CHAPTER 2

WatchDog Commands

The WatchDog is responsible for bootstrapping the MPLS VPN Solution and starting the necessary set of server processes. In addition, the WatchDog monitors the health and performance of each server to ensure it is functioning properly. In the event of a software error that causes a server to fail, the WatchDog automatically restarts the errant server.

The WatchDog is a background daemon process that is automatically installed as part of the installation procedure for MPLS VPN Solution. After the installation procedure has been completed, you can execute the **startwd** command to run the WatchDog for the first time. The WatchDog is automatically started any time the machine is rebooted.

This chapter provides the description, syntax, and arguments (listed alphabetically) for the following WatchDog commands:

startwd Command

This section provides the description and syntax for the **startwd** command.

Description

The **startwd** command starts the WatchDog and all MPLS VPN Solution processes. Running this manually is necessary after installing new software, after changing the **csm.properties** file, or when restarting after issuing a **stopwd** command. The **startwd** command is run automatically when the machine is rebooted.

Note The Orbix daemon must be running for the **startwd** command to operate correctly. If the Orbix daemon is not running, you will receive a message indicating that.

Syntax

startwd

Note The **startwd** command has no arguments.

Note The location of **startwd** is: <*MPLS VPN Directory*>**/bin**.

stopwd Command

This section provides the description and syntax for the **stopwd** command.

Description

The **stopwd** command stops the WatchDog and all MPLS VPN Solutionprocesses. Normally this will only be necessary before installing new versions of MPLS VPN Solution or changing the **csm.properties** file. When stopping and restarting the WatchDog, the **csm.properties** file is reread.

Syntax

stopwd [**-y**]

where: **-y** indicates not to prompt before shutdown. If **-y** is not specified, you are prompted with the following message: “Are you absolutely sure you want to stop the watchdog and all of its servers? Other users may be using this system as well. No activity (e.g.: collections, performance monitoring, provisioning) will occur until the system is restarted.” You are then prompted to reply **yes** or **no**.

Note The location of **stopwd** is: <*MPLS VPN Directory*>**/bin**.

wdclient Command

This section provides the description, syntax, and options (listed alphabetically) for the **wdclient** subcommands. These subcommands are diagnostic tools. This section also describes the column format of the output of each of the subcommands.

Note The location of **wdclient** is: <*MPLS VPN Directory*>**/bin**.

The following are the **wdclient** subcommands:

wdclient group Subcommand

This section provides the description and syntax for the **wdclient group** subcommand.

Description

The **wdclient group** subcommand lists the servers in the specified server group. Server groups provide a convenient way to start or stop a group of servers with a single command.

Syntax

wdclient [**-host** <*hostname*>] **group** <*group_name*>

where:

-host <*hostname*> is an optional parameter. <*hostname*> is the name of the remote host on which the WatchDog is running.

<*group_name*> is the name of a server group chosen from the list displayed by the **wdclient groups** command.

wdclient groups Subcommand

This section provides the description and syntax for the **wdclient groups** subcommand.

Description

The **wdclient groups** subcommand lists all the active server groups.

Syntax

wdclient [**-host** <*hostname*>] **groups**

where:

-host <*hostname*> is an optional parameter. <*hostname*> is the name of the remote host on which the WatchDog is running.

wdclient log Subcommand

This section provides the description and syntax for the **wdclient log** subcommand.

Description

The **wdclient log** subcommand displays the specified number of lines of the specified server log.

Syntax

wdclient [**-host** <*hostname*>] [**-poll** <*seconds*>] **log** <*log_name*> [<*lines*>]

where:

-host <*hostname*> is an optional parameter. <*hostname*> is the name of the remote host on which the WatchDog is running.

-poll <*seconds*> is an optional parameter. <*seconds*> is the number of seconds. A number other than zero indicates that when new status data is available it will be displayed every <*seconds*> seconds, where <*seconds*> is the specified number of seconds. The default **-poll** value is zero (0).

<*log_name*> is the name of a server log displayed by the **wdclient logs** command.

<*lines*> is an optional parameter. It is the number of lines (1 to 100) to be displayed from the end of the log. The default number of lines is 100.

Note The complete history (log) file of all WatchDog servers is in the **watchdog** subdirectory of the temporary directory as configured in the **csm.properties** file. This temporary directory is specified during system configuration. If the WatchDog is stopped and restarted, each log file is renamed from **server.**<*log_name*> to **server.**<*log_name*>**.**<*time_stamp*>, where <*log_name*> is the same as specified in the wdclient log subcommand and <*time_stamp*> is a time indicator of when this file was created. The new logs are then collected in **server.**<*log_name*>. If the WatchDog is not stopped and restarted within a 24-hour period, the log file is automatically renamed with a <*time_stamp*> and a new file is started. Also, any log file more than a week old is automatically deleted.

wdclient logs Subcommand

This section provides the description and syntax for the **wdclient logs** subcommand.

Description

The **wdclient logs** subcommand lists the names of all the logs.

Syntax

wdclient [**-host** <*hostname*>] **logs**

where: **-host** <*hostname*> is an optional parameter. <*hostname*> is the name of the remote host on which the WatchDog is running.

wdclient restart Subcommand

This section provides the description and syntax for the **wdclient restart** subcommand.

Description

The **wdclient restart** subcommand restarts one or more servers. Any dependent servers will also be restarted.

Note It is not necessary to restart servers in a properly functioning system. The **wdclient restart** command should only be run under the direction of Cisco Support. Restarting an individual server will not read in changes in the **csm.properties** file. For changes in the **csm.properties** file to be effective, stop the WatchDog and restart it.

Syntax

wdclient [**-host** <*hostname*>] **restart** {<*server_name*> | **group** <*group_name*> | **all**}

where:

-host <*hostname*> is an optional parameter. <*hostname*> is the name of the remote host on which the WatchDog is running.

You must choose one of the following arguments:

<*server_name*> is the name of a server chosen from the list displayed by the **wdclient status** command. See Table 2-1, "Servers and Their Functions," for server descriptions.

group <*group_name*> is the term **group** followed by the name of a server group chosen from the list displayed by the **wdclient groups** command.

all is all servers.

wdclient start Subcommand

This section provides the description and syntax for the **wdclient start** subcommand.

Description

The **wdclient start** subcommand starts one or more servers. Other servers that depend on the specified server(s) may also start.

Note It is not necessary to stop and start servers in a properly functioning system. The **wdclient start** command should only be run under the direction of Cisco Support.

Syntax

wdclient [**-host** <*hostname*>] **start** {<*server_name*> | **group** <*group_name*> | **all**}

where:

-host <*hostname*> is an optional parameter.<*hostname*> is the name of the remote host on which the WatchDog is running.

You must choose one of the following three arguments.

<*server_name*> is the name of a server chosen from the list displayed by the **wdclient status** command. See Table 2-1, "Servers and Their Functions," for server descriptions.

group <*group_name*> is the name of a server group chosen from the list displayed by the **wdclient groups** command.

all is all servers.

wdclient status Subcommand

This section provides the description, syntax, and information produced for the **wdclient status** subcommand.

Description

The **wdclient status** subcommand lists all the servers and their states. See Table 2-2 on page 2-8, "Valid States," for the list of all the states.

Syntax

wdclient [**-host** <*hostname*>] [**-poll** <*seconds*>] **status**

where:

-host <*hostname*> is an optional parameter. <*hostname*> is the name of the remote host on which the WatchDog is running.

-poll <*seconds*> is an optional parameter. <*seconds*> is the number of seconds. A number other than zero indicates that when new status data is available it will be displayed every <*seconds*> seconds, where <*seconds*> is the specified number of seconds. The default **-poll** value is zero (0).

Information Produced: Name Column

The **Name** column provides the name of each of the servers. Table 2-1 provides a list of the servers and a description of the function that each server provides.

Table 2-1 Servers and Their Functions

Server	Function
DataSetServer	Provides a CORBA front end for SA Agent and Accounting APIs.
EventGateway	Gateways events from the TIBCO domain to the CORBA domain.
LayoutServer	Provides topology layout recomputation services for web topology, which is used when selecting certain topology views.
ReportServerFactory	Launches and manages ReportServer processes that generate and provide access to dynamic web reports.
ResourceMgr	Handles device locking so a router's configuration is not modified by multiple service requests at the same time, and allocates Telnet Gateway Servers in the system for download/upload requests.
TGServer	Provides a CORBA API to download configlets, upload configuration files, and send IOS commands to the router.
TaskServer	Provides a CORBA front end to the MPLS VPN Solution task repository.
TemplateServer	Provides a CORBA front end to the Template Provisioning System.
VerifyReportServer	Back end that generates Verify Reports.
VpnInvServer	Provisioning API CORBA server.
aggregator	Aggregates collected datasets periodically.
httpd	Web server.
journal	Listens to all repository events and saves them into journal files. Also archives the journal files periodically.
lock_manager	Handles locking for the internal database.
log	Makes the output of tasks available to you in a browsable format.
poller	Gets requests from other data collectors and forwards the requests to the device. Gets the response and sends it to appropriate collectors.
rmiregistry	Underlying communication process necessary for ReportServerFactory and LayoutServer to communicate with web clients.
scheduler	Enables you to schedule tasks immediately or later in time, for one-time or repeated execution.
trapcatcher	Catches configuration change traps from routers.
watchdog_perf	Tracks performance for the system itself. The data is collected and stored in the internal database only. The data is useful for diagnostics.

Information Produced: State Column

The **State** column provides the current state of the server. Table 2-2 provides a description of each of the states in normal progression order.

Table 2-2 Valid States

State	Description
start_depends	This server has been asked to start, but is waiting for servers it depends on to start. Once all dependent servers have started, this server will transition to the state of starting.
starting	This server is currently starting. Once a successful heartbeat occurs, this server will transition to the state of started.
started	This server is currently started and running.
stop_depends	This server is supposed to be stopped, but it is waiting for servers it depends on to be stopped first.
stopping_gently	This server is in the process of stopping in a gentle fashion. That is, it was notified that it is to stop.
stopping_hard	This server is in the process of being killed because either it did not have a way to stop gently or because the gentle stop took too long.
stopped	This server is stopped. The WatchDog will either start it again or disable it if it has been frequently dying.
disabled_dependent	This server is disabled because one or more servers it depends on are disabled. If all servers it depends on are started, this server will automatically start.
disabled	This server is disabled and must be manually restarted.
restart_delay	This server is delaying before restarting. There is a short delay after a server stops and before it is restarted again.

Information Produced: Gen Column

The **Gen** column provides the generation of the server. Each time the server is started, the generation is incremented by 1.

Information Produced: Exec Time Column

The **Exec Time** column provides the date and time the server was last started.

Information Produced: PID Column

The **PID** column provides the UNIX process identifier for each server.

Information Produced: Success Column

The **Success** column provides the number of successful heartbeats since the server was last started. Heartbeats are used to verify that servers are functioning correctly.

Information Produced: Missed Column

The **Missed** column provides the number of missed heartbeats since the server was last started.

A few missed heartbeats could simply indicate the system was busy. However, more than a couple of missed heartbeats per day could indicate a problem. See the logs to diagnose the reason. If a server misses three heartbeats in a row, the server is automatically restarted.

Note Three missed heartbeats in a row is the default for restarting the server. The default number can be reset in the **csm.properties** file. After three failed attempts to restart in a row, the server is disabled.

wdclient stop Subcommand

This section provides the description and syntax for the **wdclient stop** subcommand.

Description

The **wdclient stop** subcommand stops one or more servers. Other servers that depend on the specified servers will also stop.

Note It is not necessary to stop servers in a properly functioning system. The **wdclient stop** command should only be run under the direction of Cisco Support.

Syntax

wdclient [**-host** <*hostname*>] **stop** {<*server_name*> | **group** <*group_name*> | **all**}

where:

-host <*hostname*> is an optional parameter. <*hostname*> is the name of the remote host on which the WatchDog is running.

You must choose one of the following arguments.

<*server_name*> is the name of a server chosen from the list displayed by the **wdclient status** command. See Table 2-1, "Servers and Their Functions," for server descriptions.

group <*group_name*> is the name of a server group chosen from the list displayed by the **wdclient groups** command.

all is all servers.

wdgui Command

This section provides the description and syntax for the **wdgui** command. This graphical interface to the WatchDog is a diagnostic tool that combines the functionality of the **wdclient status** and **wdclient log** commands. This section also describes the column format of the output when you click each of the tabs.

Description

The **wdgui** command activates the WatchDog user interface. See Figure 2-1, "VPN Solutions Center—Watch Dog."

The top of the screen provides a list of the names of servers. You can drag and drop the columns of information to rearrange them. The columns of information about the servers are described in the following sections:

- Name Column, page 2-11
- State Column, page 2-12
- Generation Column, page 2-13
- Exec Time Column, page 2-13
- Pid Column, page 2-13
- Success Column, page 2-14
- Missed Column, page 2-14

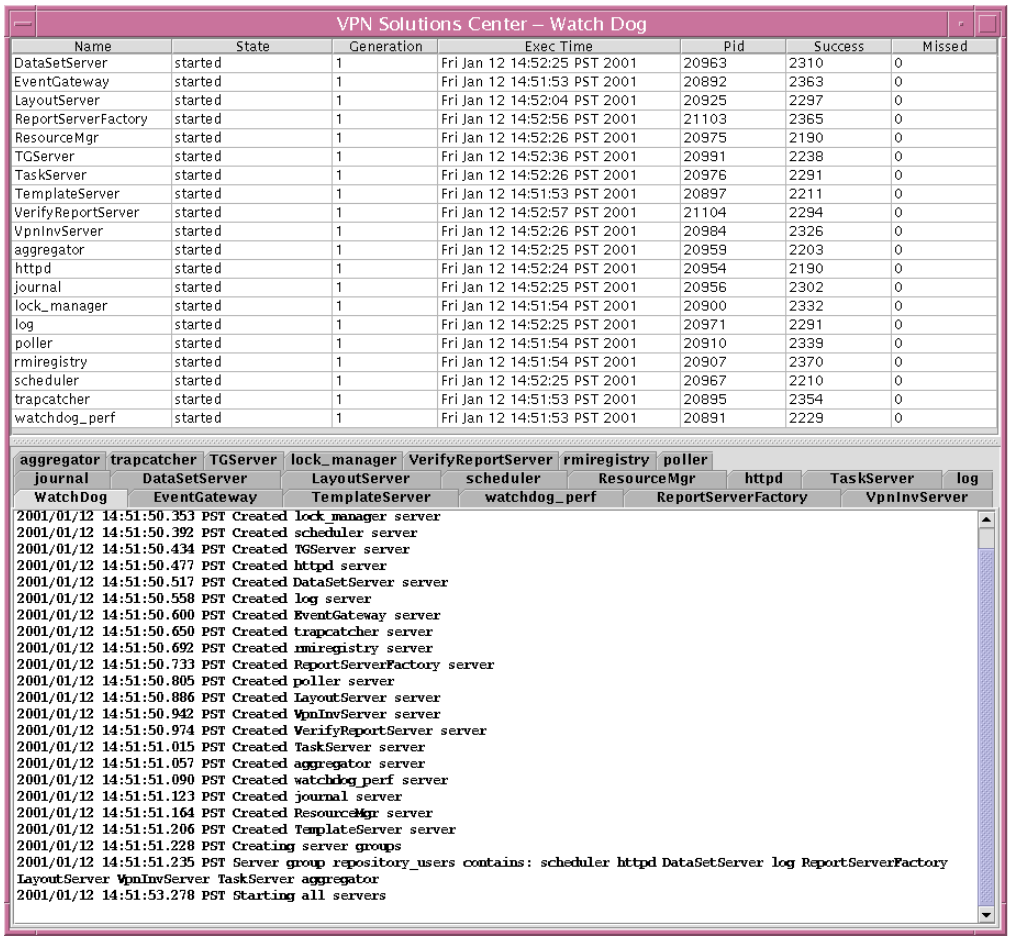
The bottom of the screen provides tabs for each of the servers. Click the tab of the server that you want to track and you will get up to the most current 250 lines of detailed log information.

Syntax

wdgui [**&**]

Note The **wdgui** command has no arguments. To run it as a background process, use the optional **&**.

Figure 2-1 VPN Solutions Center—Watch Dog



Name Column

The **Name** column provides the name of each of the servers. Table 2-3 provides a list of the servers and a description of the function that each server provides.

Note To sort alphabetically, click the column header **Name.** Uppercase sorts prior to lowercase.

Table 2-3 Servers and Their Functions

Server	Function
DataSetServer	Provides a CORBA front end for SA Agent and Accounting APIs.
EventGateway	Gateways events from the TIBCO domain to the CORBA domain.
LayoutServer	Provides topology layout recomputation services for web topology, which is used when selecting certain topology views.

Table 2-3 Servers and Their Functions (continued)

Server	Function
ReportServerFactory	Launches and manages ReportServer processes that generate and provide access to dynamic web reports.
ResourceMgr	Handles device locking so a router's configuration is not modified by multiple service requests at the same time, and allocates Telnet Gateway Servers in the system for download/upload requests.
TGServer	Provides a CORBA API to download configlets, upload configuration files, and send IOS commands to the router.
TaskServer	Provides a CORBA front end to the MPLS VPN Solution task repository.
TemplateServer	Provides a CORBA front end to the Template Provisioning System.
VerifyReportServer	Back end that generates Verify Reports.
VpnInvServer	Provisioning API CORBA server.
aggregator	Aggregates collected datasets periodically.
httpd	Web server.
journal	Listens to all repository events and saves them into journal files. Also archives the journal files periodically.
lock_manager	Handles locking for the internal database.
log	Makes the output of tasks available to you in a browsable format.
poller	Gets requests from other data collectors and forwards the requests to the device. Gets the response and sends it to appropriate collectors.
rmiregistry	Underlying communication process necessary for ReportServerFactory and LayoutServer to communicate with web clients.
scheduler	Enables you to schedule tasks immediately or later in time, for one-time or repeated execution.
trapcatcher	Catches configuration change traps from routers.
watchdog_perf	Tracks performance for the system itself. The data is collected and stored in the internal database only. The data is useful for diagnostics.

State Column

The **State** column provides the current state. Table 2-4 provides a description of each of the states in normal progression order.

Table 2-4 Valid States

State	Description
start_depends	This server has been asked to start, but is waiting for servers it depends on to start. Once all dependent servers have started, this server will transition to the state of starting.
starting	This server is currently starting. Once a successful heartbeat occurs, this server will transition to the state of started.
started	This server is currently started and running.
stop_depends	This server is supposed to be stopped, but it is waiting for servers it depends on to be stopped first.
stopping_gently	This server is in the process of stopping in a gentle fashion. That is, it was notified that it is to stop
stopping_hard	This server is in the process of being killed because either it did not have a way to stop gently or because the gentle stop took too long.
stopped	This server is stopped. The WatchDog will either start it again or disable it if it has been frequently dying.
disabled_dependent	This server is disabled because one or more servers it depends on are disabled. If all servers it depends on are started, this server will automatically start.
disabled	This server is disabled and must be manually restarted.
restart_delay	This server is delaying before restarting. There is a short delay after a server stops and before it is restarted again.

Generation Column

The **Generation** column provides the generation of the server. Each time the server is started, the generation is incremented by 1.

Exec Time Column

The **Exec Time** column provides the date and time that the server was last started.

Note To sort from the earliest to the latest date and time, click the column header **Exec Time.**

Pid Column

The **Pid** column provides the UNIX process identifier for each server.

Success Column

The **Success** column provides the number of successful heartbeats since the server was last started. Heartbeats are used to verify that servers are functioning correctly.

Note To sort from the least number of successful heartbeats to the greatest number of successful heartbeats, click the column header **Success**.

Missed Column

The **Missed** column provides the number of missed heartbeats since the server was last started.

A few missed heartbeats could indicate that the system was busy. However, more than a couple of missed heartbeats per day could indicate a problem. See the logs to diagnose the reason. If a server misses three heartbeats in a row, the server is automatically restarted.

Note Three missed heartbeats in a row is the default for restarting the server. The default number can be reset in the **csm.properties** file. After three failed attempts to restart in a row, the server is disabled.

wdperf Command

This section provides the description, syntax, and report information for the **wdperf** command. This section also describes the reports that are generated by executing this command and the common information in these reports:

- Average, Minimum, and Maximum % CPU Utilization per Time Period, page 2-17
- Average, Minimum, and Maximum % Memory Usage per Time Period, page 2-18
- Average, Minimum, and Maximum Virtual Memory Usage per Time Period, page 2-18

This graphical interface to the WatchDog provides information about system performance and resource utilization.

Note The default for netsys.watchdog.server.watchdog_perf.enable in the csm.properties file is **false**, which disables data gathering for the **wdperf**. To enable this function, set netsys.watchdog.server.watchdog_perf.enable to **true** and relaunch the WatchDog.

Description

The **wdperf** command is a monitoring tool for MPLS VPN Solution that provides reports indicating the % CPU utilization, the % Memory usage, and the amount of virtual memory used by each of the system's servers and user-defined tasks. The reported values are based on performance data gathered by the WatchDog.

Syntax

wdperf [**%cpu** | **%mem** | **vmem**] [**&**]

or

wdperf {**%cpu** | **%mem** | **vmem**} [<*date*> | **start**] [**&**]

where:

%cpu is a parameter that causes the Average % CPU Utilization per Hour report to be displayed. This is the default option.

%mem is a parameter that causes the Average % Memory Utilization per Hour report to be displayed.

vmem is a parameter that causes the Average Virtual Memory Utilization per Hour report to be displayed.

<*date*> is an optional parameter that specifies the date for which performance data will be displayed. The default date is the current date. The format of the date is either: *mm*/*dd*/*yy* or *mm*/*dd*/*yyyy*, where:

- *mm* is the month, specified as **01** to **12**.
- *dd* is the day, specified as **01** to **31**.
- *yy* or *yyyy* is the year, specified in two-character or four-character year designations.

start is an optional parameter that causes the earliest available performance data to be displayed (that is, the repository creation date).

& is an optional parameter that causes **wdperf** to be run as a background process.

Note The location of **wdperf** is: <*MPLS VPN Directory*>**/bin**.

Report Information

For a description of the reports created by the **wdperf** command, first see explanations of the generic report fields in the "Status Row" and "Filter Information" sections in Chapter 14, "Reports Overview." Additionally, each report has the following information:

- Results Area
- Detail Area
- Bottom Task Bar

Results Area

The columns of information are as follows:

- **Process**. This column lists the names of all the servers and task processes managed by WatchDog.
- The data displayed in each of the other columns depends on the current display level. The **Daily** display level displays data for each hour of the selected day, **00** to **23**. These columns start at midnight (**00:00**) and go to 11:00 p.m (**23:00**). The **Hours** display level displays data for each minute of the selected hour, **00** to **59**. To switch between the display levels, see the "Bottom Task Bar" section on page 2-16. The color of each cell depends on the value contained in the cell. A blue cell indicates the server was restarted in the designated time period. Note that all blue cells have a minus sign preceding the cell value.

Detail Area

The information in this area is:

pid = <####>
where: <####> is the Process identifier of the server or task (process) that you highlight in the Results Area.

start time = localized date, time, and time zone when the server or task (process) that you highlight in the Results Area started.

Note If the highlighted server or task restarts, multiple lines will be displayed in the Detail Area, one line for each time the server or task starts.

Bottom Task Bar

From left to right, the bottom task bar includes the following items:

- **<=** is a button that you click to display the previous day's data when at the **Daily** display level and the previous hour's data when at the **Hours** level.
- **=>** is a button that you click to display the next day's data when at the **Daily** display level and the next hour's data when at the **Hours** level.

Note If you want to view a report for a specific date, you may want to re-enter the **wdperf** command with the desired date. This may be preferable to using the **<=** and **=>** buttons, which only display adjacent days one day at a time.

- **Metric** is a drop-down list with the following choices:
  - **%cpu** displays the percentage of the CPU that is being occupied by each of the WatchDog's processes. Values below 20% are displayed in green, those between 20% and 50% are displayed in yellow, and those above 50% are displayed in red.
  - **%mem** displays the percentage of the machine's physical memory that is being used by each of the WatchDog's processes. Values below 20% are displayed in green, those between 20% and 50% are displayed in yellow, and those above 50% are displayed in red.
  - **vmem** displays the amount of virtual memory (in kilobytes) allocated to each of the WatchDog's processes. Values are displayed in various color shades to highlight memory usage trends.
- **Aggregate** is a drop-down list with the following choices:
  - **Average** displays the average value for the selected metric during the applicable time period (for example, one hour or one minute, depending on the current display level).
  - **Maximum** displays the maximum value for the selected metric during the applicable time period (for example, one hour or one minute, depending on the current display level).
  - **Minimum** displays the minimum value for the selected metric during the applicable time period (for example, one hour or one minute, depending on the current display level).
- **Hours** is a drop-down list that displays a selection of hours from which to choose (**00** to **23**). When you select an hour from the list, you switch to the **Hours** display level for the selected hour, which displays the data aggregated per minute. From the **Hours** level, you can return to the **Daily** level by clicking on the **Daily** button in the bottom task bar.

Average, Minimum, and Maximum % CPU Utilization per Time Period

These reports display the percentage of the CPU that is being occupied by each of the WatchDog's processes. Values less than 20% are displayed in green, those between 20% and 50% are displayed in yellow, and those greater than 50% are displayed in red.

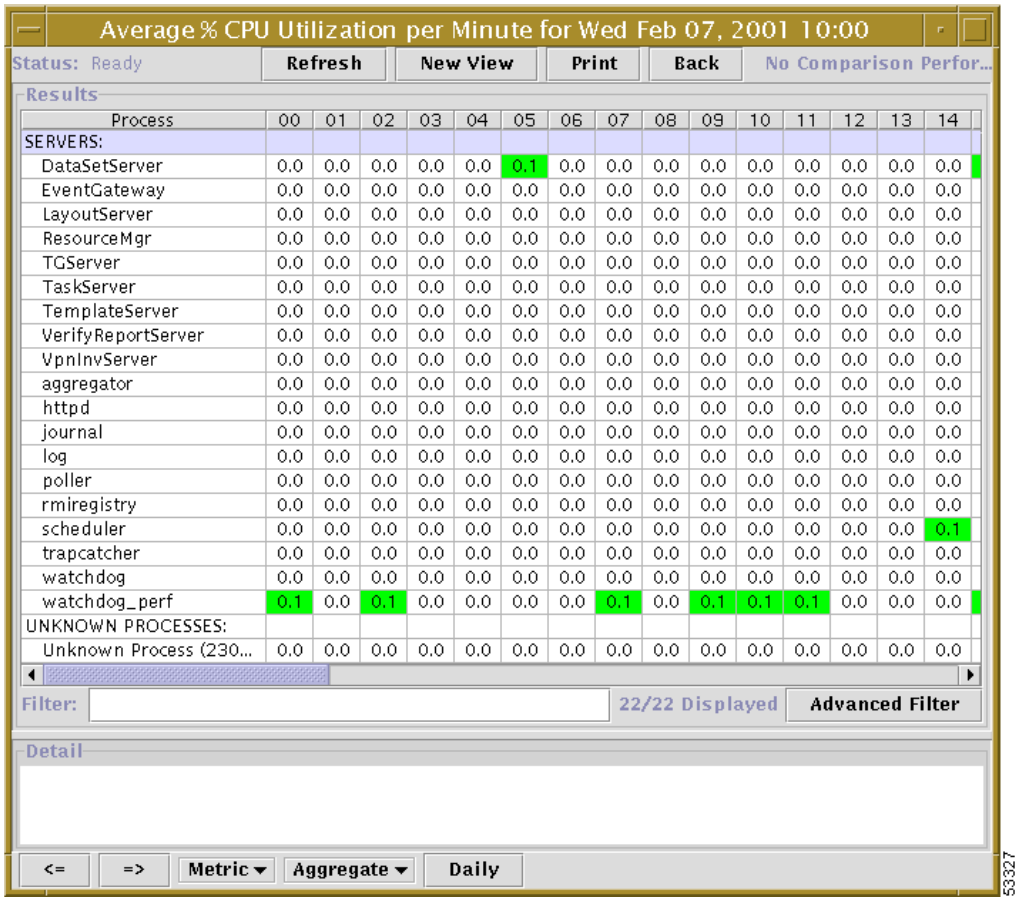
The Average % CPU Utilization per Hour report for the current date is the default report if you do not specify another **Metric** on the command line, as specified in the "Syntax" section, and maintain the default **Aggregate** selection on the bottom task bar.

See a sample of the % CPU Utilization report, as shown in Figure 2-2, "% CPU Utilization Report.

Some processes in MPLS VPN Solution launch children processes as an **Unknown Process**. The command line command that launches the **Unknown Process** can be selected in the top window, and its related arguments (args) are listed in the **Detail** pane.

From this report, you can use the controls in the bottom task bar to navigate to reports displaying other metrics, aggregates, and display periods.

Figure 2-2 % CPU Utilization Report



Process	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14
SERVERS:															
DataSetServer	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
EventGateway	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
LayoutServer	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ResourceMgr	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
TGServer	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
TaskServer	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
TemplateServer	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
VerifyReportServer	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
VpnInvServer	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
aggregator	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
httpd	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
journal	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
log	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
poller	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
rmiregistry	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
scheduler	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1
trapcatcher	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
watchdog	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
watchdog_perf	0.1	0.0	0.1	0.0	0.0	0.0	0.0	0.1	0.0	0.1	0.1	0.1	0.0	0.0	0.0
UNKNOWN PROCESSES:															
Unknown Process (230...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Average, Minimum, and Maximum % Memory Usage per Time Period

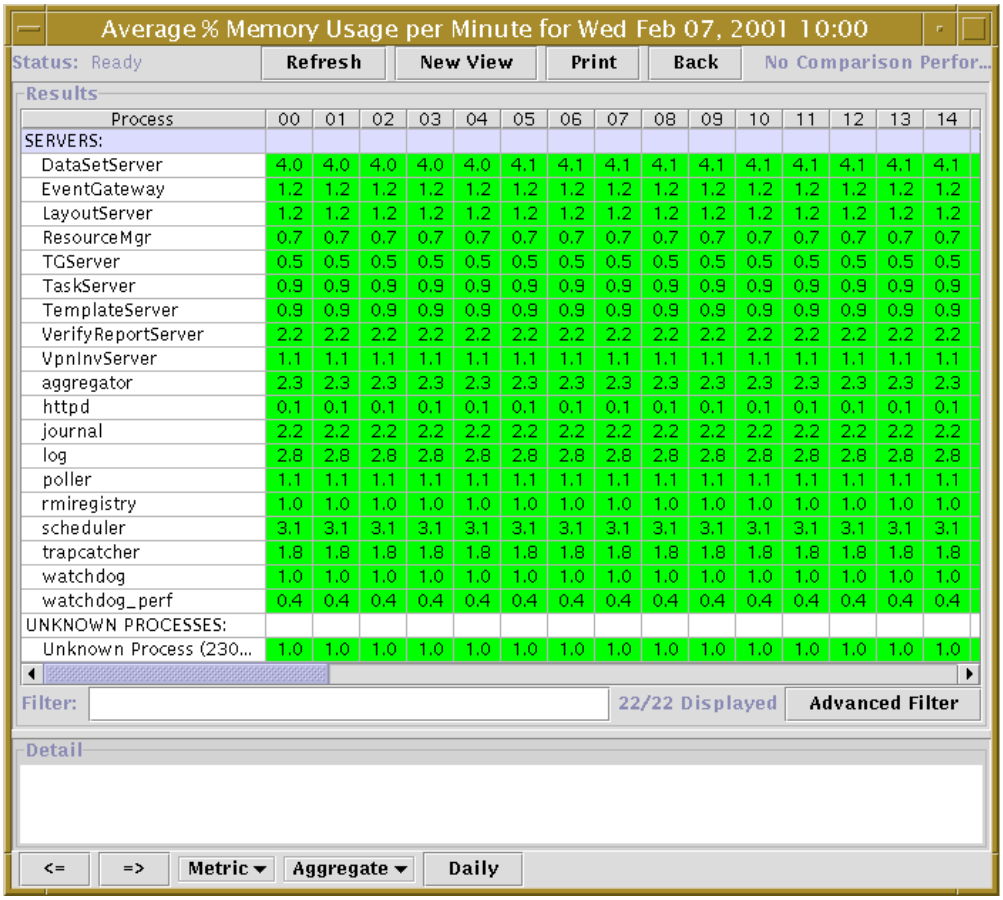
These reports display the percentage of the machine's physical memory that is being used by each of the WatchDog's processes. Values less than 20% are displayed in green, those between 20% and 50% are displayed in yellow, and those greater than 50% are displayed in red.

The Average % Memory Utilization per Hour report for the current date is the report that is displayed if you specify **%mem** on the command line and maintain the other defaults on the command line, as specified in the "Syntax" section, and the default **Aggregate** selection on the bottom task bar.

See a sample of the % Memory Utilization report, as shown in Figure 2-3, "% Memory Utilization Report."

From this report, you can use the controls in the bottom task bar to navigate to reports displaying other metrics, aggregates, and display periods.

Figure 2-3 % Memory Utilization Report



Average, Minimum, and Maximum Virtual Memory Usage per Time Period

These reports display the amount of virtual memory (in kilobytes) allocated to each of the WatchDog's processes. Values are displayed in various color shades to highlight memory usage trends.

The Average Virtual Memory Utilization per Hour report for the current date is the report that is displayed if you specify **vmem** on the command line and maintain the other defaults on the command line, as specified in the "Syntax" section, and the default **Aggregate** selection on the bottom task bar.

See a sample of the Virtual Memory Utilization report, as shown in Figure 2-4, "Virtual Memory Utilization Report."

From this report, you can use the controls in the bottom task bar to navigate to reports displaying other metrics, aggregates, and display periods.

Figure 2-4 Virtual Memory Utilization Report