CONTROL-M/Agent for UNIX and Microsoft Windows

UNIX Administrator Guide

Supporting

CONTROL-M / Agent for UNIX version 6.2.01

September 15, 2005

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

Restricted rights legend

Customer support

You can obtain technical support by using the Support page on the BMC Software website or by contacting Customer Support by telephone or e-mail. To expedite your inquiry, please see “Before Contacting BMC Software.”

Support website

You can obtain technical support from BMC Software 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 Software offers
- find the most current information about BMC Software products
- search a database for problems similar to yours and possible solutions
- order or download product documentation
- report a problem or ask a question
- subscribe to receive e-mail notices when new product versions are released
- find worldwide BMC Software 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 support@bmc.com. Outside the United States and Canada, contact your local support center for assistance.

Before contacting BMC Software

Before you contact BMC Software, have the following information available so that Customer Support can begin working on your problem 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 problem
- 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

Contents

Figures

Tables

About This Book

This book describes CONTROL-M/Agent administration and provides information about parameters and utilities.

This book contains:

Chapter /Appendix	Description
Chapter 1	**"Understanding CONTROL-M/Agent"** Introduces CONTROL-M/Agent functions and concepts.
Chapter 2	**"Implementing CONTROL-M/Agent"** Describes how to start and stop CONTROL-M/Agent, maintain CONTROL-M/Agent, and make a Communication Diagnostic report.
Chapter 3	**"Writing Scripts"** Describes how to write CONTROL-M scripts.
Chapter 4	**"Utilities"** Describes CONTROL-M/Agent utilities and how to use them.
Appendix A	**"Configuration Parameters"** Describes the parameters in the **CONFIG.dat** and **OS.dat** configuration files.
Index	

Related Publications

- **CONTROL-M Installation Guide** describes the installation of CONTROL-M components.
- **CONTROL-M/Server Administrator Guides** each describe setup, maintenance, security, and utilities for CONTROL-M/Server on a specific type of computer.
- **CONTROL-M Job Parameter and Variable Reference Guide** describes syntax and usage for all parameters and variables that are included in CONTROL-M job processing definitions.
- **CONTROL-M/Desktop User Guide** describes how to define and manage CONTROL-M job processing definitions, Scheduling tables, and Calendars.
- **CONTROL-M/Enterprise Manager User Guide** describes CONTROL-M/EM concepts, features, facilities, and operating instructions.
- **CONTROL-M/Enterprise Manager Administrator Guide** describes tasks that the CONTROL-M/EM administrator must perform to define, monitor, and maintain the CONTROL-M/EM environment.
- **CONTROL-M/Enterprise Manager Utility Guide** describes command-line utilities that can be used to perform various CONTROL-M/EM tasks in batch mode.
- **CONTROL-M/eTrigger Administrator Guide** describes how to trigger job submission and tracking activities using a web-based (HTML) interface.

Notational Conventions

The following abbreviations are used in this guide:

Abbreviation	Description
CM	CONTROL-M/Control Module A product library for a specific application or operating system used by CONTROL-M/Agent to support that application or operating system.
CONTROL-M/EM	CONTROL-M/Enterprise Manager
NIS	Network Information System

The following conventions are used in this guide:

<key>	When describing keystrokes, angle brackets are used to enclose the name of a key (for example, **<F1>**). When two keys are joined with "+" as in **<Shift>**+**<F1>**, hold down **<Shift>** while pressing **<F1>**.
Menu => Option	This represents an option selection sequence. For example: **Users and Groups => Groups => Add** means that you first select **Users and Groups** from the menu bar; then select the **Groups** option from the submenu. Finally, select the **Add** option from the Groups submenu.
{Option A | Option B}	The vertical bar is used to separate choices. For example, when used as part of a parameter, {AND | OR} means that you specify either AND or OR.
[Parameter]	Square brackets are used to enclose parameters that are optional.
<variable>	In commands and parameters, angle brackets are used to enclose variable information. For example, the command: **cd <controlm_path>** means that you specify **cd** followed by the path of CONTROL-M.
italic	An italic font is used for the name of publications.

Chapter 1

Understanding CONTROL-M/Agent

This guide describes concepts and tools required to set up and manage CONTROL-M / Agent on UNIX computers.

CONTROL-M / Agent for Unix is a component of the CONTROL-M scheduling solution. The integration of these products is illustrated in Figure 1 on page 16.

CONTROL-M / Server handles production control and scheduling, and submits and tracks jobs across your network.

CONTROL-M / Agent submits jobs for execution on the Agent computer, monitors the jobs, and performs post-processing analysis of output files. The completion status of jobs and the results of post-processing analysis are transmitted back to CONTROL-M / Server.

Other CONTROL-M components are described in the documents listed in “Related Publications” on page 12.

Figure 1 CONTROL-M Scheduling Solution

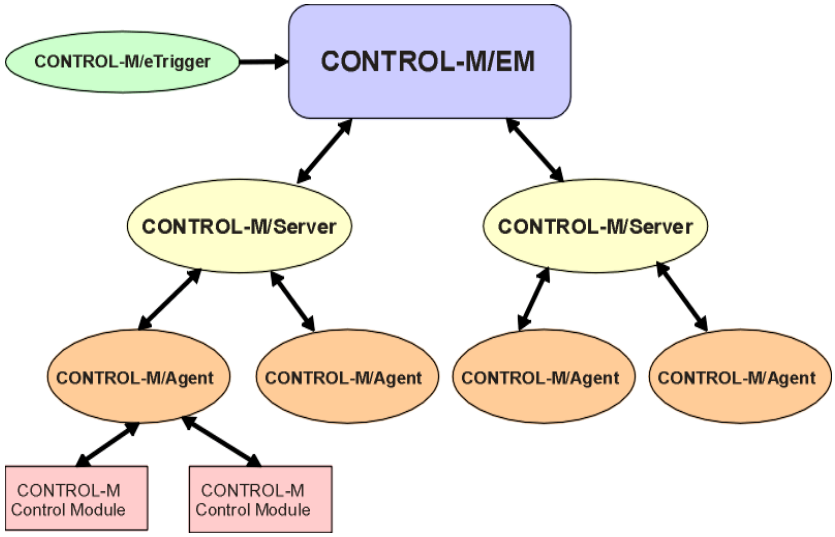


Table 1 CONTROL-M Products

Product	Description
CONTROL-M/EM	GUI-based control center from which you can manage all scheduling and workflow activities.
CONTROL-M/Server	Engine (on a UNIX or Microsoft Windows computer) used to drive scheduling in a data center. Each CONTROL-M / Server can manage multiple CONTROL-M / Agents on various computers.
CONTROL-M/Agent	Software responsible for job submission and execution. An Agent must exist on each computer that is used to execute CONTROL-M jobs.
CONTROL-M/eTrigger	Web-based product that creates and runs jobs under CONTROL-M using input from a web page.
CONTROL-M/Control Modules	CONTROL-M Control Modules enable CONTROL-M / Agents to interface with other applications (for example SAP and Oracle Applications).

CONTROL-M/Agent Functions

Job handling requests managed by CONTROL-M / Agent can consist of any of the following:

- instructions to submit a job on the Agent computer

- requests for information about jobs on the Agent computer that have been submitted, are currently executing, or have recently completed
- requests to view or edit job script statements
- requests to view job output (sysout) or job documentation
- requests to kill jobs that are currently executing

In addition, CONTROL-M / Agent can handle job output (sysout) and issue Shout messages according to job processing parameters that are supplied with a job submission request.

CONTROL-M/Agent Concepts

Additional information about CONTROL-M / Agent is contained in the following table.

Table 2 CONTROL-M/Agent Information

Item	Description
Installation	Information about installing and upgrading CONTROL-M / Agent for UNIX is described in the *CONTROL-M Installation Guide*.
Configuration Parameters	CONTROL-M / Agent configuration parameters are assigned values during the installation procedure. ■ Table 16 on page 65 lists these parameters and their default values. ■ These parameters are stored in the **CONFIG.dat** file. ■ You can use the ctmagcfg utility (see page 41) to modify these parameters after installation.
Command Line Utilities	Many CONTROL-M tasks can be implemented from a UNIX command line using utilities. For more information, see Chapter 4, "Utilities.".
Control Modules	A Control Module (CM) is a DLL component (Microsoft Windows) or shared object (UNIX) that enables CONTROL-M / Agent to interface with other applications in your data center. For example, the CM for SAP interfaces between CONTROL-M / Agent and SAP. ■ The CONTROL-M / Agent installation procedure installs a Control Module for UNIX operating systems and assigns values to its configuration parameters. ■ Table 17 on page 67 lists these parameters and their default values. ■ These parameters are stored in the **OS.dat** file. ■ You can use the "ctmunixcfg Utility" (see page 46) to modify most of these parameters after installation.

Server and Agent Communication

More than one CONTROL-M/Agent can reside on a computer. This enables more than one CONTROL-M/Server to communicate with different Agents on the same computer. For example, a CONTROL-M/Server can submit a job to an Agent in a test environment while another CONTROL-M/Server can send an accounts payable job to the same computer, using a different Agent.

NOTE

If more than one agents is installed, each agent must have different server_to_agent ports. For more information, refer to the "Agent Configuration Utility" on page 41.

CONTROL-M/Agent can be configured to work with a primary and backup CONTROL-M/Servers. In such a configuration, if the primary server fails and defers to a backup server, the agent will defer to the same backup server.

Connecting 2 CONTROL-M/Agents on the same host to a CONTROL-M/Server

UNIX machines support the configuration of more than one agent on the same machine, connected to the same CONTROL-M/Server, but using different server_to_agent ports.

To configure your system:

1. Ensure that the values in the server_to_agent port field of the Agent Configuration utility are different for each agent. For more information, refer to "Agent Configuration Utility" on page 41.

2. Define the second agent on CONTROL-M/Server using a different logical name and the port from step 1.

3. Log on, as root, to the machine on which the CONTROL-M/Server to which you want to add the agent, is installed.

4. Open the /etc/hosts file and add the following line

   <IP address of the Agent computer> <logical name>

5. On the second agent, open the ctm/data/CONFIG.dat file in a text editor.

6. Change the value of the LOCALHOST field to the logical name entered in step 2.

EXAMPLE

If two agents are installed on a machine called appserver with the IP address 11.22.33.44:

1. On the server, define the second agent as appserver2.
2. Add the line 11.22.33.44 appserver2 to the /etc/hosts file.
3. Change the LOCALHOST field for the second agent to **appserver2**.

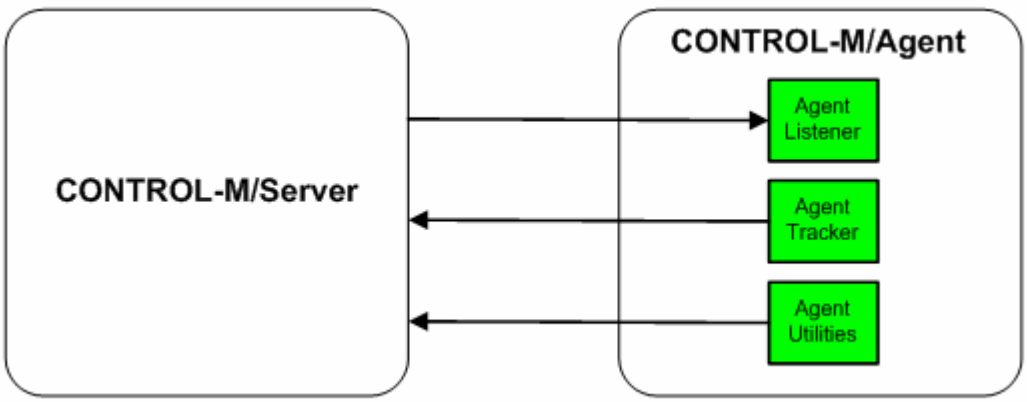
Agent to Server Connection Models

There are two possible models to guide you how you can connect to CONTROL-M/Server.

- Transient connection - default model used with new and upgrade installations. For more information, see "Transient connection model" on page 20.
- Persistent connection model - optional model with improved connectivity between the Server and Agent. For more information, see "Persistent Connection Model" on page 21

For more information about the connection model parameters, see Table 7 on page 43.

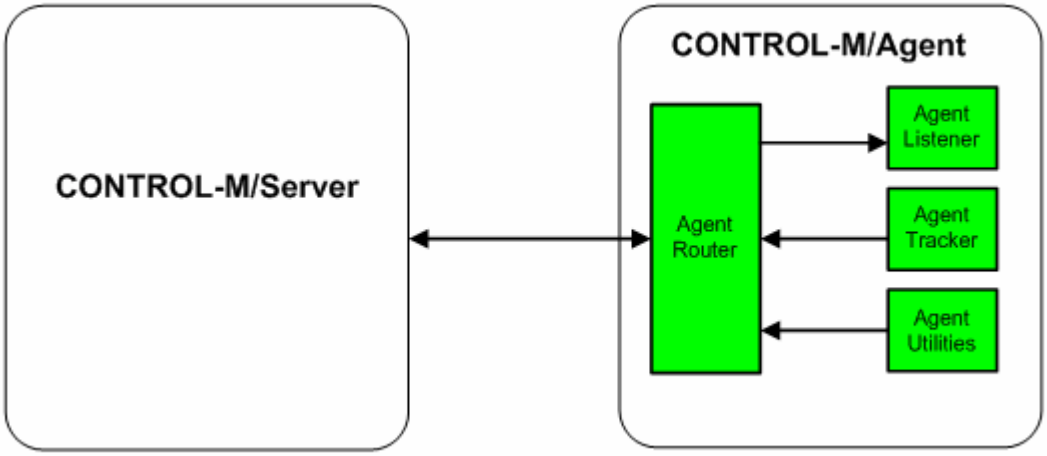
Transient connection model



In the transient connection model, CONTROL-M/Server initiates a connection with the CONTROL-M/Agent Listener process to submit jobs and other action requests. In contrast, the Agent Tracker and Agent Utilities only open a connection to CONTROL-M/Server when they need it. Once the purpose for opening these connections is finished, the connection terminates.

However, if CONTROL-M/Server sits behind a firewall, the Agent Tracker and Agent Utilities are not able to open a connection to the server. As a result, Agent Utilities cannot be run and job statuses are updated only upon server request, approximately once every 15 minutes.

Persistent Connection Model



In the persistent connection model, the connection between the server and agent is constant and can be initiated by both the server and agent. Upon startup of the Agent, the Agent Router process is started and acts as a broker between the other Agent components and the Server.

The Agent Router process allows CONTROL-M/Server to maintain a constant connection with the Agent. However, when CONTROL-M/Server sits behind a firewall, the Agent Router cannot initiate the connection with the server. Once the server creates the connection to the Agent Router, the Agent Tracker and Agent Utilities processes use this connection to communicate freely with the Server.

Chapter 2

Implementing CONTROL-M/Agent

The procedures and facilities described in this chapter enable you to keep CONTROL-M / Agent running efficiently.

The following topics are discussed in this chapter:

Starting and Stopping CONTROL-M/Agent

1. Log on to the Agent platform as **root**.
2. Enter the following command from the Agent directory or run the command interactively:

```
<agent_path>ctm/scripts/start-ag –u <agent_username> -p ALL :
```

Manually Stopping CONTROL-M/Agent and Tracker

1. Log on to the Agent platform as **root**.
2. Enter the following command from the Agent directory or run the command interactively. This command will shut down all CONTROL-M/Agent processes.

```
<agent_path>ctm/scripts/shut-ag –u <agent_username> -p ALL :
```

Verifying Communication with the Server

BMC Software recommends that you verify the ability of the Agent computer to communicate with the primary Server computer and with all other authorized Server host computers.

Generating the Communication Diagnostic Report

CONTROL-M/Agent includes a diagnostic program that checks parameters and environmental conditions relevant to communication between the Agent and Server computers. This program is typically used at the request of Technical Support to determine the cause of a communication problem.

How to Generate the Communication Diagnostic Report

1 Navigate to the directory in which CONTROL-M/Agent is installed.

2 Enter the `ag_diag_comm` command. After several seconds, the CONTROL-M/Agent Communication Diagnostic Report is displayed.

```
CONTROL-M/Agent Communication Diagnostic Report
-----------------------------------------------

 Agent User Name                : ag620
 Agent Directory                : /home/ag620/ctm
 Agent Platform Architecture    : AIX
 Agent Version                  : 6.2.01.000
 Agent Host Name                : appsrv002
 Server Host Name               : sunsrv001
 Authorized Servers Host Names  : sunsrv001
 Server-to-Agent Port Number    : 7006
 Agent-to-Server Port Number    : 7005
 Server-Agent Protocol Version  : 06
 Server-Agent Comm. Protocol    : TCP
 Server-Agent Connection mode   : Transient
 Unix Ping to Server Platform   : Succeeded
 Agent Ping to Control-M/Server : Succeeded

 Agent processes status
 ======================
 Agent Router                   : Not running
 Agent Listener                 : Running (42762)
 Agent Tracker                  : Running (51208)
```

Language Capabilities

This section provides information about CONTROL-M / Agent language support.

Western European Languages

CONTROL-M / EM, CONTROL-M / Desktop, CONTROL-M / Server, CONTROL-M / Agent, and CONTROL-M / eTrigger, support Western European language characters (the Latin-1 character set). These products can accept characters in English, German, Spanish, and French from the Latin-1 character set (ISO 8859-1) in almost all text fields and parameters.

For additional information, see the following guides.

Task	Topic and Guide
Indications of which parameters do *not* support Western European Language special characters	Individual parameter descriptions in the *CONTROL-M/Enterprise Manager Parameter and Variable Reference Guide*
Indications of which CONTROL-M / Server and CONTROL-M / Agent utilities do *not* support Western European Language special characters	Individual utility descriptions in the *CONTROL-M/Server Administrator Guide* and the *CONTROL-M/Agent Administrator Guide*
Indications of which dialog boxes support Western European Language special characters	Individual descriptions of the various dialog boxes in the *CONTROL-M/Enterprise Manager User Guide* and the *CONTROL-M/Desktop User Guide*
Language configuration instructions	*CONTROL-M Installation Guide*
Upgrade and migration instructions	*CONTROL-M Upgrade Guide*
CONTROL-M / eTrigger customization instructions	"Use Locale" topic in Chapter 4 of the *CONTROL-M/eTrigger Administrator Guide*

Japanese

CONTROL-M can run on Japanese operating systems.

The CONTROL-M components, such as the EM GUI and CONTROL-M / Desktop, do not accept Japanese characters in any free text fields or parameters and display values only in English. For example, Japanese job sysouts do not display correctly. Therefore, in these cases, job sysout analysis is not possible.

No additional customization is necessary after installation or upgrade to run CONTROL-M components with Japanese operating systems.

Writing Scripts

When writing a shell script to be run as a CONTROL-M job on an Agent computer, the following factors must be considered:

- Specification of the shell type under which the script will run.
- Run-time environmental factors affecting execution of the script.
- Usage of the **On Statement/Code** job processing parameter.

BMC Software recommends that you run each script manually to validate the script syntax before running the script under CONTROL-M.

NOTE

Command type jobs must be in Bourne shell syntax only.

Specifying the Shell Type

To enable CONTROL-M to recognize the script shell type, specify the shell path (as listed in Table 3) on the first line of the script:

```
#! <shell path>
```

Table 3 Shell Paths

Shell Type	Shell Path	Default Switch	Other Switches
Bourne	**/bin/sh**	-x	-v/n
Korn	**/bin/ksh**	-x	-v/n
csh	**/bin/csh**	-v	
tcsh	**/bin/tcsh**	-v	

NOTE

BMC Software recommends that you not add a switch to the above syntax. To toggle the switches for the scripts, use the Agent Configuration Utility.

Table 4 describes the affect Shell parameter switches have on CONTROL-M / Agent processing.

Table 4 Shell Parameters

Parameter	Description
-x	This parameter causes CONTROL-M / Agent to submit the script as is. The script runs under the specified shell and prints commands and related arguments as they are executed. **Note**: In the sysout file, the command arguments contain the value of the variable and not the variable name. Each command is prefixed by the '+' sign. This sign is later used during an On statement post-processing phase of the jobs output to distinguish between the different commands and their output.
-v	This parameter causes CONTROL-M / Agent to parse the original script to a temporary script. The script commands are appended with an identifying string. This temporary script is then executed, where the *-v* switch causes the shell to print each command before its output. The added identifying string is later used during an On statement post-processing phase of the job's output to distinguish between commands and their output.
n	This CONTROL-M / Agent-specific flag is used to indicate that the script should be executed as is and no commands will be included in the job's output. As a result no On-statement processing is possible.
For more information about the different flags, refer to the example on page 29.	

NOTE

Arguments specified after the shell name are ignored by CONTROL-M/Agent with the following exception: **-x** is supported when running a script under the Bourne shell or Korn shell. If **-x** is specified as an argument after the shell name, it overrides any option set in the CTM_PRM_SH_FLAGS or CTM_PRM_KSH_FLAGS parameters.

Example

The following script uses the app, dbadmin, and stx111 parameters. The app parameter sets an environment variable. The script uses the dbadmin and stx111 parameters to call a utility that performs an action. The output of the job varies depending on the shell flag.

EXAMPLE

```
#!/bin/sh
DBNAME=$1
export DBNAME
dbrefresh -U $2 -P $3
exit $?
```

- If the *-x* flag was set when running the sample script, the job produces the following output.

```
DBNAME=app
+ export DBNAME
+ dbrefresh -U dbadmin -P stx111
DB refreshed
+ exit 0
```

- If the *-v* flag was set when running the sample script, the job produces the following output.

```
#! /bin/sh -v
CTM_RSVD=
CTM_RSVD_START=
CTM_RSVD_END=
CTM0='/home2/ag620/refreshDB.sh'
CTM00=$0
DBNAME=$1 $CTM_RSVD
export DBNAME $CTM_RSVD
dbrefresh -U $2 -P $3 $CTM_RSVD
DB refreshed
exit $? $CTM_RSVD
```

- If the *n* flag was set when running the sample script, the job produces the following output.

```
DB refreshed
```

Support for REXX-Language Scripts

On certain computers, job scripts to be run under CONTROL-M can be written using the REXX shell language. REXX job scripts are supported on the following computers: AIX, SunOS, and Solaris.

To activate a REXX script, the REXX product must first be installed on the Agent platform. The first line of the REXX script must specify the full path under which REXX is installed.

EXAMPLE

```
#!/usr/local/bin/rxx
```

Run-Time Environment

CONTROL-M runs a job script under the environment specified for the job owner (that is, the user specified in the **Owner** parameter). The environment affects these factors in the execution of the script:

- User log on process
- Shell script startup process
- Working directory

Each of these factors is described below.

User Log on Process

As jobs are submitted for execution, CONTROL-M / Agent logs on as the user and executes the job (the shell script) using the following command:

```
su - <owner> -c <script name>
```

During the logon process, the user environment is set according to the shell type specified in / **etc/passwd**.

Shell Script Startup Process

The startup process for running the script depends upon the type of shell under which the script will run.

- When a csh or tcsh script is run, the **.cshrc** file of the job owner is executed as part of the startup process for the script.
- For all other shell types, the **.profile** file of the job owner is executed as part of the startup process for the script.

NOTE

The **.login** file is not executed as part of the startup process.

When CONTROL-M executes job scripts, there is no terminal associated with the job. Therefore, do not use commands in a script that query terminal characteristics or take input from a terminal.

The shell script startup process sets the environment variables that will be available when the script is run. The **#!** statement (see "Specifying the Shell Type" on page 28) indicates the shell under which the script is intended to run.

Working Directory

The working directory at the time the script runs is initially set to the home directory of the job owner (the home directory for each user is set by the UNIX administrator in **/etc/passwd**).

When writing scripts that access files, the file name in the script should be specified with a full path or with a path relative to the home directory of the job owner.

On Statement/Code Parameter

The following items describe how the **On Statement/Code** job processing parameter interprets script lines.

- **Type of Script Statement**
  Depending on the shell used, CONTROL-M / Agent does not process certain types of script statements for comparison with the text specified in the **Stmt** subparameter of the **On Statement/Code** parameter. Therefore, text contained in these script statements should not be specified in the **Stmt** subparameter:

  — For a Bourne shell, text in **if, for, while**, and **case** statements.

  — For a csh shell, text in **if** statements.

EXAMPLE

No part of the following script line should be used in the **Stmt** subparameter of the **On Statement/Code** parameter:

```
if [ 'baseline' - eq 0 ]; then
```

- **Continuation Lines**
  CONTROL-M / Agent does not process continuation lines for comparison with text specified in the **Stmt** subparameter of the **On Statement/Code** parameter. Therefore, text on a continuation line in a script should not be specified in the **Stmt** subparameter.

- **Length of Script Statement**
  CONTROL-M / Agent only processes the first 132 characters of a script statement for comparison with the text specified in the **Stmt** subparameter of the **On Statement/Code** parameter. Therefore, text that occurs after the first 132 characters of a script statement should not be specified in the **Stmt** subparameter.

- **HERE Documents**
  The term HERE document refers to lines of text in a script that are passed to a command as input, but are not passed to the shell. The current version of CONTROL-M / Agent does not support the **On Statement/Code** job processing parameter for HERE documents.

EXAMPLE

In the following script, **line 1** and **line 2** of a **HERE** document are passed to the specified **cat** command:

```
cat > /tmp/junk << EOF_EOF
line 1
line 2
EOF_EOF
echo "DONE"
```

For more information about the **On Statement/Code** parameter, see Chapter 7 of the *CONTROL-M Job Parameter and Variable Reference Guide*. Job processing parameters are described in Chapter 5 of the *CONTROL-M/Enterprise Manager User Guide*.

Utilization of Exit Codes by CONTROL-M

You can cause CONTROL-M to distinguish between different exit codes by using the following expression in the **Code** subparameter of the **On Statement/Code** job processing parameter:

```
COMPSTAT=<value>
```

<value> is the exit code of the script.

EXAMPLE

Assume that a script exits with an exit code of 5.
This condition can be detected by defining the following **On Statement/Code** parameters:

```
Stmt: *
Code: COMPSTAT=5
```

Use of the $0 Reserved Variable

The **$0** reserved variable can be used in a script to retrieve the name of the script. This variable is automatically replaced by a file name before the script is run. When a script runs as a CONTROL-M job using the *–v* flag (see Specifying the Shell Type), it is parsed into a temporary script so any reference to $0 in the script is resolved to the temporary script name. The name of the original script is saved in the CTM0 variable. This differentiates between a script run from the command line run and a script run from a CONTROL-M job.

To resolve this problem you need to set the Translate_$0 flag using the Agent Configuration Utility. Setting the flag causes CONTROL-M / Agent to replace any occurrence of $0 in the original script with $CTM0 when it parses the original script to the temporary script. This will restore the original functionality of the script as if it ran from the command line.

The following example shows the dollar0.sh script, which is supposed to print out the script name.

EXAMPLE

```
#!/bin/sh
echo $0
```

- When the script runs as part of a CONTROL-M job using the -v flag, the name of the temporary script is printed.

```
#! /bin/sh -v
CTM_RSVD=
CTM_RSVD_START=
CTM_RSVD_END=
CTM0='/home/ag620/dollar0.sh'
CTM00=$0
echo $0 $CTM_RSVD
/tmp/ctm/CM_SH.11939
```

- When the script runs in a CONTROL-M job using the *-v* flag *and* the Translate_$0 flag is set, the name of the original script is printed.

```
#! /bin/sh -v
CTM_RSVD=
CTM_RSVD_START=
CTM_RSVD_END=
CTM0='/home/ag620/dollar0.sh'
CTM00=$0
echo $CTM0 $CTM_RSVD
/home/ag6220/dollar0.sh
```

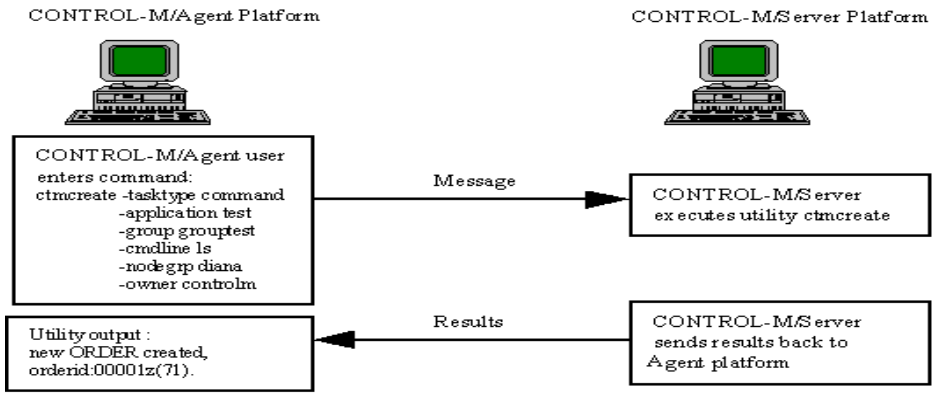
Utilities

The utilities in Table 6 on page 40 can be invoked on an Agent platform by the user or by a batch job running on the Agent platform.

Some of these utilities create jobs in the CONTROL-M/Server Active Jobs file. Their output is sent to the Agent computer. These utilities are described in Chapter 2 of the *CONTROL-M/Server Administrator Guide*. Their processing workflow is illustrated in Figure 2.

Most utilities that create a job in the CONTROL-M/Server Active Jobs file are interactive when invoked from the Server platform, but not interactive when invoked from the Agent platform. When invoked from the Agent, they must be invoked with all the required parameters.

Figure 2 CONTROL-M/Server Utility Workflow



NOTE

If the primary CONTROL-M/Server does not respond to a CONTROL-M/Agent request to execute a utility (other than ag_ping), the request is automatically redirected to the first non-primary Server listed in the Authorized CONTROL-M/Server Hosts parameter. If the redirection is successful, that Agent continues to work with the replacement Server.

Timeout Intervals

The Agent-to-Server and Server-to-Agent communication timeout intervals are described on page 41. If the Agent requests a utility that runs on the Server, or the Server requests the Agent to order a job, and there is no response within the timeout interval, the requested action will fail.

You can increase these timeout intervals by using the **ctmagcfg** utility described on page 41. However, increasing these timeout intervals tends to reduce CONTROL-M / Agent performance.

Specifying Utility Parameters

The command used to invoke a CONTROL-M utility is normally specified with all relevant parameters.

NOTE

When a utility is invoked from the command line, a maximum of 1000 characters can be entered on the command line. The number of characters can be increased by using the **-input_file** parameter.

The utilities in Table 5 enable you to place utility parameters in an input file. The **-input_file** parameter identifies the file that contains parameters for the utility. In this file, each parameter and its values (if any) are on a separate line with the same syntax they would have on a command line.

Using the **-input_file** parameter enables you to

- prepare and save files of utility parameters that can be reused
- exceed the number of characters allowed in the command line

Table 5 Utilities That Support the -input_file Parameter

ctmcontb	ctmkilljob	ctmvar
ctmcreate	ctmorder	ecaqrtab
ctmdefine	ctmshout	

Example

```
ctmcreate -input_file <file_name>
```

where file_name is the name of a file that contains the ctmcreate parameters. For example:

```
-tasktype command
-cmdline ls
```

Utility Parameter Syntax

Some utilities require special formatting for transmission to the Server platform. See "Special Utility Parameter Formats"below.

Values for utility parameters must not contain the apostrophe or single quote character.

When invoking these utilities from CONTROL-M/Agent, the presence of a special character in the argument values may cause problems. The following command contains a back-slash before the string DELETEME:

```
ctmcreate -tasktype COMMAND -jobname servertest -cmdline "ctmvar -action set -var
'%%#\DELETEME' -varexpr to_be_deleted"
```

When this command is invoked from CONTROL-M/Agent, the back-slash before DELETEME may be "eaten" by the shell. To avoid this problem, add a back-slash before the special character that causes the problem (in this case, the original back-slash).

When invoking the ctmcreate utility or ctmdefine utility from CONTROL-M/Agent with a **date_ref** of **$$$$**, put a back-slash before each **$** as shown here:

```
ctmcreate -tasktype command -cmdline ls -incond a '\$\$\$\$' AND ...
```

Special Utility Parameter Formats

Commands invoked from Unix Agent platforms are embedded in double quotes when sent to the Server platform. Therefore, use single quotes for command elements that must be within quotation marks. For example:

```
ctmcreate ...-cmdline "ls -l '$HOME'"
```

For a description of syntax rules when invoking the ctmcreate utility, see "ctmcreate" in Chapter 2, "Utilities," of the *CONTROL-M/Server for Unix Administrator Guide*.

Directing Output From Utilities

Some utilities generate reports that can be directed to a file. These utilities have <Output_ parameters>.

- If output parameters are specified, the utility output is directed to a file on the Server platform.
- If output parameters are not specified, the output is routed to the default output device.

Output can be redirected to the Agent platform by specifying a full path name of the file after the redirection (>) character.

Enabling Other Users to Run Agent Utilities

To enable users other than the CONTROL-M/Agent user to invoke these utilities from CONTROL-M/Agent, add the following environment variables to **.cshrc** or **.profile**:

Add to .cshrc :

```
setenv IOA_HOME <ctmagent>/ctm
setenv CONTROLM <ctmagent>/ctm

[for AIX]
setenv LIBPATH "$LIBPATH : <ctmagent>/ctm/exe"

[for HP-UX]
setenv SHLIB_PATH "$SHLIB_PATH : <ctmagent>/ctm/exe"

[for other platforms]
setenv LD_LIBRARY_PATH "$LD_LIBRARY_PATH :     <ctmagent>/ctm/exe"
```

Add to .profile :

```
export IOA_HOME="<ctmagent>/ctm"
set CONTROLM=<ctmagent>/ctm/
export CONTROLM

[for AIX]
export  LIBPATH="$LIBPATH : <ctmagent>/ctm/exe"

[for HP-UX]
export SHLIB_PATH="$SHLIB_PATH : <ctmagent>/ctm/exe"

[for other platforms]
export LD_LIBRARY_PATH="$LD_LIBRARY_PATH :     <ctmagent>/ctm/exe"
```

Utility Descriptions

Table 6 lists utilities that can be invoked from CONTROL-M/Agent. Some of these utilities can use the **-input_file** parameter. For more information, see "Specifying Utility Parameters" on page 36.

Utilities that are not described in this chapter are described in Chapter 2 of the *CONTROL-M/Server Administrator Guide*.

Table 6 CONTROL-M/Agent Utilities (Part 1 of 2)

Utility	Description
ag_diag_comm	Agent communication diagnostic utility.
ag_ping	Verifies that CONTROL-M/Server is active on the Server platform that is connected to the Agent platform. For more information, see "ag_ping Utility" on page 41.
ctmag	Interactively configures parameters in the config table. For more information, see "Agent Configuration Utility" on page 41.
ctmcontb	Performs operations on the Prerequisite Conditions table.
ctmcreate	Creates a job in the CONTROL-M/Server Active Jobs file.
ctmdefine	Defines a job in the CONTROL-M/Server database.
ctmfw	Detects completion of file transfer activity by checking file size. Signals when desired file size is achieved. Can be invoked from the **<CONTROL-M/Agent>/exe** directory. For more information, see page 49**.**
ctmkilljob	Terminates a CONTROL-M job and its associated processes.
ctmloadset	Updates a Quantitative resource in the Resources table. For more information, refer to the administrator guide for CONTROL-M/Server.
ctmnodegrp	View and maintain node groups. For more information, see the utilities chapter of the administrator guide for CONTROL-M/Server.
ctmorder	Orders or forces one or more jobs from a Scheduling table contained in the CONTROL-M/Server database.
	Note: When this utility is invoked from CONTROL-M/Server, parameters can be specified in a fixed order without parameter tags or in any order with tags. When invoked from a CONTROL-M/Agent platform, each parameter must be preceded by its tag.
ctmpsm	Displays the CONTROL-M Active jobs file (AJF). For more information, refer to the administrator guide for CONTROL-M/Server.
ctmshout	Issues a shout message to an indicated destination.
ctmstvar	Displays the current value of an AutoEdit variable or function.
	Note: When this utility is invoked from CONTROL-M/Server, parameters can be specified in a fixed order without parameter tags or in any order with tags. When invoked from a CONTROL-M/Agent platform, each parameter must be preceded by its tag.
ctmudly	Orders jobs for a specific User Daily name.

Table 6 CONTROL-M/Agent Utilities (Part 2 of 2)

Utility	Description
ctmunixcfg	Enables you to view and modify most of the configuration parameters in the **OS.dat** file. For more information, see the "Agent Configuration Utility" on page 41.
ctmvar	Maintains AutoEdit variables.
	Note: The value of parameter **-filename** is the full path and name of a file that is accessible to CONTROL-M/Server.
ecactltb	Lists the status of each Control resource in the Resources table.
ecaqrtab	Performs operations on quantitative resources in the Resources table. Syntax when invoked from the Agent: `ecaqrtab LIST "*" [-OUTPUT <Output>]`
shagent	Shows if an Agent and Tracker are running.
	Note: This utility can be invoked only from CONTROL-M/Agent.

Utilities that are not described on the following pages are executed in CONTROL-M/Server and are described in the utility chapter of the *CONTROL-M/Server Administrator Guide*.

ag_ping Utility

This utility verifies that CONTROL-M/Server is active on the Server platform connected to the Agent platform. From the operating system prompt, specify the following command:

```
ag_ping
```

The utility attempts to communicate with CONTROL-M/Server and indicates whether the attempt succeeded or failed. If the attempt succeeds, you will receive the message:

```
Output:
Server is alive.
Result: Success.
```

Agent Configuration Utility

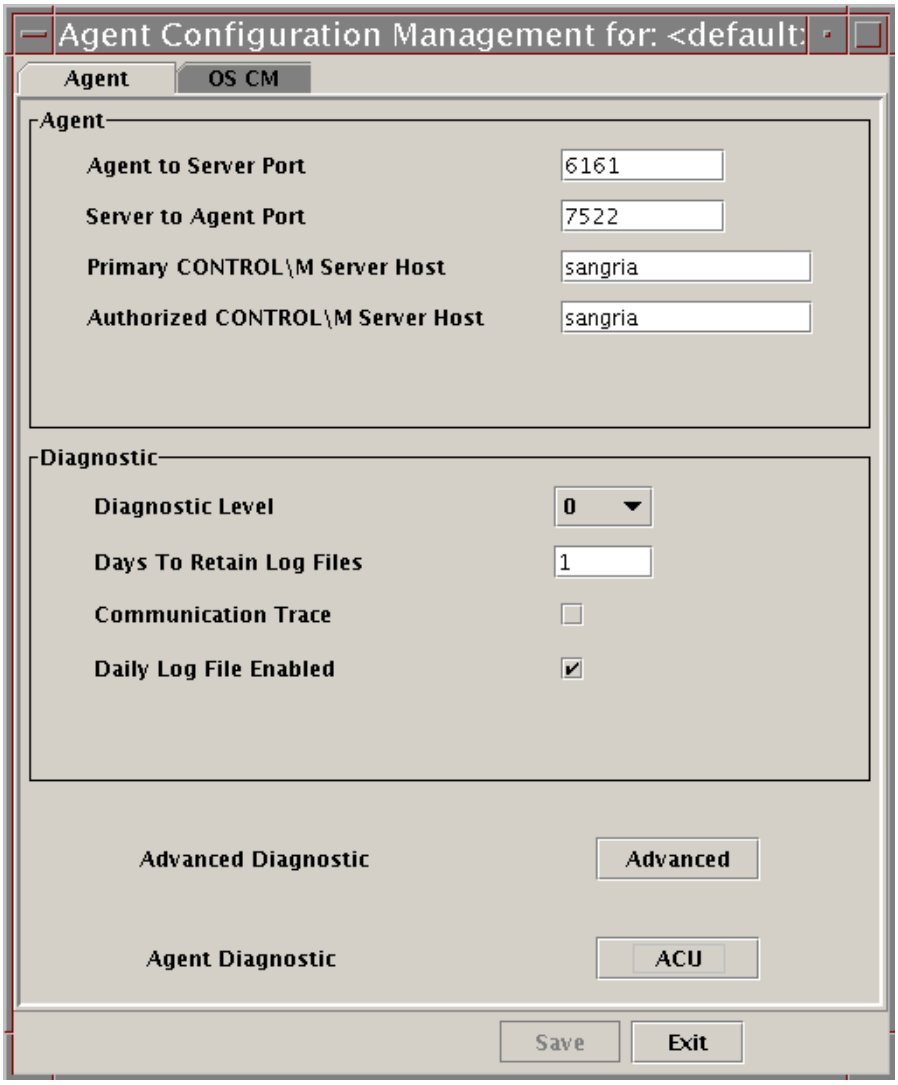
The Agent Configuration (ctmag) utility is a Java application used to maintain CONTROL-M/Agent configuration parameters, and to view and modify most of the operating system parameters. This utility replaces the ctmagcfg and ctmunixcfg utilities.

NOTE

For information about running the Agent Configuration utilities as command line utilities, refer to "Command Line Utilities" on page 47.

There are additional tabs that represent each of the Control Modules (CMs) installed on the agent.

This utility is located at `<installation_dir>\EXE\ctmag` and can be run from the command prompt.



Agent tab

Table 7 lists the parameters displayed in the Agent tab and their descriptions.

Table 7 ctmag Utility Parameters

Parameter	Attributes
Agent-to-Server Port Number	CONTROL-M/Server and CONTROL-M/Agent communicate by means of two TCP/IP ports. This parameter specifies the port in the Server platform that receives data from the Agent platform. The value specified must match the value assigned to the Agent-to-Server Port Number parameter on the Server platform. Verify that the specified port number is not used for any other purpose in the Agent. Specify a numeric value from 1025 to 65535, inclusive. Default: **7005**
Server-to-Agent Port Number	CONTROL-M/Server and CONTROL-M/Agent communicate by means of two TCP/IP ports. This parameter specifies the port in the Server platform that sends data to the Agent platform. The value specified must match the value assigned to the Server-to-Agent Port Number parameter on the Server platform. Verify that the specified port number is not used for any other purpose in the Agent platform. Specify a numeric value from 1025 to 65535, inclusive. Default: **7006**
Primary CONTROL-M/ Server Host	Host computer for the CONTROL-M/Server that handles this Agent. Type the name of the primary CONTROL-M/Server host in the field box or select a host name from the list box. Default: Computer on which the installation was made.
	Note: Do not use a numeric IP address, such as **173.19.6.14,** to specify the name of the server.
Authorized CONTROL-M/ Server Hosts	Names of all CONTROL-M/Servers authorized to handle this Agent (including the primary Server). Specify the host names separated with the "|" symbol. Do not include spaces between host names. The host name of the primary CONTROL-M/Server is automatically included in this field. Default: Name of the primary CONTROL-M/Server host.
	Note: Specify names, such as UNIX_HOSTNAME. Do not specify IP addresses.
Diagnostic Level	Flag that indicates whether to generate diagnostic messages. Valid values: **0**–**4**. Level **0** generates no diagnostics. Level **4** generates maximum diagnostics. This parameter can only be changed after completing the installation.
Communication Trace	Flag that indicates whether to debug communications between CONTROL-M/Agent and CONTROL-M/Server. Valid values: **0** = no. **1** = yes. Default: **0** This parameter can only be changed after completing the installation.
Days To Retain Log Files	Number of days that Agent proclog files are retained. After this period, all Agent proclog files are deleted by the New Day procedure. Default: 1.
Daily Log File Enabled	Indicates whether the **ctmag_<year><month><day>.log** file is generated (**Y**) or not (**N**). Default: **Y**.

NOTE

You can also access the ACU application by clicking on the ACU button located at the bottom of the Agent tab. For more information about ACU, refer to "Agent Check utility" on page 58.

Agent tab (Advanced)

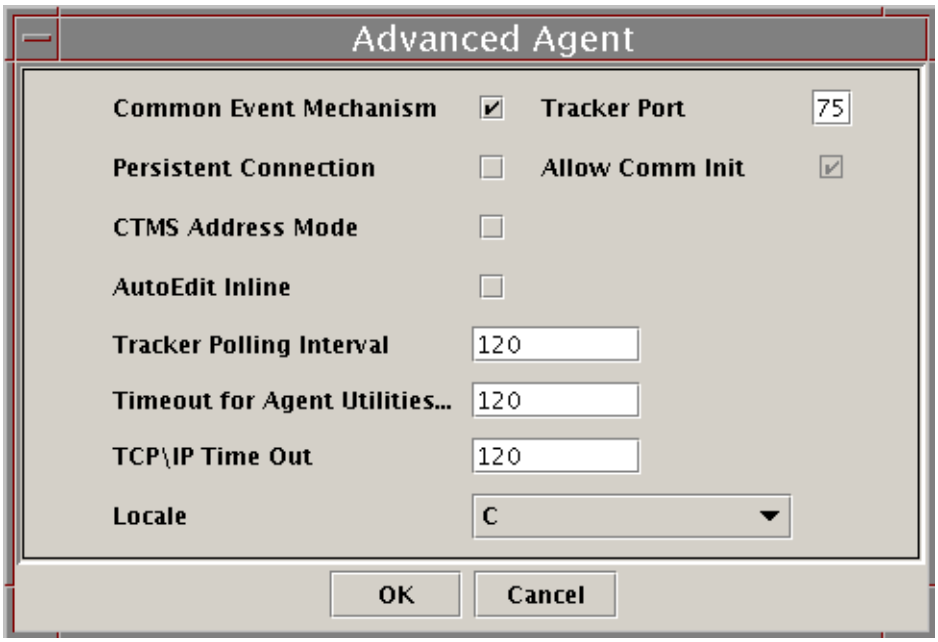


Table 8 lists the parameters found under the Advanced Agent window of the Agent tab and their descriptions.

Table 8 ctmag (Advanced) Utility Parameters (Part 1 of 2)

Parameter	Attributes
TCP/IP Timeout	Communication timeout in seconds. Specify a numeric value greater than or equal to zero. Default: **120**
Tracker Polling Interval	Time in seconds that the Tracker waits after starting the job status checking process before re-starting that process. This parameter can only be changed after completing the installation. Default: **120** (seconds).
AutoEdit Inline	Flag that indicates whether AutoEdit variables defined in a CONTROL-M job are set as Environment variables in the user job environment. This parameter can only be changed after completing the installation.Valid values: **Y** = AutoEdit variables are set. **N** = AutoEdit variables are not set. Default.
CTMS Address Mode	If this parameter is set to **IP**, the IP address instead of the host name is saved in CTMS_HOSTNAME. Use this parameter when CONTROL-M runs on a platform with more than one network card.

Table 8 ctmag (Advanced) Utility Parameters (Part 2 of 2)

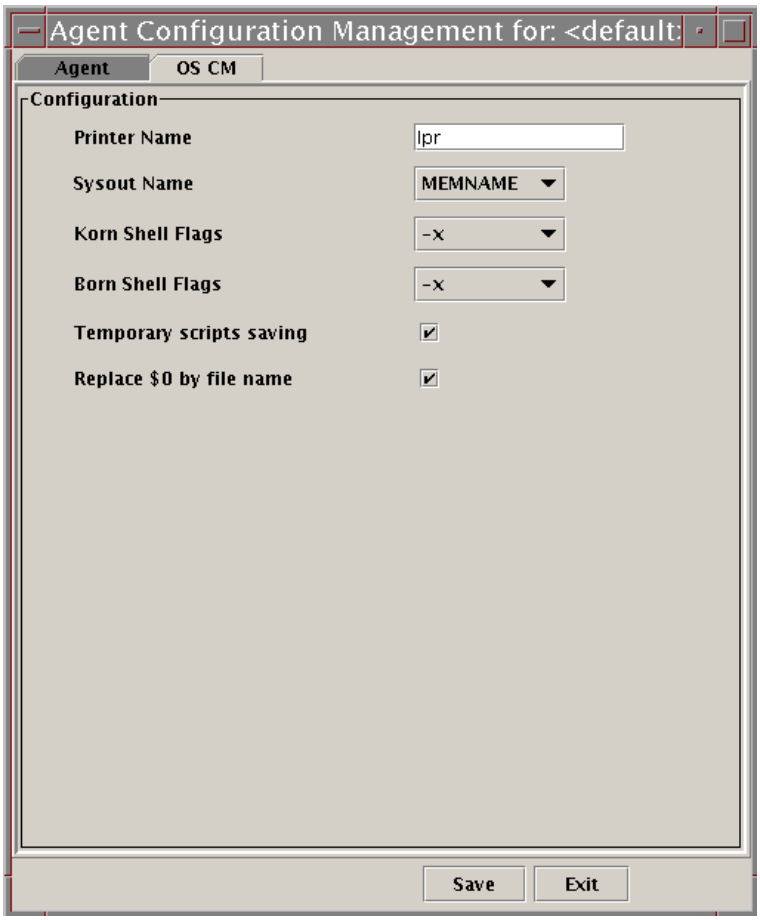
Parameter	Attributes
Timeout for Agent utilities	Maximum time (in seconds) the Agent waits after sending a request to CONTROL-M / Server. This timeout interval should be longer than the TCP / IP Timeout. Recommended value and default: **120**
Common Event Mechanism	Flag for specifying if all control modules, not just the default control module, should be able send messages to the Tracker process without waiting for the tracker polling interval. Messages are sent using the port specified in the Tracker Port parameter to inform the Tracker that a job status changed. Valid values are: **Y** = Use the Common Event Mechanism. Default. **N** = Do not use the Common Event mechanism. Only the default control module can send messages directly to the Tracker process.
Tracker Port	Number of the port for sending messages to the Tracker process when jobs status changes. The tracker event port enables CONTROL-M / Agent to receive updates regarding job status from all Control Modules associated with the current CONTROL-M / Agent. This parameter is used in conjunction with the Common Event Mechanism parameter. Default: **7009**
Persistent Connection	Indicates the connection model between the Agent and Server. Valid values: Y / N. Default: N When set to Y, the Server and Agent maintain a constant connection allowing traffic from all Agent processes to reach the Server, even if the Server is behind a firewall. For more information about connection models, see "Agent to Server Connection Models" on page 20.
Allow_Comm_Init	Determines whether the Agent Router can initiate a session with the server. Valid values: Y / N. Default: Y If the server sits behind a firewall, this parameter should be set to N. For more information about connection models, see "Agent to Server Connection Models" on page 20.
Locale	Determines the regional settings used by the CONTROL-M / Agent account. Supported locale settings are listed in Table 9.

Table 9 Supported Locale Settings

Language	AIX and Solaris	HP-UX
English (USA)	en_US.ISO8859-1	en_US.iso88591
English (British)	en_GB.ISO8859-1	en_GB.iso88591
German	de_DE.ISO8859-1	de_DE.iso88591
French	fr_FR.ISO8859-1	fr_FR.iso88591
Spanish	es_ES.ISO8859-1	es_ES.iso88591

OS CM tab

The parameters in the OS CM tab are described in Table 17 on page 67.



To prevent Bourne and Korn shell script output from being included in system output (sysout), set Korn Shell Flags and Borne Shell Flags to **n**.

CM tab

In the past, to configure a CM, you had to run the management applications for each CM separately. In the Agent Configuration utility, you can manage all of the applications using the relevant CM's tab.

When you install a CM, a *cm_name*.xml data file is placed in the **ctm/data/GUI/** directory. The Agent Configuration utility reads each data file and creates a tab for the CM with each of the CM management applications.

NOTE

CM versions released prior to version 6.2.01 may not have the XML file and the respective CM tab.

For information about each of the CM management applications, refer to the relevant CM documentation.

Command Line Utilities

This section describes the agent configuration command line utilities.

ctmagcfg

This section provides information about running the ctmagcfg utility from the command line.

NOTE

This utility can also be accessed as a Java application. For more information, refer to the "Agent tab" on page 42.

To access the ctmagcfg utility, enter **ctmagcfg** from a command prompt.

```
                Agent Configuration Utility


1)      Agent-to-Server Port Number . . . : [7666]
2)      Server-to-Agent Port Number . . . : [7009]
3)      TCP/IP Timeout. . . . . . . . . . : [120]
     For items 4 and 5 do not use IP address
4)      Primary CONTROL-M/Server Host . . : [martini]
5)      Authorized CONTROL-M/Server Hosts : [martini]
6)      Tracker Polling Interval. . . . . : [120]
7)      Diagnostic Level. . . . . . . . . : [4]
8)      AutoEdit Inline. . . . . . (Y|N): [N]
9)      Comm Trace. . . . . .(0-OFF|1-ON) : [0]
10)     CTMS Address Mode . . . . . (IP|) : []
11)     Days To Retain Log Files. . . . . : [1]
12)     Daily Log File Enabled. . .  (Y|N): [Y]
13)     Timeout for Agent utilities . . . : [120]
14)     Locale. . . . . : [POSIX]
15)     Common Event mechanism. . . . . : [Y]
16)     Tracker Event Port. . . . . : [7011]
17)     Persistent Connection . . . . . : [Y]
18)     Allow Comm Init. . .  . . . . . : [Y]


s)      Save and Restart CONTROL-M/Agent Services

q)      Quit

Please enter your choice:
```

For an explanation of the parameters in the ctmagcfg utility, refer to Table 7 and Table 8.

ctmunixcfg

This section provides information about the ctmunixcfg utility.

NOTE

This utility can also be accessed as a Java application. For more information, refer to the "OS CM tab" on page 46.

To access the ctmunixcfg utility, enter **ctmunixcfg** from a command prompt.

```
                    OS Configuration Utility


1)      Sysout Name. . .(MEMNAME|JOBNAME). : [MEMNAME]
2)      Printer Name . . . . . . . . . . . : [lpr]
3)      Korn Shell Flags. . . .. .(-x|-v|n): []
4)      Borne Shell Flags . . .  .(-x|-v|n): []
5)      Temporary scripts saving  (YES|NO) : []
6)      Replace $0 by file name . (Y/N). . : []


s)      Save

q)      Quit


Please enter your choice:
```

For a description of the parameters in the ctmunixcfg utility, refer to Appendix A. If the parameter you want to modify is not listed, see the "Agent Configuration Utility" on page 41.

ctmfw Utility (File Watcher)

The File Watcher utility, **ctmfw**, can be used to detect the

- successful completion of a file transfer activity
- creation of a file
- deletion of a file

ctmfw can be used before activating a job or before performing a task (for example, sending a shout message or adding / deleting conditions) that is dependent upon creation or deletion of a file.

The ctmfw utility runs as a process on a client machine. The process waits for the creation or deletion of specified files.

- For a file transfer activity, when the file is detected, the job continues to monitor the size of the file. When the file reaches a specified minimum size and does not increase in size for a specified period of time, the File Watcher utility either completes with a status of **OK** or executes a specified **DO** action. **DO** actions can consist of adding or deleting conditions or executing a command.

- For file creation, file size is ignored if a wildcard is specified as part of the filename unless the mon_size_wildcard parameter is set to **Y**.
- For file deletion, ctmfw must first detect the existence of the file before it can detect its deletion.

The ctmfw utility can also be run from the command line, or be invoked to detect either a single file or multiple files.

Usage as a Utility

When running as a utility, ctmfw is invoked from the command line. Rules can be provided on the command line or by a rule file.

To watch a single file:

The syntax of the ctmfw utility is:

```
ctmfw FILE (absolute path)
      < mode (CREATE|DELETE)>  Default: CREATE
      < minimum detected size <number>
       [' '|Bytes(B)|Kilo(K)|Mega(M)|Giga(G)] >Default:0
      < interval between file search (seconds) > Default: 60sec
       < interval between filesize comparison iterations (seconds) >
       Default: 10sec
       < number of iterations while the size is static > Default: 3 iterations
        < time limit for the process (minutes). Default: 0 (no time limit)
         Effective while the file does not exists or,
         the file size is static and the minimum size
         was not reached >
        < monitor file size when wildcard is used > Default: N
         < starting  time for detecting files (HHMM or YYYYMMDDHHMM >
         Default: NOW
        < absolute stop time (HHMM or YYYYMMDDHHMM > Default: 0 ( No stop time )
        < minimal age of file ( modified time )
          format:xxxxYxxxxMxxxxDxxxxHxxxxMin > Default: 0
```

The parameters of the ctmfw utility are described in Table 10.

All parameters must be assigned a value, even if that value is zero. If only six values are specified, the default value for **mon_size_wildcard** is used. If five parameters are specified, default values for **wait_time** and **mon_size_wildcard** are used, and so forth.

EXAMPLE

ctmfw /home/watchedfile.txt CREATE 100 10

is resolved using default values for **mon_int**, **min_detect**, **wait_time**, and **mon_size_wildcard** as follows:

ctmfw /home/samplefile.txt CREATE 100 10 10 3 0 N

Table 10 ctmfw – Parameters (Part 1 of 2)

Parameter	Description	
FILE	Path of the file to be detected. The file name can include mask character ***** to represent any number of characters (including no characters) or **?** to represent any one character.	
mode	CREATE	Detects creation of a file. Default. File size is ignored if the filename parameter contains wildcards (unless the **monitor file size when wildcard is used** parameter is set to **Y**).
		Note: If a mask is specified for the filename, and the **monitor file size when wildcard is used** parameter is set to ■ **N**, the ctmfw utility will end **OK** after detection of the first file that matches the specified mask. ■ **Y**, the ctmfw utility will end **OK** after detection of the first file that matches the filename and file size.
		For more information about **monitor file size when wildcard is used**, see below.
	DELETE	Detects deletion of a file. When the ctmfw utility is run in this mode, it first checks for files that match the specified name. After a specified file is detected, the ctmfw utility checks at the specified interval for deletion of that file.
		Note: If a mask is specified as the filename, the ctmfw utility will end successfully only after all detected files that match the specified mask have been deleted.
minimum detected size	Minimum file size in bytes. This parameter is ignored if the **FILE** parameter contains wildcards (unless the **monitor file size when wildcard is used** parameter is set to **Y**) or if the **mode** parameter is set to **DELETE**. Default: **0** (any size detected).	
interval between file searches	Interval between successive attempts to detect the existence / deletion of a file (in seconds). Default: **60**	
interval between filesizecomparison iterations	Interval between attempts to monitor the size of a file after it is detected (in seconds). This parameter is ignored when using wildcards in **FILE** or when using **DELETE** mode. Default: **10**	
number of iterations while size is static	Number of attempts to monitor file size where the size remains static and greater than or equal to **minimum detected size** (indicating successful creation of the file). This parameter is ignored when using wildcards in **FILE** or when using **DELETE** mode. Default: **3**	

Table 10 ctmfw – Parameters (Part 2 of 2)

Parameter	Description
time limit for the process	Maximum time (in minutes) to run the process without detecting the file at its minimum size (**CREATE**) or detecting its deletion (**DELETE**). If the file is not detected / deleted in this specified time frame, the process terminates with an error return code, as described in Table 13. Default: **0** (no time limit).
monitor file size when wildcard is used	Indicates whether file size should be monitored if the filename contains wildcards. This parameter is ignored if the filename does not contain a wildcard. Valid values: **N** – do not monitor file size. Default. **Y** – monitor the file size. If this parameter is set to **Y** and more than one file matches the specified mask, the ctmfw utility randomly selects one matching file, monitors its file size, and ignores all other matching files.
start_time	Indicates an absolute time at which the utility starts monitoring the file. For example, 200512061400, means that at 2 PM on December 6th, 2005 the FileWatcher utility will stop watching the file. Alternatively, you can use the HHMM format, in which case the current date is used.
stop_time	Indicates an absolute time at which the file is no longer watched. For example, 200502061400, would mean that at 2 PM on February 6th, 2005 the FileWatcher utility will stop watching the file. Alternatively, you can use the HHMM format, in which case the current date is used.
minimal_file_age	Indicates the minimum amount of time that must have passed since the file you want to watch was last modified. For example, 2y3d5h means that 2years, 3 days, and 5 hours must pass before the file will be watched. Entering a value of 2H10Min, means that 2 hours and 10 minutes must pass before the file will be detected. This parameter is ignored if the mode parameter is set to **DELETE**. Default: **0**

To watch multiple files:

Use the following command to invoke the ctmfw utility for multiple files:

```
ctmfw -input <rule_file_name>
```

The variable **<rule_file_name>** is the complete path name of the file containing the definitions for each file to be detected.

Sample Rule File

Figure 3 displays a sample rule file. In this sample:

- **#** indicates comments.
- Default values are shown for all global parameters.
- **<action>** refers to any of the actions described in Table 12.

Figure 3 Sample Rule file

```
#*****************************************************************
# Global Parameters
INTERVAL <60>  # Sleep interval (seconds)
MIN_SIZE 4Kilo
MIN_AGE  3M24D4h5min
FROM_TIME <0000>  # Starting time for all files (hhmm)
MIN_SIZE <0>  # Minimum size for all files (bytes)
MIN_DETECT <3>  # Number of iterations for all files
WAIT_TIME <0>  # Time limit for all files (minutes)

# ON_FILEWATCH statements
ON_FILEWATCH <filename>(absolute path) [CREATE/DELETE] [min_size] [min_detect]
[wait_time]
            [start_time] [cyclic_interval] [wildcards] [minimal_file_age]
THEN
<action>
ELSE
<action>
END_ON
#*****************************************************************
```

NOTE

All global parameters must be delimited by the new line character.

The Rules file contains two sections:

- Global parameters, whose default values apply to all the files in the rule file.
- **ON_FILEWATCH** statements identifying which files to detect, specific criteria for each file, and the action to take upon detection or non-detection. Any number of **ON_FILEWATCH** statements can appear in a Rules file.

NOTE

All keywords must be entered in uppercase.

Table 11 Rule file Global Parameters (Part 1 of 2)

Param	Description
INTERVAL	Sleep interval (in seconds) between successive scans for all the files. This parameter replaces individual **sleep_int** and **mon_int** parameters for each file. Default: **60**
MIN_SIZE	Minimum file size in bytes. This parameter is ignored if the **FILE** parameter contains wildcards (unless the **monitor file size when wildcard is used** parameter is set to **Y**) or if the mode parameter is set to **DELETE**. Default: **0** (any size detected).

Table 11 Rule file Global Parameters (Part 2 of 2)

Param	Description
MIN_AGE	Indicates the minimum amount of time that must have passed since the file you want to watch was last modified. For example, 2y3d5h means that 2years, 3 days, and 5 hours must pass before the file will be watched. This parameter is ignored if the mode parameter is set to **DELETE**. Default: **0**
FROM_TIME	Starting time for detecting all the files (default FROM_TIME). Used with WAIT_TIME to identify the time frame for detecting and monitoring the files. This parameter is expressed in 24-hour, hhmm format. Default: **0000** or **Now**
STOP_TIME	Indicates an absolute time at which the file is no longer watched. For example, 200502061400, means that at 2 PM on February 6th, 2005 the FileWatcher utility will stop watching the file. You can also use the HHMM format, which uses the current date, plus the HHMM entered. Default: **0** (meaning, no stop time) **Note**: STOP_TIME can only be used as a global parameter.
CYCLIC_INTERVAL	Indicates the interval between multiple operations of detecting the file (in minutes). This interval must be greater than the value for WAIT_TIME. If the cyclic_interval is 0, only one attempt to detect the file will be performed. Default: **0**
MON_SIZE_WILDCARD	Indicates whether file size should be monitored if the filename contains wildcards. This parameter is ignored if the filename does not contain a wildcard. Valid values: **N** – do not monitor file size. Default. **Y** – monitor the file size. If this parameter is set to **Y** and more than one file matches the specified mask, the ctmfw utility randomly selects one matching file, monitors its file size, and ignores all other matching files.

NOTE

For a description of the ON_FILEWATCH parameters, refer to Table 10 on page 51.

If any mandatory parameter is omitted from a Rules file, the default value for that parameter is used. Parameters entered for **ON_FILEWATCH** statements override the default values. If entered, they must appear in the order shown in Figure 3.

Table 12 ctmfw – Valid Actions

Action	Description
DO_COND <condition name> <condition date> <+|->	Add (+) or delete (-) a condition.
DO_CMD <command>	Execute a valid command under the command interpreter. Full path names are required for files.

Table 12 ctmfw – Valid Actions

Action	Description
DO_OK	Terminate ctmfw with status **OK**. If there is more than one file in the Rule file, the result displayed is that of an AND algorithm.
DO_NOTOK [exit code]	Terminate ctmfw with status **NOTOK**. Exit code is optional and replaces the standard return code, as described in Table 13.

- If the file is detected and the size remains static within the time frame (**CREATE**) or the file has been deleted (**DELETE**), the **DO** commands in the **THEN** block are executed.
- If the file is not detected or deleted within the time frame, the statements following the **ELSE** block are executed.
- ctmfw terminates when all the files in the Rules file have been processed.

NOTE

If an ON_FILEWATCH statement contains a **cyclic_interval** parameter, ctmfw will only stop monitoring a file on a DO_OK or DO_NOTOK action.

Example 1

The ctmfw utility is invoked to watch multiple conditions. The definitions the ctmfw utility uses for watching each file are contained in a rule file.

The following instructions are defined in the Rules file:

- The sleep interval between succeeding scans must be 10 seconds.
- If the ctmfw utility detects that the datafile.txt file in the **/home/controlm** directory is created in the specified time interval, then:
  - the **datafile** condition dated 1 January must be added.
  - The command interpreter must execute the command to move the contents of the file /home/ctm/datafile.txt to /home/ctm/workfile.txt.
- If the ctmfw utility detects that the **datafile.txt** file in the **/home/controlm** directory is not created in the specified time interval, then condition **datafile** dated 1 January must be deleted.
- When the ctmfw utility detects that the **/home/ctm/tempfile.txt** file is deleted, condition **tempfile** dated 1 January must be deleted.

```
#*****************************************************************
INTERVAL 10
ON_FILEWATCH \ctmsrv\datafile.txt CREATE
THEN
DO_COND datafile 0101 +
DO_CMD move \ctm\datafile.txt \ctm\workfile.txt
ELSE
DO_COND datafile 0101 -
END_ON
ON_FILEWATCH \ctm\tempfile.txt DELETE
THEN
DO_COND tempfile 0101 -
END_ON
#*****************************************************************
```

Example 2

A job processing definition is created to implement a File Watcher job. The file must arrive between 19:00 and 22:00, and be created in the **/tmp** directory under the name **trans.dat**. The minimum file size is 100 bytes. The detection process should be performed each minute. The file size monitored every 10 seconds, and the number of intervals where the file size remains static is 5. If the file is not detected by 22:00, an alert should be sent to CONTROL-M/Enterprise Manager.

Parameter	Value
Job Name	FileWatch
Mem Name	FileWatch
Owner	<control_m_user>
From Time	1900
Command line	ctmfw "\tmp\trans.dat" CREATE 100 60 10 5 180
On Statement/Code processing:	
Stmt	*
Code	COMPSTAT=0
Do Cond	file_trans_dat_ok **Date:** ODAT **Sign:** +
Stmt	*
Code	COMPSTAT=1
Do Shout	**To:** CONTROL-M/Enterprise Manager **Text:** "File trans.dat did not arrive on time"

The ctmfw utility processes On Statement/Code combinations in the following order:

1. On Statement/Code combinations related to sysout, for example:
   `ON "*cp aaa bbb*" "*not found*"`

2. On Statement/Code combinations based on the state {OK | NOTOK} of the job, for example:
   `ON "*"  "NOTOK"`

Return Codes

The return codes listed in Table 13 are issued by the ctmfw utility after detecting if a file is created or deleted in the specified time frame.

Table 13 ctmfw – Return Codes

Return Code	Description
0	File successfully created or deleted (file arrived in the specified time frame and file size is above or equal to the minimum specified size).
1	■ Utility failed. For example, because of a syntax error. ■ A DO_NOTOK statement occured, but no user-defined exit code was provided for the DO_NOTOK statement.
7	Indicates that the ctmfw request timed out. That is, the file was not detected in the specified time frame.

shagent Utility

The shagent utility checks that the **p_ctmag** and **p_ctmat** processes are running. It can be invoked only from the CONTROL-M / Agent platform. The utility has no parameters.

If the Persistent Connection parameter is set to Y, the utility verifies that the p_ctmar process is running.

From the operating system prompt, specify the following command:

```
shagent
```

Sample Output

If the Router process is running, output similar to the following is displayed:

```
root         7660   0:00 p_ctmag
root         7644   0:00 p_ctmar
root         7745   0:29 p_ctmat
```

Agent Check utility

The CONTROL-M Agent Check Utility (ACU) is a tool that collects information and diagnostic data about the CONTROL-M Agent installation, execution state, and target environment. The data collected by the ACU is designed to assist CONTROL-M/Agent administrators and BMC Software technical support engineers to troubleshoot, fine-tune, and maintain the CONTROL-M/Agent.

With this tool, you can send generated reports to interested parties using e-mail or FTP to BMC Software (ftp://ftp.bmc.com/incoming). You can print the report to a hierarchical XML file, or save the report as a text file. In addition, you can set the agent debug level and download the most recent agent and CM patches.

This utility is located at **/<*agent_directory*>/ctm/exe/** and can be run from either the command line or as a Java application.

To start the ACU application:

1. Log on to the target CONTROL-M/Agent under the Agent account on which ACU is installed.
2. Set the display variable using the command

   ```
   setenv DISPLAY <host name>:0.0
   ```

3. Enter the `acu_gui` command to run the Java application

   or

   enter `acu` to run the application from the command line.

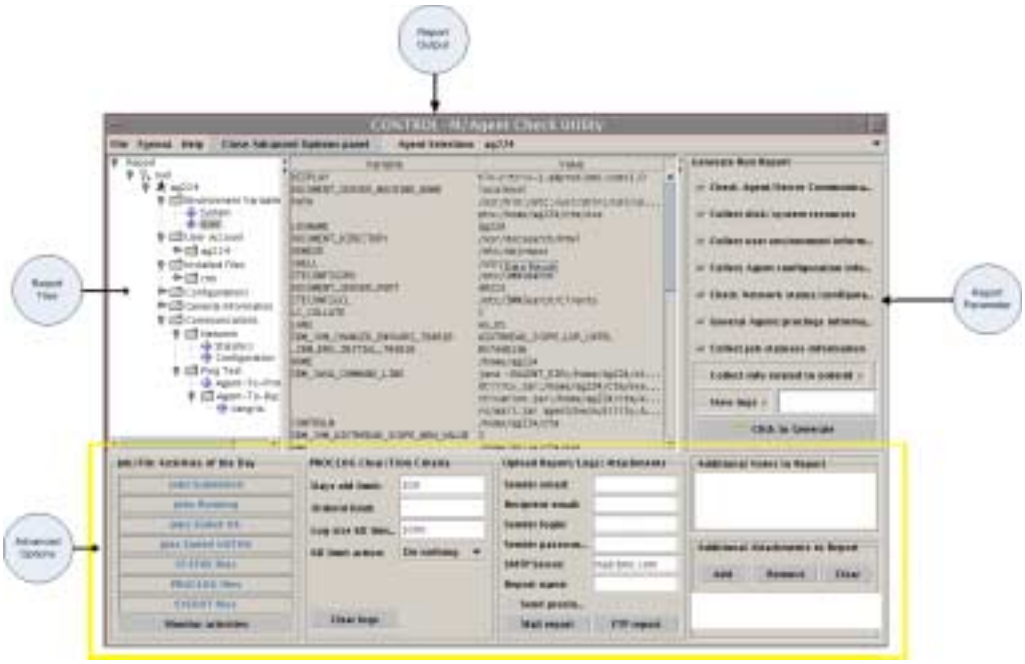
   For more information about running the utility from the command line, refer to "Command line usage" on page 61.

Using ACU

The ACU is divided into the following panels:

- Report tree
- Report output
- Report parameters selection
- Advanced Options panel

Figure 4 Agent Check Utility Window



Report Parameter selection

The right-most panel of the ACU is used to select the data that you want to include in your report.

Table 14 Report Parameters

Parameter Name	Description
Check Agent/Server Communication	Selecting this option runs the ag_ping utility. This utility verifies that CONTROL-M/Server is active on the Server computer connected to the Agent computer.
Check Network status/configuration	Selecting this option provides information about the network configuration and statistics using the netstat and ipconfig commands.
Collect Agent configuration information	Selecting this option provides information about the agent on which you ran the report. This also checks the dll versions and configurations for the agent and CMs installed, and lists the files installed.
Collect disk/system resources	Selecting this option provides system information about the memory in use, free disk space, number of processes running, and so on.
Collect info related to orderid	Selecting this option collects information from files with names containing the specified orderid or information related to the specified orderid.

Table 14 Report Parameters

Parameter Name	Description
Collect job statuses information	Selecting this option provides you with information about the number of jobs processed since the last NewDay procedure, their status, job start time, job end time, and so on.
Collect user environment information	Selecting this option provides Java-applications system information and user specific information, such as root directories, domain, path, and so on.
General Agent/proclogs information	Selecting this option enables you to activate the newly configurated debug level and start collecting the log files from the PROCLOG directory. The logs can then be transferred using e-mail or ftp (if it is a large size zip file) in zipped format.
View logs	Selecting this option opens a browser window from which you can select log files to attach to the generated report. You can then view the content of the attached log files in the output panel.

Report tree

Once you select the areas for which you want to collect data and generate the report, a report tree appears on the left side of the screen. This tree lets you drill down into the various report parameters and select them for viewing in the Output panel.

Output panel

The Output panel is where the data of the selected parameters is displayed. You can save, print, or email the information for further analysis.

Advanced Options

The Advanced options panel enables you to view additional information about the day's activities, define parameters by which the PROCLOG files are saved, enter mail and FTP information for sending and uploading reports, and so on.

Printing

You can print a generated or uploaded report or save it as a text file using the **File > Print** menu option.

Figure 5 Print window

To print the report, click **Print**. To save to a text file, click **Save** and enter a file name in the Save window.

Command line usage

To run ACU as a command line utility, enter the command **acu** with the relevant parameters. When ACU runs as a command line utility, a report of the ACU output is automatically saved to the **/<*agent_directory*>/temp** directory.

Syntax

```
acu   [agent/<name>] [all] [silent] [system] [environment]
        [configuration] [ping] [analyze] [jobinfo] [network]
        [orderid/<orderid num>]
        [ftplogs/<ftp_server> ftpdir/<ftpdir> filename/<filename>]
        [mailreport/<from>/<to>/<id>/<pwd> mailsmtp/<smtp server>
[maillog]]
```

Table 15 lists the command line parameters and their descriptions.

Table 15 Command line parameters

Parameter	Description
agent <name>	Used to specify an agent when more than one agent installation exists. Optional. When used, this parameter must be first.
all	Collects information for all options, including ping, environment, system, network, configuration, and analyze.
analyze	Analyzes the Agent logs and provides information about the possible symptom and its solution.
configuration	information about the agent on which you ran the report.
environment	Collects system information and user-specific information, such as root directories, domain, path, and so on.
filename	The name of the ZIP file you are uploading to the BMC Software ftp site.
ftpdir	Specify the directory into which the ZIP file is uploaded.
ftplogs	Zip the Agent PROCLOG directory and upload it to the BMC ftp server.
jobinfo	Provides information about the number of jobs processed since the last NewDay procedure, their status, job start time, job end time, and so on.
maillog	Zip the Agent PROCLOG directory and attach it to the e-mail.
mailreport	Output the report to a specific e-mail address.
mailsmtp	Specify the smtp server of the corresponding e-mail address.
network	Collects information about the network configuration and statistics.
orderid	Collects information from the files with names containing the specified orderid or information related to the specified orderid.
ping	Checks the connection between CONTROL-M / Agent and CONTROL-M / Server.
silent	Run the utility with no on-screen output.
system	Collects information about system resources.

Command line examples

- In the following example, ACU collects all the information about a specific agent and outputs it to your screen and to the **/<*agent_directory*>/temp** folder.

EXAMPLE

```
>acu all
```

- In the following example, ACU checks the connection between the Agent and Server and collects information about the system, environment, and agent configurations.

EXAMPLE

```
>acu ping network configuration system environment
```

- In the following example, ACU collects all the information about a specific agent and sends the information, including all the logs from the **/<*agent_directory*>/proclog** directory, to a specific e-mail address using the corresponding smtp server.

EXAMPLE

```
>acu all mailreport/user@domain.com/support@bmc.com
mailsmtp/mail.domain.com maillog
```

- In the following example, ACU collects all the information about a specific agent and uploads the information to the BMC ftp site.

EXAMPLE

```
>acu all ftplogs/ftp.bmc.com ftpdir/incoming filename/case_4684
```

Configuration Parameters

The CONTROL-M/Agent configuration parameters in Table 16 are stored in the **CONFIG.dat** file. These parameters can be modified using the "Agent Configuration Utility" on page 41. Some of these parameters are also described in Chapter 5 of the *CONTROL-M/Server for Unix Administrator Guide*.

Table 16 UNIX Agent – CONFIG.dat Parameters (Part 1 of 3)

Parameter	Description
AG_LOG_ON	Indicates if the **ctmag_<year><month><day>.log** file is generated (**Y**) or not (**N**). Default: **Y**.
AGCMNDATA	<Port number>/<Timeout> for the Server-to-Agent port. Port number specifies Agent platform port that receives data from the Server platform. Verify that this port is not used for any other purpose. Must match Server-to-Agent port number in CONTROL-M/Server. Must be between **1024** and **65533** inclusive. Default: **7006**. The timeout value is the **COMTIMOUT** communication job-tracking timeout in seconds. Mandatory. Example: **7006/30**
AGENT_DIR	Location of files used by CONTROL-M/Agent.
ATCMNDATA	<Port number>/<Timeout> for the Agent-to-Server port. Port number specifies the Server platform port that receives data from the Agent platform. Verify that this port is not used for any other purpose. This value must match the Agent-to-Server Port Number in CONTROL-M/Server. The value must be a number between **1024** and **65533** inclusive. Default: **7005**. The Timeout value is the **COMTIMOUT** communication job-tracking timeout in seconds. Mandatory. Example: **7005/30**. **Note:** Increasing the Timeout value reduces Agent performance.
AUTOEDIT_ INLINE	Flag that indicates whether all AutoEdit variables will be set as environment variables in the script. Valid values: **Y** (yes), **N** (no). Default: **N**
CM_APPL_TYPE	Default control module. Default: **OS**
CMLIST	List of Control Modules. For internal use only.

Table 16 UNIX Agent – CONFIG.dat Parameters (Part 2 of 3)

Parameter	Description
COMM_TRACE	Flag indicating whether communication packets that CONTROL-M / Agent sends to and receives from CONTROL-M / Server are written to a file. Valid values: **1** (on), **0** (off). Default: **0** (off). If set to **1**, separate files are created for each session (job, ping, and so forth). This parameter can only be changed after completing the installation.
COMMON_EVENT	Flag specifying if all control modules, not just the default control module, should be able send messages to the Tracker process without waiting for the tracker polling interval. Messages are sent using the port specified in the Tracker Port parameter to inform the Tracker that a job ended. Valid values are: **Y** = Use the Common Event Mechanism. Default. **N** = Do not use the Common Event mechanism. Only the default control module can send messages directly to the Tracker process.
COMMRETSLP	Time in seconds (integer value} to wait between each attempt to attach to the CONTROL-M / Server. Default: **1**
CTMPERMHOSTS	<one or more TCP / IP addresses or DNS names separated by |>. Each value identifies an authorized CONTROL-M / Server host where a backup CONTROL-M / Server is installed. (This parameter was previously called Mirror CONTROL-M / Server Host Name.) Specify this parameter if one or more CONTROL-M / Servers have been designated as backups and can connect to this Agent in case of failover. See the *CONTROL-M/Server Administrator Guide* for information about backup server configuration. Mandatory. At least one primary host name should be specified. Example: **192.138.28.121 | aristo.isr.bmc.com/mybksys1 | 192.138.28.123**
CTMS_ADDR_ MODE	**{IP}** If this parameter is set to **IP**, the IP address instead of the host name is saved in CTMS_HOSTNAME. Use this parameter when CONTROL-M runs on a platform with more than one network card.
CTMSHOST	CONTROL-M / Server host name. Name of the primary host running CONTROL-M / Server.
DBGLVL	CONTROL-M / Agent diagnostic level (for use by Technical Support). Determines types of diagnostic messages generated. This parameter is normally set to zero (no diagnostics). Range: **0 - 4**. Default: **0**
EVENT_TIMEOUT	Job Tracking Timeout. Tracker event timeout in seconds. Default: **120**
LOCALHOST	Specifies a local Host Name other than the machine default Host Name.

Table 16 UNIX Agent – CONFIG.dat Parameters (Part 3 of 3)

Parameter	Description
LOGKEEPDAYS	Number of days to retain Agent proclog files. After this period, Agent proclog files are deleted by the New Day procedure. Default: **1** **Note:** This parameter is relevant only if CONTROL-M / Server does not pass a parameter that determines how many days to keep log files.
PROTOCOL_ VERSION	Server-Agent communication protocol version. Default: **06**
TRACKER_EVENT_P ORT	Number of the port for sending messages to the Tracker process when jobs end. This parameter is used in conjunction with the Common Event Mechanism parameter.
UTTIMEOUT	Maximum time (in seconds) the Agent waits after sending a request to CONTROL-M / Server. This timeout interval should be longer than the TCP / IP Timeout. Recommended value and default: **120**

The Control Module configuration parameters in Table 17 are stored in the **OS.dat** file.These parameters can be modified by the "shagent Utility" on page 57.

Table 17 CM for UNIX – OS.dat Parameters (Part 1 of 2)

Parameter	Description
CTM_SU_PATH	Alternative path that CONTROL-M / Agent will use to look for the **su** command. Default: **/bin/su**
CTM_PARM_ENC	Character used to enclose job processing parameters passed to jobs by CONTROL-M / Agent. Any character or string can be specified. A blank space (in single or double quotes) is valid.
CTM_PRM_DONT _DELETE	By default, temporary scripts generated from jobs are deleted at the end of job execution. If this value is set to **YES**, temporary scripts are not deleted.
CTM_PRM_KSH _FLAGS	Flags to include when running a job that is written in Korn Shell. Valid values: **-x**, **-v, n**. Default: **-x**
CTM_PRM_SH _FLAGS	Flags to include when running a job that is written in Borne Shell. Valid values: **-x**, **-v**, **n**. Default: **-x**
CTM_PRM_KSH _FLAGS	Flags to include when running a job that is written in Korn Shell. Valid values: **-x**,**-v**, **n**. Default: **-x**
PRINTER_NAME	Default printer for job output (sysout).
SYSOUT_MODE	Octal value indicating file access mode of the Sysout (output) file. **777** indicates the highest level of access.
PROCLOG_MODE	Octal value indicating file access mode of the Proclog (output) file. **777** indicates the highest level of access.

Table 17 CM for UNIX – OS.dat Parameters (Part 2 of 2)

Parameter	Description
SYSOUT_NAME	{JOBNAME | MEMNAME} If set to **JOBNAME**, parameter **Jobname** is used in the sysout file instead of parameter **Memname**. Default: **MEMNAME**
TRANSLATE_$0	If set to **Y**, reserved variable **$0** is replaced by a file name before a script is run. If set to **N**, this functionality is disabled. For more information, see "Use of the $0 Reserved Variable" on page 33.

Index

Symbols

A

B

C

D

E

F

H

I

J

L

O

P

R

S

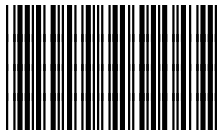
T

U

V

W

Notes



58742