERROR	Information request canceled (time-out) in gateway, no answer from data center
179	ERROR	Upload of <i>value</i> canceled (time-out), no answer from data center
180	ERROR	Upload of <i>value</i> canceled in gateway
179	ERROR	Upload canceled (time-out), no answer from data center
180	ERROR	Upload canceled in gateway
181	ERROR	No fields defined
182	ERROR	Field <i>value</i> not in table
182	ERROR	Field not in table
183	ERROR	Internal error on field <i>value</i>
183	ERROR	Internal error on field
184	ERROR	Cannot print %d characters; Maximum of %d allowed
185	ERROR	Cannot print more than %d fields
186	ERROR	You are not authorized to load an active network
187	ERROR	<i>value</i> was not found in data center. Delete in workstation?
188	ERROR	You are not authorized to access the Global Conditions window
189	ERROR	Definition table <i>value</i> was deleted only on workstation
190	ERROR	Calendar <i>value</i> was deleted only on workstation
191	ERROR	Internal error. Action failed
192	ERROR	Insert failed. Record with same key already exists.
193	ERROR	Record deleted in the meantime.
194	ERROR	Error message <i>value</i> Received from data center <i>value</i> .

Table 168 Gateway messages (part 7 of 7)

Minor code	Severity	Description
194	ERROR	Error message received from data center.
195	ERROR	Field <i>value</i> is required if field <i>value</i> is defined
195	ERROR	Field required.
196	ERROR	Field <i>value</i> should be at most %d characters
197	ERROR	Field <i>value</i> may not contain the characters <i>value</i>
198	ERROR	Field <i>value</i> must be between <i>value</i> and <i>value</i>
199	ERROR	Field <i>value</i> should be empty
200	ERROR	Field <i>value</i> must be blank if field When is OK or NOTOK or RERUN
201	ERROR	Field <i>value</i> must be <i>value</i>
202	ERROR	No response to heartbeat check from data center: <i>value</i>
202	ERROR	No response to heartbeat check from data center.
203	ERROR	Unsupported version <i>value</i> of data center <i>value</i> . Check CTM installation
203	ERROR	Unsupported version of data center, check CTM installation.
204	ERROR	Not the smallest ISN for ordered/forced job

Job and Scheduling Group XML parameters

Table 169 job and sched_group XML parameters description (part 1 of 9)

Parameter	Description
active_from	Indicates the start of a period of time during which the job or group scheduling table can be ordered.
active_till	Indicates the end of a period of time during which the job or group scheduling table can be ordered.
adjust_condition	Indicates whether to ignore prerequisite conditions normally set by predecessor jobs if the relevant predecessor jobs are not scheduled. This parameter is relevant only for jobs in a Group Scheduling table. Optional. Valid values: <ul style="list-style-type: none"> ■ no – (Do not ignore. Default.) ■ yes – (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 MEMLIB parameter of the job set to DUMMY.
and_or	Indicates the relationship between Month Days parameter values and Week Days parameter values.
application	Name of the application to which the job's group belongs. Used as a descriptive name for related groups of jobs.
application_cm_version	Indicates the version of external application (for example, SAP or Oracle Applications) Control Module (CM) that is installed in the CONTROL-M installation. Specified together with the application_form , application_type , and application_version elements.
application_form	Specifies a predefined set of external application parameters that are displayed in the External Application panel of the CONTROL-M/EM Job Editing form. Specified together with the application_cm_version , application_type , and application_version elements.
application_type	Indicates the type of external application (for example, SAP or Oracle Applications) on which the external application job runs. Specified together with the application_cm_version , application_form , and application_version elements.

Table 169 job and sched_group XML parameters description (part 2 of 9)

Parameter	Description
application_version	Indicates the version of the external application (for example, SAP or Oracle Applications) on which the external application job runs. Specified together with the application_cm_version , application_form , and application_type elements.
arch_max_days	Maximum number of days to retains the SYSDATA archive data set for jobs that ended NOTOK . [z/OS only]
arch_max_runs	Maximum number of job runs to retain the SYSDATA archive data set for jobs that ended NOTOK . [z/OS only]
author	CONTROL-M/EM user who defined the job. Note: This argument is used by the CONTROL-M security mechanism and under certain circumstances, cannot be modified. For more information, see the Security chapter and the description of the AuthorSecurity system parameter in the <i>CONTROL-M Administrator Guide</i> .
auto_archive	Determines whether or not SYSDATA is to be archived. [z/OS only] Valid values: <ul style="list-style-type: none"> ■ yes ■ no
autoedit_assignments	A sequence of autoedit_assignment . See Table 185 .
command	Command string supplied when the job Task Type (the task_type element) is Command . Optional.
confirm_flag	Specifies whether user confirmation is required before the job is submitted for execution. String. Valid values: <ul style="list-style-type: none"> ■ no - Job needs no confirmation to run. Default. ■ yes - Job must be confirmed to run.
control_resources	A sequence of control_resource . See Table 187 .
count_cyclic_from	Indicates whether the interval between successive runs of a cyclic job is calculated from the start or the end of the previous job run. Specified only for cyclic jobs (when the cyclic element is specified). Valid values: <ul style="list-style-type: none"> ■ start - Counts interval from the start of the previous job run ■ end - Counts interval from the end of the previous job run ■ target - Counts interval from the scheduling time of the current job run.
conf_cal	Calendar used to confirm job scheduling dates.
critical	Indicates that the job is a critical-path job in CONTROL-M. Valid values: <ul style="list-style-type: none"> ■ yes - a critical path job ■ no - not a critical path job
ctld_category	CONTROL-M for z/OS only: Name of a CONTROL-D Report Decollating Mission category to be scheduled whenever the job is run.

Table 169 job and sched_group XML parameters description (part 3 of 9)

Parameter	Description
cyclic	Indicates if a job is cyclic. Valid values: <ul style="list-style-type: none"> ■ yes – Cyclic job ■ no – Non-cyclic job
cyclic_type	If job is cyclic (cyclic equal yes), indicates how the intervals for running the job are specified. Valid values are: <ul style="list-style-type: none"> ■ interval: Job is run at fixed interval. See rerun_interval. ■ interval_sequence: Job is run according to a list of time periods. See interval_sequence. ■ specific_times: Job is run according to a list of specific times. See specific_times. Note: This parameter is relevant only for jobs running in CONTROL-M/Server version 6.4.01 and later.
dates	A sequence of date . For more information, refer to Table 170 . CONTROL-M for z/OS: A maximum of 12 date elements can be specified.
days_cal	Name of a user-defined calendar containing a list of days of the month, used with Month Days, to determine a set of working days.
days_due_out_offset	The number of days that job execution can be extended after the ODAT. Note: days_due_out_offset is relevant only for jobs running in CONTROL-M for z/OS version 6.2.00 and later.
description	Text description of the job.
doc_lib	Name of the directory/library containing the job documentation file.
doc_member	Name of the file containing job documentation.
group	Name of the group to which the job belongs. Used as a descriptive name for related groups of jobs.
in_conditions	A sequence of in_condition . See Table 186 .
instream_jcl	JCL stream forming part of the job definition. Note: instream_jcl is relevant for jobs running in: <ul style="list-style-type: none"> ■ CONTROL-M for z/OS version 6.2.00 and later ■ CONTROL-M/Server version 6.4.01 and later
interval_sequence	A sequence of interval_item . See Table 195 Note: This parameter is relevant only for jobs running in CONTROL-M/Server version 6.4.01 and later.
job_name	The name of the job.
job_sched_tags	A sequence of job_tag . For more information, refer to Table 194 .
max_wait	Number of extra days (beyond the original scheduling date) that the job is allowed to remain in the Active Jobs file awaiting execution. Integer.
mem_lib	Name of the library/directory in which the job script resides. String.

Table 169 job and sched_group XML parameters description (part 4 of 9)

Parameter	Description
mem_name	Name of the file that contains the job script. String.
min_pds_tracks	Minimum number of free partitioned data set tracks required by the library specified for the PDS parameter.
month_days	Indicates the days of the month on which the job should be scheduled for processing.
multiagent	When selected, broadcasts job submission details to all Agents within a specified Node Group. Not for z/OS. Optional. Valid values: <ul style="list-style-type: none"> ■ yes – run as multi-agent job ■ no – not run as multi-agent job. Default.
node_id	Node ID of the host on which the job was most recently run. Note: This parameter is not available for CONTROL-M for z/OS jobs.
on_do_statements	A sequence of on_do_statement . For more information, refer to Table 171 .
out_conditions	A sequence of out_condition . See Table 189 .
over_lib	Name of the alternate job script library/directory.
owner	The owner of the job.
pds	CONTROL-M for z/OS only: Name of a partitioned data set (PDS) to be checked for free space. If the PDS has fewer than the minimum number of required free tracks (as specified for the Minimum parameter), the job is executed.
prevent_nct2	Prevents data set cleanup before the original job run. [z/OS only]. Optional. Valid values: <ul style="list-style-type: none"> ■ no – Does not prevent data set cleanup. Default. ■ yes – Prevents data set cleanup. ■ blank - Do not perform data set cleanup before the original job run. ■ 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. ■ flush - Halt processing of the job if any data set cleanup error is detected (even if MVS would not have stopped processing the job).
priority	Indicates CONTROL-M job priority. String.
quantitative_resources	A sequence of quantitative_resource . See Table 188 .
request_nje	Specifies the node in the JES network on which the job is to execute. [z/OS only] String.

Table 169 job and sched_group XML parameters description (part 5 of 9)

Parameter	Description
rerun_interval	Specifies the length of time to wait between reruns of a job or between cyclic runs of a job. The value is expressed as a number and a letter. The number indicates the amount. The letter indicates the unit of measurement. Valid values: <ul style="list-style-type: none"> ■ 0 - 64800M (minutes) ■ 0 - 1080H (hours) ■ 0 - 45D (days) Default: 0 .
rerun_max	Specifies the maximum number of reruns that can be performed for the job. Integer. Valid values: 0-99 .
rerun_member	Name of the JCL member to use when the job is automatically rerun. [z/OS only] String. Optional.
reten_days	Number of days to retain the job in the History Jobs file. [z/OS, only]. String.
reten_gen	Maximum number of generations of the job to keep in the History Jobs file. [z/OS, only]. String.
retro	Indicates whether the job should be scheduled for possible execution after its original scheduled date has passed.
sac	CONTROL-M for z/OS, version 6.2xx and later Indicates whether to adjust the logical date for a job converted from a scheduling product other than CONTROL-M. Optional. String. Valid values are: <ul style="list-style-type: none"> ■ blank. Default. ■ previous ■ next ■ plus (for group scheduling definitions only) ■ minus (for group scheduling definitions only)
schedule_environment	Indicates the JES2 workload management scheduling environment that is to be associated with the job. z/OS, only. String.
sched_tags	A sequence of tag. For information about tag XML parameters, refer to Table 192
shift	When to schedule the job if the date is not confirmed. (Option) Valid values are: <ul style="list-style-type: none"> ■ ignore_job. Do not shift the job to a different date. The job is not scheduled. ■ next_day. Shift to the next working date. ■ prev_day. Shift to the previous working date. ■ no_confcal. Tentatively schedule the job for the current day (even if not a working day). Additional shifting may or may not be performed, depending on the value indicated in the shift_num parameter.
shift_num	The number of working days that a job can be shifted. Values from -62 to 62 can be entered. This function is also called Extended Shift.
shouts	A sequence of shout . See Table 191 .

Table 169 job and sched_group XML parameters description (part 6 of 9)

Parameter	Description
specific_times	A sequence of specific_time . See Table 196 . Note: This parameter is relevant only for jobs running in CONTROL-M/Server version 6.4.01 and later.
statistic_calendar	Name of the CONTROL-M periodic calendar within which statistics relating to the job are collected. Note: statistic_calendar is relevant only for jobs running in CONTROL-M for z/OS version 6.2.00 and later.
step_ranges	A sequence of step_range . See Table 190 .
sys_db	Indicates that a single data set is used for archiving the SYSDATA of all jobs until it is full, when another data set is started. [z/OS only] Valid values: <ul style="list-style-type: none"> ■ yes – Single data set created for the SYSDATA of each job run. ■ no – Separate data set created for the SYSDATA of each job run.
sysout_from_class	Limits the sysout handling operation to only sysouts from the specified class. [z/OS only]
sysout_option	Sysout Handling options. Optional. Valid values (non-z/OS): <ul style="list-style-type: none"> ■ copy ■ delete ■ move ■ release Valid values (z/OS): <ul style="list-style-type: none"> ■ copy ■ delete ■ move ■ release ■ change_class
sysout_parameter	Certain sysout_option values require that you supply additional information (such as Copy , NewDest): <ul style="list-style-type: none"> ■ If the sysout_option element is change_class, the sysout_parameter value corresponds to the new class name. ■ If the sysout_option element is copy, the sysout_parameter value corresponds to the destination file name. ■ If the sysout_option element is move, the sysout_parameter value corresponds to the new destination for the file.
system_affinity	Indicates the identity of the system in which the job must be initiated and executed (in JES2). Indicates the identity of the processor on which the job must execute (in JES3). Note: For z/OS jobs only.
tag_relationship	Indicates the relationship (AND/OR) between Scheduling Tag 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).

Table 169 job and sched_group XML parameters description (part 7 of 9)

Parameter	Description
task_type	<p>Type of the job (task) to be performed by CONTROL-M.</p> <p>Microsoft Windows and UNIX</p> <ul style="list-style-type: none"> ■ job ■ command ■ dummy ■ detached ■ external <p>CONTROL-M for z/OS</p> <ul style="list-style-type: none"> ■ job ■ task ■ cyclic_job ■ emergency_job ■ emergency_cyclic_job ■ cyclic_task ■ emergency_task ■ emergency_cyclic_task
time_due_out	Time that the job is expected to finish.
time_from	Indicates the earliest time for submitting the job.
time_from_days_offset	<p>Number of days after the original scheduling date of the job during which execution of the job can begin.</p> <p>Note: time_from_days_offset is relevant only for jobs running in CONTROL-M for z/OS version 6.2.00 and later.</p>
time_until	Indicates the latest time for submitting the job.
time_until_days_offset	<p>Number of days after the original scheduling date of the job during which execution of the job can end.</p> <p>Note: time_until_days_offset is relevant only for jobs running in CONTROL-M for z/OS version 6.2.00 and later.</p>
time_zone	Indicates the global time zone used to calculate the interval for time-related conditions.
tolerance	<p>Maximum delay in minutes permitted for late submission when selecting a specific time.</p> <p>Note: This parameter is relevant only for jobs running in CONTROL-M/Server version 6.4.01 and later.</p>
use_instream_jcl	<p>Whether the job submits a JCL stream defined within the job scheduling definition.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> ■ yes ■ no <p>Note: use_instream_jcl is relevant for jobs running in:</p> <ul style="list-style-type: none"> ■ CONTROL-M for z/OS version 6.2.00 and later ■ CONTROL-M/Server version 6.4.01 and later

Table 169 job and sched_group XML parameters description (part 8 of 9)

Parameter	Description
weeks_cal	Name of a user-defined, week-based calendar used together with parameter Week Days to specify a set of working days.
week_days	Indicates the days of the week on which the job should be scheduled for processing.
JAN	Indicates whether to run the job in this month. Valid values are: <ul style="list-style-type: none"> ■ yes ■ no
FEB	Indicates whether to run the job in this month. Valid values are: <ul style="list-style-type: none"> ■ yes ■ no
MAR	Indicates whether to run the job in this month. Valid values are: <ul style="list-style-type: none"> ■ yes ■ no
APR	Indicates whether to run the job in this month. Valid values are: <ul style="list-style-type: none"> ■ yes ■ no
MAY	Indicates whether to run the job in this month. Valid values are: <ul style="list-style-type: none"> ■ yes ■ no
JUN	Indicates whether to run the job in this month. Valid values are: <ul style="list-style-type: none"> ■ yes ■ no
JUL	Indicates whether to run the job in this month. Valid values are: <ul style="list-style-type: none"> ■ yes ■ no
AUG	Indicates whether to run the job in this month. Valid values are: <ul style="list-style-type: none"> ■ yes ■ no
SEP	Indicates whether to run the job in this month. Valid values are: <ul style="list-style-type: none"> ■ yes ■ no
OCT	Indicates whether to run the job in this month. Valid values are: <ul style="list-style-type: none"> ■ yes ■ no

Table 169 job and sched_group XML parameters description (part 9 of 9)

Parameter	Description
NOV	Indicates whether to run the job in this month. Valid values are: <ul style="list-style-type: none"> ■ yes ■ no
DEC	Indicates whether to run the job in this month. Valid values are: <ul style="list-style-type: none"> ■ yes ■ no

Table 170 date XML parameter description

Parameter	Description
date	Indicates a specific date, in either mmdd or ddmmy format (depending on the site standard), on which the job should be scheduled.

Figure 120 on_do_statement XML parameters

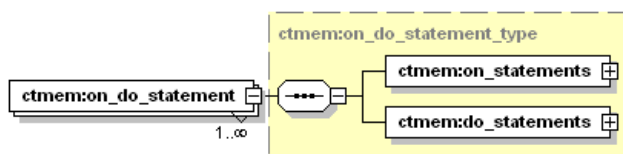


Table 171 on_do_statement XML Parameters Description

Parameter	Description
on_statements	A sequence of on_statements . See Table 172 .
do_statements	A sequence of do_statements . See Table 175 .

Figure 121 on_statements XML parameters

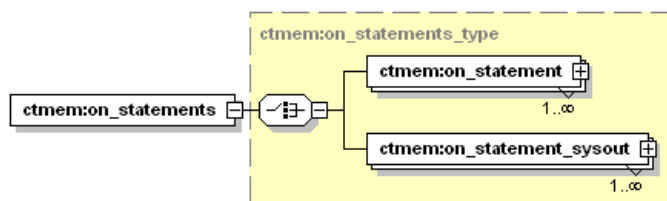


Table 172 on_statements XML parameters description

Parameter	Description
on_statement	A sequence of on_statement . See Table 173
OR	
on_statement_sysout	A sequence of on_statement_sysout . See Table 174

Figure 122 on_statement XML parameters

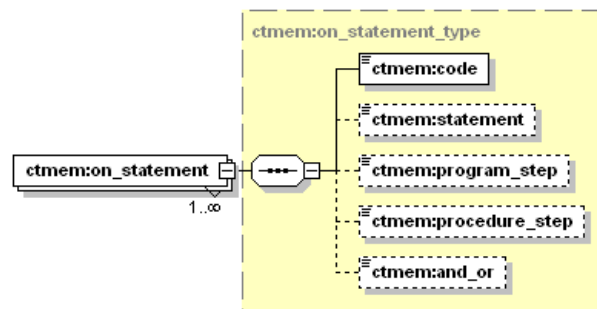


Table 173 on_statement XML parameters description

Parameter	Description
and_or	Indicates the relationship between Month Days parameter values and Week Days parameter values.
code	Code value for the On Statement/Code parameter. Valid values: <ul style="list-style-type: none"> ■ ok ■ not_ok
procedure_step	Step in the procedure that triggers the On statement. String.
program_step	Step in the program that triggers the On statement. String.
statement	statement can be: <ul style="list-style-type: none"> ■ A character string containing a statement from the job script file (1-132 characters). The specified string can be a portion of the statement. ■ An asterisk (*), when code is a completion status for a job.

Figure 123 on_statement_sysout XML parameters

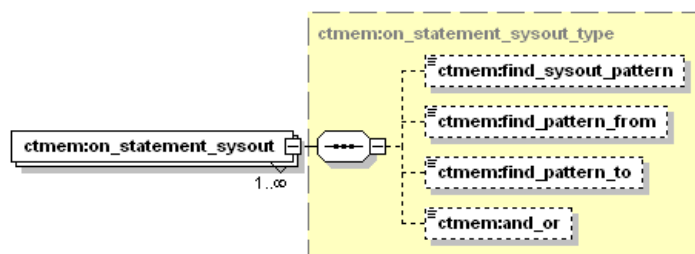


Table 174 on_statement_sysout XML parameters description

Parameter	Description
find_sysout_pattern	A string of up to 40 characters.
find_pattern_from	A number from 001 through 132, indicating the column at which the search should start. If this field is blank, the value 001 is assumed. The value in this field must be lower than that in the To Column field.
find_pattern_to	A number from 001 through 132, indicating the column at which the search should end. If this field is blank, the value 132 is assumed. The value in this field must be higher than that in the From Column field.
and_or	Option buttons that set the logical relationship between multiple On statements.

Figure 124 do statements type XML parameters

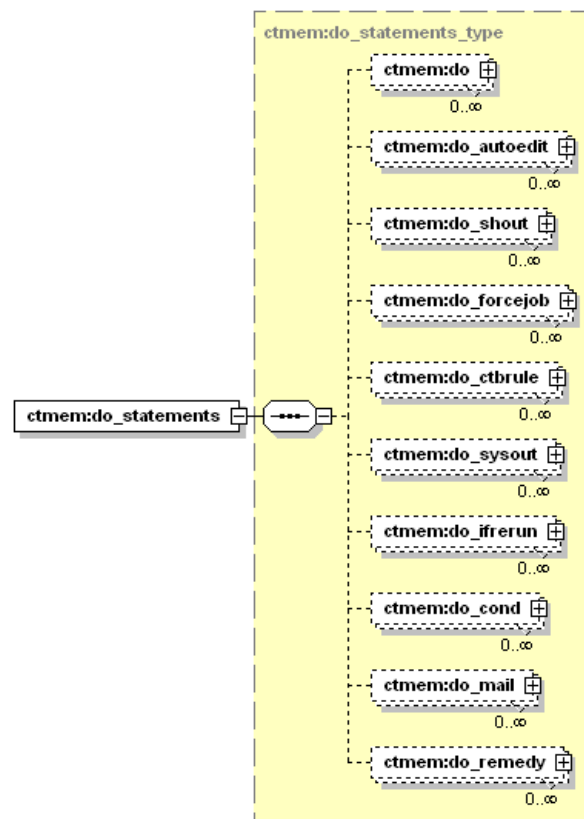


Table 175 do statements type XML Parameters Description (part 1 of 2)

Parameter	Description
do	A sequence of do_statements .
do_autoedit	Assigns an AutoEdit variable when the On criteria are met. See Table 176 .
do_cond	Assigns an In or Out condition when the On criteria are met. See Table 177 .
do_ctbrule	Invokes a CONTROL-M/Analyzer rule to be executed during the processing of a specific program step when an On condition is met. See Table 178 .

Table 175 do statements type XML Parameters Description (part 2 of 2)

Parameter	Description
do_forcejob	Forces a specified job when the current job is performed. Note: The dsn element is for z/OS jobs only. See Table 179 .
do_ifrerun	Specifies job steps to be executed during rerun of a job. Only for networks using CONTROL-M/Restart. See Table 180 .
do_mail	Sends mail.
do_remedy	Creates a Remedy ticket.
do_shout	Sends a shout message when the On criteria are met.
do_sysout	Determines what to do with the sysout documentation when On criteria are met.

Figure 125 do_autoedit XML parameters

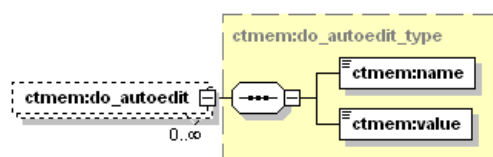


Table 176 do_autoedit XML parameters description

Parameter	Description
name	Name of the item in question (for example, when specified for request , name is the name of the request; when specified for pipe , name is the name of the pipe)
value	Value of the AutoEdit expression.

Figure 126 do_cond XML parameters

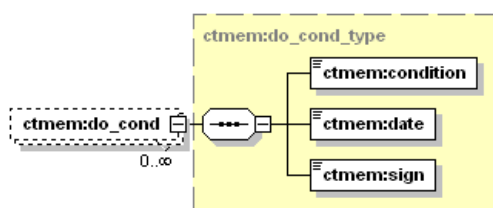


Table 177 do_cond XML parameters description

Parameter	Description
condition	Condition name. When specified, it is be accompanied by the other condition parameter element, date (and, optionally, by sign or and_or). ■ Wrapped in the in_condition and out_condition elements.
date	Specifies an order date for various condition formats.
sign	Indicates whether to add or delete an Out condition Valid values: ■ add ■ delete

Figure 127 do_ctbrule XML parameters

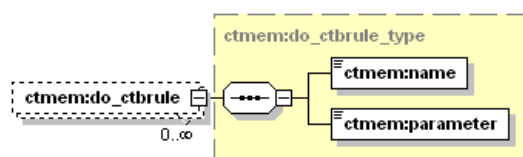


Table 178 do_ctbrule XML parameters description

Parameter	Description
name	Name of the CONTROL-M/Analyzer rule.
parameter	Contains arguments that are passed to the CONTROL-M/Analyzer rule.

Figure 128 do_forcejob XML parameters

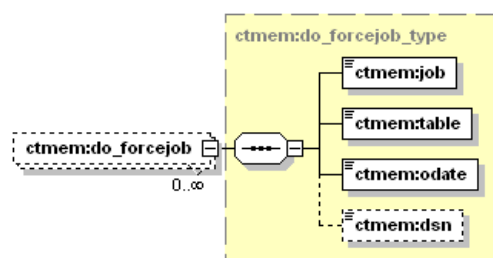


Table 179 do_forcejob XML parameters description

Parameter	Description
dsn	Name of the directory/library containing Scheduling table file. [z/OS only]
job	Specifies the job name of the job that is forced.
odate	Original scheduling date of a job.
table	Name of the Scheduling table with which the job specified in <code>do_forcejob</code> is associated.

Figure 129 do_ifrerun XML parameters

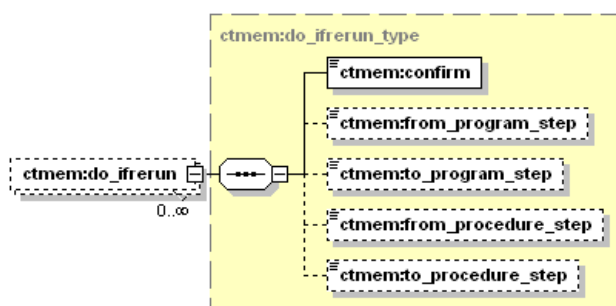


Table 180 do_ifrerun XML parameters description

Parameter	Description
confirm	Indicates that a job rerun specified by the Do If Rerun parameter must be manually confirmed before it is executed. Valid values: <ul style="list-style-type: none"> ■ yes – requires confirmation ■ no – no confirmation required
from_procedure_step	Procedure step (EXEC statement) that invokes a procedure from which the specified program step program is executed.
from_program_step	Job step. The execution results of the program executed by the job step are checked against the specified codes criteria.
to_procedure_step	Last procedure step in a range.
to_program_step	Last program step in a range.

Figure 130 do_mail XML parameters

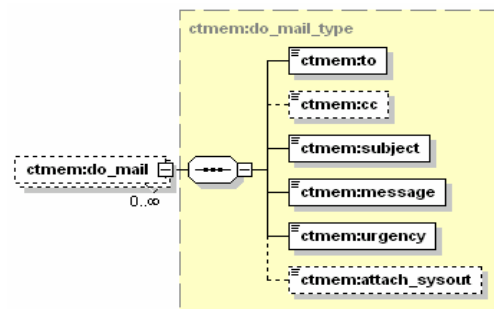


Table 181 do_mail XML parameters description

Parameter	Description
cc	Optional additional address to which a Do Mail can be sent. Optional.
message	Text of the message. String.
to	Recipient of the do_mail message.
subject	Subject of the do_mail message.
urgency	Indicates the severity of a mail or shout message. Valid values: <ul style="list-style-type: none"> ■ regular (Default) ■ urgent ■ very_urgent
attach_sysout	Specifies whether the sysout should be sent as an email attachment. Valid values are: <ul style="list-style-type: none"> ■ yes - Send the job's sysout as an attachment ■ no - Do not send the job's sysout as an attachment ■ default - Use the settings configured for the relevant CONTROL-M server to determine whether the job's sysout should be sent as an attachment. Note: This parameter is relevant only for jobs running in CONTROL-M/Server version 6.4.01 and later.

Figure 131 do_shout XML parameters

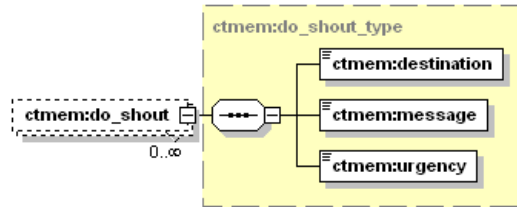


Table 182 do_shout XML parameters description

Parameter	Description
destination	Recipient of a Shout message. Specified in both the Shout or the Do Shout parameters.
message	Text of the message. String.
urgency	Indicates the severity of a mail or shout message. Valid values: <ul style="list-style-type: none"> ■ regular (Default) ■ urgent ■ very_urgent

Figure 132 do_sysout XML parameters

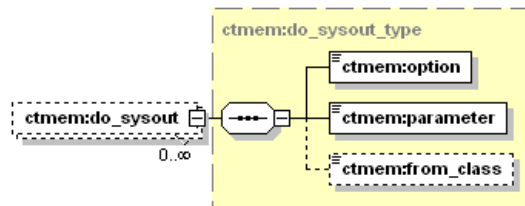


Table 183 do_sysout XML parameters description

Parameter	Description
from_class	Specifies the class of jobs with sysouts that are handled using the Do Sysout specifications of the job.

Table 183 do_sysout XML parameters description

Parameter	Description
option	<p>Do Sysout parameter sysout handling options.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ■ Release ■ Delete ■ Copy ■ Move ■ File ■ NewDest ■ ChangeClass <p>Note: Copy and Move are not used with z/OS. File, NewDest, and ChangeClass are used only with z/OS.</p>
parameter	<p>Contains additional sysout handling information. The type of information required is dependent on the value of the option element.</p> <ul style="list-style-type: none"> ■ If the option element is ChangeClass, the parameter value corresponds to the new class name. ■ If the option element is Copy, the parameter value corresponds to the destination file name. ■ If the option element is Move, the parameter value corresponds to the new destination for the file.

Figure 133 do_remedy XML parameters

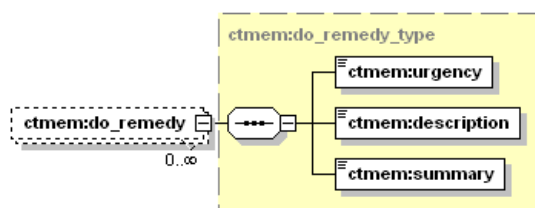


Table 184 do_remedy XML parameters description

Parameter	Description
urgency	<p>The urgency of the Remedy ticket. Valid values are:</p> <ul style="list-style-type: none"> ■ low ■ medium ■ high ■ urgent ■ clear
description	The description of the problem for which you are opening up the ticket.
summary	A summary of the problem for which you are opening up a ticket.

Figure 134 autoedit_assignment XML parameters

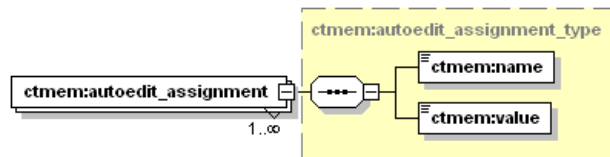


Table 185 autoedit_assignment XML parameters description

Parameter	Description
name	Name of theAutoEdit variable.
value	Value of the AutoEdit expression.

Figure 135 in_condition XML parameters

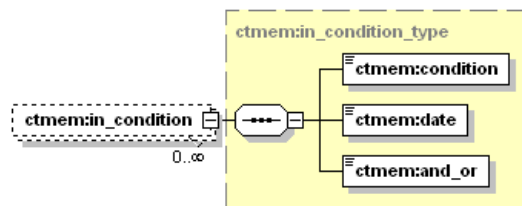


Table 186 in_condition XML parameters description

Parameter	Description
and_or	Specifies the relationship between two successive items in a series. Optional. Valid values: <ul style="list-style-type: none"> ■ and ■ or
condition	Condition name.
date	Specifies an order date for various condition formats.

Figure 136 control_resources XML parameters

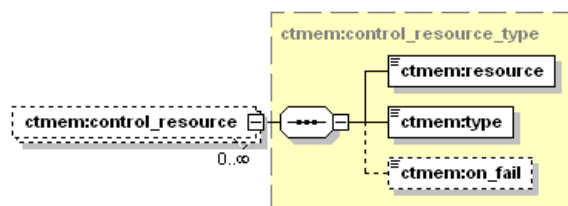


Table 187 control_resources XML parameters description

Parameter	Description
resource	Name of the specified resource.
type	Indicates job access to a Control resource. Valid values are: <ul style="list-style-type: none"> ■ exclusive - default ■ shared
on_fail	Whether to keep a Control resource tied to a job if the job does not end OK. Valid values: <ul style="list-style-type: none"> ■ keep ■ release - default

Figure 137 quantitative_resource XML parameter

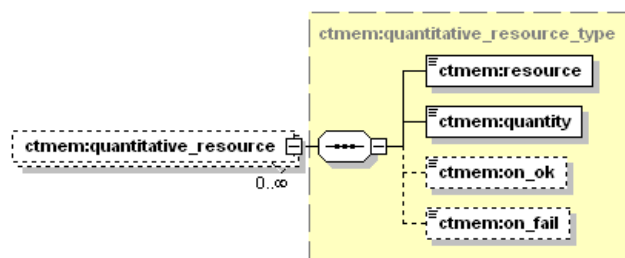


Table 188 quantitative_resource XML parameters description

Parameter	Description
quantity	Amount of the specified quantitative resource.
resource	Name of the specified resource.
on_ok	Whether to keep a Quantitative resource tied to a job if the job ends OK. Valid values are: <ul style="list-style-type: none"> ■ release ■ discard <p>Note: on_ok is relevant only for jobs running in CONTROL-M for z/OS version 6.2.00 and later.</p>
on_fail	Whether to keep a Quantitative resource tied to a job if the job does not end OK. Valid values are: <ul style="list-style-type: none"> ■ keep ■ release <p>Note: on_fail is relevant only for jobs running in CONTROL-M for z/OS version 6.2.00 and later.</p>

Figure 138 out_condition XML parameters

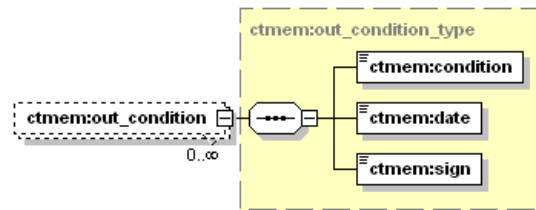


Table 189 out_condition XML parameters description

Parameter	Description
condition	Condition name. When specified, it is be accompanied by the other condition parameter element, date (and, optionally, by sign or and_or).
date	Specifies an order date for various condition formats.
sign	Indicates whether to add or delete an Out condition Valid values: <ul style="list-style-type: none"> ■ add ■ delete

Figure 139 step_range XML parameters

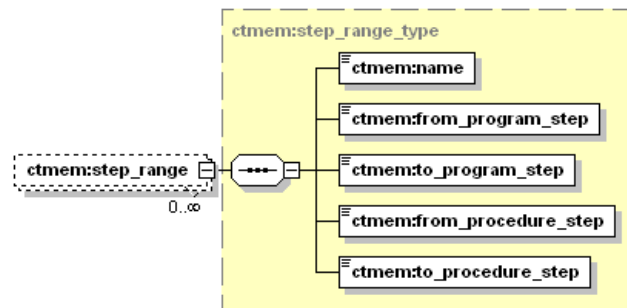


Table 190 step_range XML parameters description

Parameter	Description
from_procedure_step	Procedure step (EXEC statement) that invokes a procedure from which the specified program step program is executed.
from_program_step	Job step. The execution results of the program executed by the job step are checked against the specified codes criteria.
to_procedure_step	Last procedure step in a range.
to_program_step	Last program step in a range.

Figure 140 shouts XML parameters

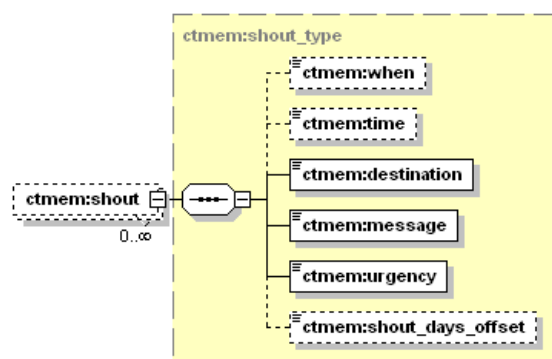


Table 191 shouts XML parameters description

Parameter	Description
destination	Recipient of a Shout message. Specified in both the Shout or the Do Shout parameters.
message	Text of the message. String.
time	Time that the message is sent.
urgency	Indicates the severity of a mail or shout message. Valid values: <ul style="list-style-type: none"> ■ regular (Default) ■ urgent ■ very_urgent
when	Time that the Shout message was sent. Valid values: <ul style="list-style-type: none"> ■ ok ■ not_ok ■ rerun (not valid for scheduling group entities) ■ late_submission ■ late_time ■ execution_time
shout_days_offset	The number of days relative to the ODAT by which the sending of the Shout message is offset. Valid values are: <ul style="list-style-type: none"> ■ a number from 0 through 254 ■ blank – no offset Note: shout_days_offset is relevant only for jobs running in CONTROL-M for z/OS version 6.2.00 and later.

Figure 141 tag XML parameters

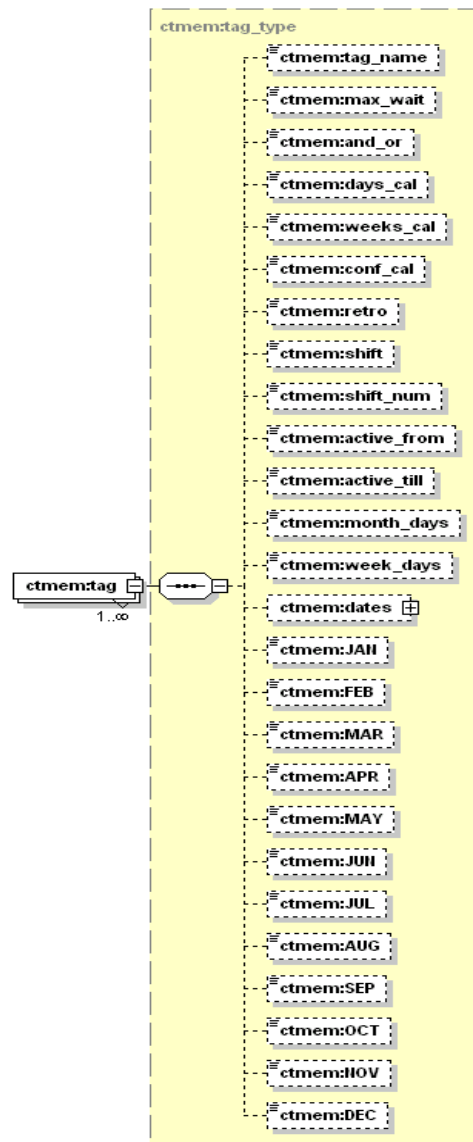


Table 192 tag XML parameters description(part 1 of 3)

Parameter	Description
tag_name	The name of the tag.
max_wait	Maximum number of days that the job can wait to be executed after its original scheduling date has passed.
and_or	Indicates the relationship between Month Days parameter values and Week Days parameter values. Optional.
days_cal	Name of a user-defined calendar containing a list of days of the month, used with Month Days, to determine a set of working days.
weeks_cal	Name of a user-defined, week-based calendar used together with parameter Week Days to specify a set of working days.

Table 192 tag XML parameters description(part 2 of 3)

Parameter	Description
conf_cal	Calendar used to confirm job scheduling dates.
retro	Indicates whether the job should be scheduled for possible execution after its original scheduled date has passed.
shift	When to schedule the job if the date is not confirmed. (Option) Valid values are: <ul style="list-style-type: none"> ■ ignore_job. Do not shift the job to a different date. The job is not scheduled. ■ next_day. Shift to the next working date. ■ prev_day. Shift to the previous working date. ■ no_confcal. Tentatively schedule the job for the current day (even if not a working day). Additional shifting may or may not be performed, depending on the value indicated in the shift_num parameter.
shift_num	The number of working days that a job can be shifted. Values from -62 to 62 can be entered. This function is also called Extended Shift.
active_from	Indicates the start of a period of time during which the job or group scheduling table can be ordered.
active_till	Indicates the end of a period of time during which the job or group scheduling table can be ordered.
month_days	Indicates the days of the month on which the job should be scheduled for processing.
week_days	Indicates the days of the week on which the job should be scheduled for processing.
dates	A sequence of date . For more information, refer to Table 170 . CONTROL-M for z/OS: A maximum of 12 date elements can be specified.
JAN	Indicates whether to run the job in this month. Valid values are: <ul style="list-style-type: none"> ■ yes ■ no
FEB	Indicates whether to run the job in this month. Valid values are: <ul style="list-style-type: none"> ■ yes ■ no
MAR	Indicates whether to run the job in this month. Valid values are: <ul style="list-style-type: none"> ■ yes ■ no
APR	Indicates whether to run the job in this month. Valid values are: <ul style="list-style-type: none"> ■ yes ■ no

Table 192 tag XML parameters description(part 3 of 3)

Parameter	Description
MAY	Indicates whether to run the job in this month. Valid values are: <ul style="list-style-type: none"> ■ yes ■ no
JUN	Indicates whether to run the job in this month. Valid values are: <ul style="list-style-type: none"> ■ yes ■ no
JUL	Indicates whether to run the job in this month. Valid values are: <ul style="list-style-type: none"> ■ yes ■ no
AUG	Indicates whether to run the job in this month. Valid values are: <ul style="list-style-type: none"> ■ yes ■ no
SEP	Indicates whether to run the job in this month. Valid values are: <ul style="list-style-type: none"> ■ yes ■ no
OCT	Indicates whether to run the job in this month. Valid values are: <ul style="list-style-type: none"> ■ yes ■ no
NOV	Indicates whether to run the job in this month. Valid values are: <ul style="list-style-type: none"> ■ yes ■ no
DEC	Indicates whether to run the job in this month. Valid values are: <ul style="list-style-type: none"> ■ yes ■ no

Table 193 date parameters description

Parameter	Description
date	Indicates a specific date, in either mmdd or ddmm format (depending on the site standard), on which the job should be scheduled.

Figure 142 job_tag XML parameters

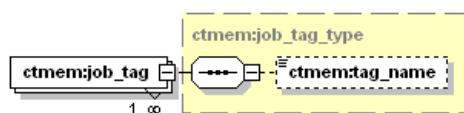


Table 194 job_tag XML parameters description

Parameter	Description
tag_name	The name of the tag.

Figure 143 interval_sequence XML parameters

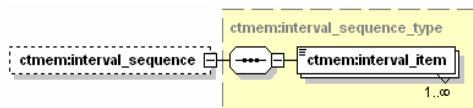


Table 195 interval_sequence XML parameters description

Parameter	Description
interval_item	Time interval to rerun a cyclic job such as +2H, +1D, or +30M. Limited to 4000 characters for all fields.

Figure 144 specific_times XML parameters

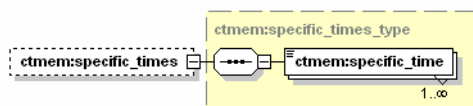


Table 196 specific_times XML parameters description

Parameter	Description
specific_time	Specific time for a cyclic job to run, such as 7:00 or 11:00. Limited to 4000 for all fields.

Glossary

A

Active Jobs File

The Active Jobs file lists all jobs scheduled for submission during the current day. Each job in the Active Jobs file is not submitted until all conditions in the job processing definition for the job are satisfied. The Active Jobs file is in the CONTROL-M database.

Alert

Alerts are messages that indicate important information for the CONTROL-M/EM user. These messages normally indicate when a problem or exception has occurred for a job controlled by CONTROL-M. All alerts are displayed in the Global Alert Client **Alerts** window.

C

CONTROL-M

Software product that schedules, submits, tracks and follows up the execution of jobs in a specific installation. In certain releases, CONTROL-M functions are divided between two separate components: CONTROL-M/Server and CONTROL-M/Agent.

CONTROL-M/Enterprise Manager or CONTROL_M/EM

Software product that provides a central point of control for CONTROL-M installations. CONTROL-M/EM provides the GUI that allows users to graphically view the status of job schedules and execution in CONTROL-M installations, to issue requests for additional information, to make changes in the Active Jobs file, and to handle problems. CONTROL-M/EM also passes global conditions among CONTROL-M installations.

CONTROL-M/EM GUI Server

Process that handles communication between CONTROL-M/EM GUI workstations and other components of CONTROL-M/EM. The CONTROL-M/EM Server executes database queries, calculations and procedures for each GUI, thereby lessening its workload and streamlining productivity by enabling data-sharing between GUIs. Multiple CONTROL-M/EM Servers can be installed in an CONTROL-M/EM environment.

CORBA

An architecture for creating, distributing, and managing program objects across distributed computers.

G**Gateway**

The process that handles communication between CONTROL-M and CONTROL-M/EM. There are gateway processes on both the CONTROL-M platform and on the CONTROL-M/EM workstation.

Global Alerts Server

The process that identifies and distributes alerts between CONTROL-M installations and CONTROL-M/EM workstations. The Global Alerts Server connects to each CONTROL-M/EM gateway to receive alerts from CONTROL-M and transmit them to the CONTROL-M/EM GUIs.

J**Java**

Java is an object-oriented programming language for use in a distributed computing environment.

Job Processing Definition

Set of user-defined parameters for each job which provide CONTROL-M with detailed instructions on processing the job. job processing definitions are organized into Scheduling

P**Prerequisite Conditions or Conditions**

A flag representing a user-specified situation or condition. Submission of a job for execution can be made dependent upon the existence of one or more prerequisite conditions. Prerequisite conditions are recorded in the CONTROL-M/EM Conditions/Resources table.

R**Response**

Text string returned by CONTROL-M/EM containing status information regarding the request for which it is returned.

Request

An action that is sent to CONTROL-M/EM from a remote location, using the CONTROL-M/EM API. Some requests are processed by CONTROL-M/EM; others are forwarded to CONTROL-M for processing. There are two broad types of requests:

- Requests that manage the transfer of information between the CONTROL-M/EM API and CONTROL-M/EM.
- Requests that contain actions that influence the CONTROL-M/EM active environment.

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

X

XML

Extensible Markup Language (XML) is a specification for designing markup languages used to organize information.

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Index

Symbols

.dtd files, removal during upgrade 41

A

accessing
 CORBA naming service 35
 Global Alerts Server 35
 GUI Server 35
 actions in CONTROL-M/EM API session 62
 activating project 47
 add condition request
 description 159
 errors 161, 259
 examples 162
 parameters 159
 polling request parameters 160
 polling requirement 59
 polling response parameters 160
 response parameters 160
 adding
 conditions 159
 directory pathnames to CLASSPATH 35
 advanced features and optimization
 getting and setting CONTROL-M/EM API properties 219
 polling interval timeout configuration 221
 advantages
 to EMBasicXMLInvoker 55
 to EMXMLInvoker 59
 alerts
 changing status 205
 errors in changing status 261
 API
 initializing 61
 initializing processes 61
 low-level exceptions 254
 modifying initialization properties 215
 application runtime 229
 Authorization request errors 260

B

BMC Software, contacting 2

C

calls
 creating, sending, handling 49
 EMXMLInvoker 61
 function, failure 229
 request types 49
 response types 50
 change alert status request
 description 205
 errors 261
 changePass script 35
 ChangePass.bat script 35
 choosing a CONTROL-M/EM API class 54
 classes
 choosing 54
 ComponentType 68
 EMBasicXMLInvoker 69
 EMXMLInvoker 72
 GASComponent 78
 GSRComponent 80
 InvokeException 81
 list 67
 sample 44
 CLASSPATH
 adding directory pathnames 35
 omissions 226
 troubleshooting 226
 com.bmc.ctmem.emapi.XMLDATAPATH property 65
 communication errors 265
 ComponentType class 68
 conditions
 errors in adding 259
 errors in deleting 259
 configuration files 35
 configuring
 CONTROL-M/EM API 43
 differences between platforms 34
 environment 226
 log file 225
 polling timeout 221
 project environment 43
 script 36
 setPollRequestIntervalMilli 221
 setPollRequestTimeoutMilli 221

- CONTROL-M supported versions 31
- CONTROL-M/EM API
 - adding directory pathnames to CLASSPATH 35
 - configuration files 35
 - configuration script 36
 - configuring 43
 - connecting to CONTROL-M/EM 26
 - how it works 26
 - importing into project 45
 - initializing 26
 - initializing services 61
 - installing on Microsoft Windows 33
 - installing on Red Hat 33
 - installing on Suse 33
 - installing on Unix 33
 - prerequisites 32
 - primary subdirectories 34
 - programming methods 54
 - session example 27
 - sessions 27
 - software compatibility 31
 - software requirements 31
 - stopping services 61
 - supported CONTROL-M versions 31
 - supported CONTROL-M/EM components 26
- CONTROL-M/EM API session
 - flow of actions 62
 - starting and stopping services 62
- CONTROL-M/EM component
 - specifying in a request call 57, 63
- CONTROL-M/Server
 - communication errors 258
 - errors 255, 257
 - internal error 255
- conventions, documentation 22
- CORBA
 - accessing naming service 35
 - advanced implementation 215
 - description 215
 - initializing process 61
 - initializing processes 26
 - modifying properties 215
 - passing parameters to project 47
 - specifying implementation 227
- creating
 - emapi_log.cfg file 224
 - EMXMLInvoker calls 61
 - EMXMLInvoker class request calls 61
 - group scheduling tables 133
 - jobs in group scheduling tables 133
 - regular jobs 133
 - XML strings 67
- creating job definitions 102, 112
- creating scheduling group definitions 106
- ctmemapi.properties file
 - component hostnames 57, 64
 - configuration 35

- copy 44
- troubleshooting 230
- customer support 2

D

- DEBUG severity level 225, 253
- debugging
 - severity levels 225, 253
 - using Log4J 223
- delete condition request
 - description 164
 - errors 167, 259
 - fault response parameters 167
 - parameters 165
 - polling request parameters 166
 - polling requirement 59
 - polling response parameters 166
 - response parameters 165
- development environment 43
- diagnostics. *See* troubleshooting
- differences
 - between EMBasicXMLInvoker and EMXMLInvoker 54
 - between the native and basic APIs 54
- directories
 - missing 226
 - of the CONTROL-M/EM API 34
- disadvantages
 - to EMBasicXMLInvoker 55
 - to EMXMLInvoker 59
- done method
 - primary listing 73
 - stopping API services 62

E

- emapi_env.bat file 35, 44, 47
- emapi_env.csh 35
- emapi_env.sh 35
- emapi_env.sh file 44, 47
- emapi_log.cfg file 224
- emapi-613 directory tree 33, 34
- emapi-admin file 35
- emapi-admin.bat file 35
- emapi-configure file 35
- emapi-configure.exe file 35
- EMAPIMessages.txt file 252
- EMBasicXMLInvoker
 - derived methods 69
 - description 69
 - invoking 58
 - when to use 55
- EMXMLInvoker
 - calls 61

- derived methods 72
- invoking 64
- when to use 59
- environment
 - setting variables 47
 - solving problems 226
- error codes
 - list of major codes 253
 - major code (defined) 252
 - minor code (defined) 252
 - reference 253
- ERROR severity level 225, 253
- errors
 - active jobs file 258
 - adding conditions 259
 - application runtime 229
 - authorizations 260
 - categories 251
 - change alert status request 261
 - communication 229, 265
 - conditions 256
 - contents 252
 - CONTROL-M/Server 255, 257
 - CONTROL-M/Server communication 258
 - creating group scheduling tables 261, 262, 263
 - creating jobs 261, 262, 263
 - deleting conditions 259
 - described in EMAPIMessages.txt file 252
 - description 83, 252
 - gateway 265
 - generic request exceptions 258
 - hostnames 229
 - in XML format 265
 - Java client 264
 - job creation request 261, 262, 263
 - job tracking request 260
 - JVM parameters 227
 - low-level API exceptions 254
 - major code 252
 - minor code 252
 - NULL exceptions 254
 - order/force request 259
 - ordering jobs 257
 - parser exceptions 255
 - polling 259
 - registration request 260
 - resources 256
 - severity level 224, 252
 - tracking jobs 260
 - unregistration request 260
 - user name invalid 260
 - user token invalid 260
- exceptions, throwing 58
- executing
 - java commands 47

F

- FATAL severity level 225, 253
- fault response parameters 161, 167
- fixing
 - application runtime errors 229
 - CLASSPATH problems 226
 - communication errors 229
 - environment problems 226
 - JVM problems 227
- force request 124
 - errors 259
 - polling requirement 59
- forcing
 - group scheduling tables 124
 - jobs 124
- formatting requests 49
- function call failure 229

G

- GAS
 - hostname 35
 - hostname as property 219
 - hostname errors 230
 - location 35
 - specifying in a request call 63
- GASComponent
 - derived methods for class 78
 - prototype 1 79
 - prototype 2 79
 - specifying in a request call 57
- gateway errors 265
- generic request exceptions 258
- getMajorCode method 82
- getMinorCode method 82
- getProperties method
 - advanced use 219
 - defaults 219
 - description 74
- getReason method 83
- Global Alerts Server
 - component type class 68
 - configuration 35
 - connecting to CONTROL-M/EM 26
 - CONTROL-M/EM API class 68
 - CONTROL-M/EM API properties parameters 219
 - described 298
 - EMBasicXMLInvoker class 69
 - GASComponent class 78
 - getProperties 74
 - prototype 79
 - request types 63
 - troubleshooting 230
 - with fatal error 253
- group scheduling tables

- creating [133](#)
- errors in creating [261](#), [262](#), [263](#)
- force a job [124](#)
- order a job [124](#)

GSRComponent

- class [80](#)
- method [80](#)
- prototype 1 [80](#)
- prototype 2 [81](#)
- specifying in an XMLBasicInvoker request call [63](#)
- specifying in an XMLInvoker request call [57](#)

GUI Server

- connecting to CONTROL-M/EM [26](#)
- getProperties [74](#)
- hostname [35](#)
- hostname as property [219](#)
- hostname errors [230](#)
- location [35](#)
- request types [63](#)
- with fatal error [253](#)

H

hostnames

- errors [230](#)
- recording in the ctmemapi.properties file [57](#), [64](#)
- specifying [35](#)
- specifying in a call [63](#)

I

import command [45](#)

importing CONTROL-M/EM API into project [45](#)

INFO severity level [225](#), [253](#)

init method

- modifying properties [215](#)
- primary listing [75](#)
- prototype 1 [75](#), [216](#)
- prototype 2 [75](#), [216](#)
- prototype 3 [76](#), [217](#)
- prototypes [216](#)
- starting API services [62](#)
- troubleshooting [228](#)

initializing

- API [61](#)
- API processes [61](#)
- CONTROL-M/EM API [26](#)
- CONTROL-M/EM API services [61](#)
- CORBA processes [26](#), [61](#)

installing CONTROL-M/EM API [33](#)

invalid user name [260](#)

invoke method

- primary listing [70](#), [77](#)
- string parameter [57](#), [63](#)
- throwing exceptions [58](#)

InvokeException class [81](#)

invoking EMBasicXMLInvoker [58](#)

invoking EMXMLInvoker [64](#)

J

JacORB, specifying implementation [227](#)

Java

- classes, role in API function [26](#)
- client errors [264](#)

java command [47](#)

JAVA_HOME environment variable [32](#)

JDK requirement and support [32](#)

job creation request

- description [133](#)
- determining parameter names [88](#)
- errors [262](#), [263](#)
- polling requirement [59](#)
- response [154](#)
- responses [133](#)

job tracking request errors [260](#)

jobs

- creating [133](#)
- errors in creating [261](#)
- errors in forcing [259](#)
- errors in ordering [257](#), [259](#)
- parameter names [88](#)

JRE requirement and support [32](#)

JVM

- advanced parameter configuration [227](#)
- parameter problems [227](#)
- passing parameters to project [47](#)
- requirement and support [32](#)
- requirement to run project [46](#)

L

Log4J

- logging mechanism [223](#)
- supported versions [223](#)

log4j.jar file [223](#)

logging

- category [224](#)
- configuring log file [225](#)
- default behavior [224](#)
- modifying behavior [224](#)
- parameters [224](#)
- using Log4J [223](#)

low-level API exceptions [254](#)

M

methods

- BuildPasswordString [73](#)
- done [62](#), [73](#)

- GASComponent 79
- getMajorCode 82
- getMinorCode 82
- getProperties 74, 219
- getReason primary listing 83
- GSRComponent 80
- init 62, 75
- invoke 57, 63, 70
- invoke (EMXMLInvoker) 77
- of the EMBasicXMLInvoker class 69
- of the EMXMLInvoker class 72
- of the GASComponent class 78
- of the GSRComponent class 80
- of the InvokeException class 81
- setPollRequestIntervalMilli 71
- setPollRequestIntervalMilli implementation 221
- setPollRequestTimeoutMilli 71
- setPollRequestTimeoutMilli implementation 221
- setProperties 77, 220
- Microsoft Windows, installing CONTROL-M/EM API 33
- migration. *See* upgrade considerations
- modifying
 - initialization properties 215
 - logging behavior 224
 - project for use with the CONTROL-M/EM API 44

N

- NamingViewer file 35
- NamingViewer.vbs file 35
- null exceptions 226, 254

O

- objects
 - invoking using EMBasicXMLInvoker 58
 - invoking using EMXMLInvoker 64
- order request
 - errors 259
 - polling requirement 59
- ordering
 - group scheduling tables 124
 - jobs 124

P

- parser exceptions 255
- parsing
 - response string 50
 - XML errors 255
- passing CORBA parameters to project 47
- performing actions on jobs in active environment 170
- polling
 - configuring interval timeout 221
 - request errors 259

- request parameters 160, 166
- polling response parameters 160, 166
- post-installation configuration 43
- preparing project environment 43
- product support 2
- project
 - modifying for use with the CONTROL-M/EM API 44
 - preparing environment 43
 - running 47
 - writing 44
- properties
 - com.bmc.ctmem.emapi.XMLDATAPATH 65
 - errors 264
 - getting 219
 - Global Alerts Server 35
 - GUI Server 35
 - modifying init 215
 - setting 219
- prototypes
 - done method 62
 - GASComponent prototype 1 79
 - GASComponent prototype 2 79
 - getProperties method 219
 - GSRComponent prototype 1 80
 - GSRComponent prototype 2 81
 - init 62
 - init method 216
 - init prototype 1 75, 216
 - init prototype 2 75, 216
 - init prototype 3 76, 217
 - setPollRequestIntervalMilli method 222
 - setPollRequestTimeoutMilli method 222
 - setProperties method 220
- publications, related 21

R

- Red Hat, installing CONTROL-M/EM API 33
- registration request, errors 260
- related publications 21
- request format examples
 - add condition or delete condition 237, 238
 - job creation requests 239
 - order or force requests 247
- request types
 - asynchronous 49
 - synchronous 49
- requests
 - add condition 159
 - change alert status 205
 - check user token validity 94
 - client keep alive 96
 - create job definitions 102
 - create scheduling group definitions 106
 - delete condition 164
 - delete job definitions 112

- fault response [213](#)
- format examples [237](#)
- job actions in active environment [170](#)
- job creation [133](#)
- job tracking [184](#)
- order or force [124](#)
- parameter names [88](#)
- parameters [159](#), [165](#)
- registration [260](#)
- retrieve BIM services list [209](#)
- retrieve jobs in active environment [194](#)
- sending [54](#)
- SOAP envelope [89](#)
- submitting [54](#)
- upload scheduling table [119](#)
- user registration [90](#)
- user unregistration [99](#)
- response
 - handling [50](#)
 - parameters [160](#), [165](#)
 - string, parsing [50](#)
 - to job creation request [154](#)
- retrieving active jobs [194](#)
- running
 - environment [43](#)
 - java command [47](#)
 - project [47](#)

S

- selecting
 - a CONTROL-M/EM API class [54](#)
 - component to process the request [57](#), [63](#)
- sending requests [54](#)
- setPollRequestIntervalMilli method
 - configuring [221](#)
 - default [222](#)
 - primary listing [71](#)
- setPollRequestTimeoutMilli method
 - configuring [221](#)
 - default [222](#)
 - primary listing [71](#)
- setProperty method
 - primary listing [77](#)
 - use [220](#)
- setting environmental variables [44](#), [47](#)
- severity levels
 - DEBUG [225](#), [253](#)
 - ERROR [225](#), [253](#)
 - FATAL [225](#), [253](#)
 - INFO [225](#), [253](#)
 - WARN [225](#), [253](#)
- Simple Object Access Protocol. *See* SOAP
- SOAP
 - envelope for requests and responses [89](#)
 - Simple Object Access Protocol [89](#)

- software requirements
 - CONTROL-M [31](#)
 - JDK [32](#)
 - JRE [32](#)
 - JVM [32](#)
- solving
 - application runtime errors [229](#)
 - CLASSPATH problems [226](#)
 - communication errors [229](#)
 - environment problems [226](#)
 - JVM problems [227](#)
- specifying
 - component to process a request [57](#), [63](#)
 - hostnames [35](#)
- starting
 - API processes [61](#)
 - CONTROL-M/EM API services [61](#)
 - CORBA processes [61](#)
 - project [47](#)
- stopping
 - API processes [61](#)
 - CONTROL-M/EM API services [61](#)
 - CORBA processes [61](#)
 - project [62](#)
- submitting requests [54](#)
- support
 - CONTROL-M versions [31](#)
 - JDK [32](#)
 - JRE [32](#)
 - JVM [32](#)
- support, customer [2](#)
- Suse, installing CONTROL-M/EM API [33](#)
- syntax statement conventions [23](#)

T

- technical support [2](#)
- throwing exceptions [58](#)
- timeout configuration [221](#)
- tracking
 - ID used with EMXMLInvoker [59](#)
 - job [184](#)
 - job request errors [260](#)
- troubleshooting
 - application runtime errors [229](#)
 - CLASSPATH problems [226](#)
 - communication problems [229](#)
 - CONTROL-M/EM API logging [223](#)
 - default logging behavior [224](#)
 - environment configuration [226](#)
 - incompatible object argument for function call [229](#)
 - init method [228](#)
 - Java virtual machine parameters [227](#)
 - JVM problems [227](#)
 - missing libraries or directories [226](#)
 - modifying logging behavior [224](#)

U

Unix, installing CONTROL-M/EM API [33](#)
unregistration request errors [260](#)
uploading a scheduling table [119](#)
user token, invalid [260](#)

V

validation
 file location [65](#)
 XML strings [65](#)

W

WARN severity level [225](#), [253](#)

X

XML strings
 creating [67](#)
 parsing errors [255](#)
 validation [65](#)
XML, invalid format errors [265](#)
XMLDATAPATH [65](#)

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Notes



94083