NAME

contool − capture and display console output

SYNOPSIS

contool [**-c** *file*] [**-f**] [**-i** *input*] [**-l**] [**-L** *logfile*] [**-n**]

DESCRIPTION

Contool captures and displays any messages sent to the system console. Each message is timestamped as it arrives. The messages are displayed in a scrolling text window, so the user can scroll through old messages.

When a message arrives, **contool** will beep and, if closed, begin blinking its icon until the user opens the tool. This behavior can be changed by modifying the default contool properties.

Contool must be run under either Open Windows or X Windows. It accepts the standard Open Windows command line options.

OPTIONS

-c *file* specifies an alternate configuration file. *Contool* normally reads its configuration information from the path specified in the CONTOOL_FILTERS environment variable or, if CONTOOL_FILTERS is not defined, from ˜/.contool.

-f forces *contool* to fork itself into the background after acquiring the console. This option is most useful when starting *contool* from your ˜/.xinitrc or ˜/.openwin-init file, where multiple tools are started in the background. Even if *contool* is started first, other tools may begin running and produce error messages before *contool* can acquire the console. By using the **-f** option you can start *contool* in the foreground, preventing other tools from starting until *contool* has acquired the console.

-i *input* specifies an alternate input source. *Contool* normally reads messages from the system console. If **-i** is used, *contool* will read messages from the specified *input*. *Input* must be the pathname of a either a FIFO, or a character special device like /dev/tty. *Input* can also be given as “-”, indicating that contool should read messages from its standard input.

-l enables logging when *contool* starts. Initially, *contool* does not log messages to a file. This option allows logging to be started automatically. Logging behavior can be enabled and disabled via the “File” menu (see below).

-L *logfile*

specifies an alternate log file. The value of *logfile* overrides any log file stored in the configuration file. The **-l** and **-L** options used together allow multiple copies of contool to be started on a single machine, with each copy logging to a different file, without having to create and maintain multiple configuration files.

-n prevents *contool* from acquiring the console when it starts. This is most useful when testing a new version of contool, so that the test copy does not interfere with any previously running console tool.

USER INTERFACE

Contool presents the user with a control panel containing three buttons and a scrolling text window. Console messages appear in the text window; *contool*’s behavior is managed with the **File**, **View**, and **Edit** buttons. These buttons have several menu choices associated with them:

File: Load Configuration...

brings up the **Load Configuration** dialog box. This dialog contains a non-exclusive setting, a text field, and a single **Load** button. The setting has two entries: “Tool Properties” and “Filter Definitions”. The user can choose which of these items to load from the configuration file. The default action is to load both properties and filters from the file.

The text field contains the path of the configuration file. By default, this is either the value of the **-c** option (above), the value of the CONTOOL_FILTERS environment variable, or ˜/.contool. This text field provides file name completion, like *csh*(1), by typing a space or carriage return.

After determining which components to be read from the file, and the name of the file, clicking on the **Load** button will cause the desired elements to be read from the file. If filters are loaded, any existing filters are discarded. If *contool* was in the middle of filtering a multi-line message, that filtering action is terminated.

Contool can read files created by all previous versions of *contool*. Files written by *contool* cannot be read by any previous version.

File: Save Configuration...
is analogous to the **Load Configuration** operation, bringing up a dialog box containing a non-exclusive setting, a text field, and a **Save** button. In the same manner as the **Load Configuration** dialog, the user indiactes which items he desires to save, specifies the destination file, and clicks on the **Save** button to save the information.

Contool writes the data in a format incompatible with previous versions (release 3.0 or earlier) of the tool.

File: Start Logging
starts logging console messages to the log file specified by the tool properties, or the **-L** option (above). The tool properties also contains a switch which determines whether messages are written before or after filtering occurs. If before, all messages are logged. If after, a message is written to the log if it does not match a filter and the default action in the tool properties has “Log message” enabled; or if it matches a filter with both the “Save” and “Log message” attributes set.

The log file is written in such a manner that multiple invocations of *contool* can write to the same log file without conflict. New messages are always appended to the log file. Log files can be forcibly flushed with a SIGHUP; see SIGNALS, below.

File: Stop Logging
stops the logging process.

File: Print
uses the “Print Filter” specified in the tool properties to print the contents of the console. Only messages saved in the console can be printed.

File: About Contool...
provides a brief history of *contool*, and allows users to send e-mail to the contool developer.

View: Archive...
opens the message archive dialog box. This dialog box displays a scrolling window containing all messages archived from the main console display. See MESSAGE ARCHIVE, below.

View: Archive Messages
moves all the messages in the console display to the message archive. This is a handy way to remove already viewed messages from the console without losing them for later review. By periodically moving messages to the archive, only the most recent messages will be visible in the main console display. Messages can be archived automatically as well; see TOOL PROPERTIES, below.

View: Become Console
ensures that *contool* has the system console attribute. SunOS allows exactly one process in the system to own the console. If some other process takes control of the console, the user can reassign the console attribute to *contool* using this menu selection.

View: Clear Messages
clears all messages from the console text display.

View: Reset Filter
interrupts filtering of a multi-line filter. If the user incorrectly specifies the end pattern of a multi-line filter, *contool* may never stop processing the filter, causing all subsequent console messages to be incorrectly handled. This selection resets the effect of any multi-line filter that may be in effect.

When processing a multi-line filter, *contool* displays the starting filter pattern in the left window footer. If the left footer is blank, *contool* is not processing a multi-line filter.

Edit: Filters...
brings up the **Filters** dialog box. See EDITING FILTERS, below.

Edit: Properties...
brings up the **Tool Properties** dialog box. See TOOL PROPERTIES, below.

EDITING FILTERS

The **Filters** dialog presents a scrolling list of filters and a variety of devices used to modify the current filter set. The items in this dialog are:

Filters This scrolling list shows the starting pattern of each filter currently used by *contool*. The pattern text is preceded by a small glyph indicating whether the filter is a single-line or multi-line filter.

If exactly one item in the list is selected, the properties of that filter will be displayed in the dialog box. If no, or more than one, filter is selected, the remainder of the dialog box is grayed out.

Insert This button inserts a new blank filter into the scrolling list, allowing the user to add new filters to the filter set. A menu attached to the button allows the user to select the insert point: at the top of the list, before the current selection, after the current selection, or at the bottom of the list. The “before” and “after” choices are only enabled if exactly one filter in the list is selected. The default position is the bottom of the list.

Edit This button edits the currently selected filters in the list. The menu attached to this button has four choices: **Cut**, **Copy**, **Paste**, and **Delete**.

The **Cut** operation removes the selected filters from the list and places them on the clipboard, where they can subsequently be pasted back into the list.

The **Copy** operation copies the selected filters to the clipboard. The filters are not removed from the list. The copied filters can subsequently be pasted back into the list.

The **Paste** operation copies filters from the clipboard into the list. This selection has a menu which specifies the paste position: at the top of the list, before the current selection, after the current selection, or at the bottom of the list. The “before” and “after” choices are only enabled if exactly one filter in the list is selected. The default position is the bottom of the list.

The **Delete** operation removes the selected filters from the list without placing them on the clipboard. Once deleted, filters cannot be recovered with a paste operation.

The **Cut**, **Copy**, and **Delete** selections are only presented if one or more filters in the list are selected. The **Paste** selection is only available after a **Cut** or **Copy** operation.

Update
This button updates the currently selected filter using the values presented in the remainder of the dialog box. This button is only accessible if exactly one filter in the list is selected.

Update is used to modify an existing filter. When just that filter is selected, its attributes are placed into the other dialog elements described below. After adjusting the filter attributes, the user clicks the **Update** button to apply the changes to the currently selected filter.

In a similar manner, **Update** is used to apply attributes to a new, blank filter placed in the list via the **Insert** button.

Type The **Type** toggle indicates whether a filter will match just a single line message, or will match a multiple line message. When “Single line filter” is chosen, the **End pattern** item is disabled, and the user must specify the pattern which will match a single line of text written to the console. When “Multi-line filter” is selected, the **End pattern** item is enabled, and the user needs to specify both a starting and an ending pattern. All text following a line which matches the starting pattern, up to and including a line which matches the ending pattern, is considered to be part of the filtered message.

Pattern

This text field specifies the regular expression which matches the first (and, in the case of single line filters, the only) line of text in the filtered message. Any valid regular expression is permitted. Users that are trying to match some text exactly should be aware that regular expressions can match text anywhere in a line, and that the expression should be anchored to the start (or end) of the line by using the "^" (or "$") metacharacters. For more information on regular expressions, see *ed*(1).

As a special extension to regular expressions, *contool* recognizes a backslash ("\") followed by one or more octal digits as a single character in the expression. This allows non-printing characters, such as control characters, to be inserted in the text pattern. If a backslash is followed by any other character, it is placed in the pattern verbatim. Thus, to create an expression which matches a control-G followed by a backslash, the pattern "\007\" would suffice.

End pattern

If the **Type** is set to "Multi-line filter", this field must contain the regular expression which matches the last line of the block of text handled by this filter.

Timeout

If the **Type** is set to "Multi-line filter", this field sets a limit on how long *contool* will process the filter. This prevents filters with erroneous end patterns from absorbing all console output once they begin filtering. The default value, 0, indicates that no timeout is in effect.

Comment

This text field contains any comments regarding the filter the user wishes to record. Since some filters can be rather arcane, it is suggested that users comment their filters.

When matched

This exclusive setting dictates the behavior of *contool* when a filter is matched. If "Save message" is chosen, the message is copied into the console display, and various actions can be taken. If "Ignore message" is selected, the filter text is discarded and no further actions are taken by *contool*.

When saved

If **When matched** is set to "Save message", this non-exclusive choice item will be enabled, allowing the user to specify what contool should do with this message.

The "Beep" choice causes the terminal bell to be sounded. If selected, the beep counter to the right of this item is enabled, allowing the user to choose anywhere from one brief beep up to 99 annoying beeps.

The "Command" choice causes a single command to be executed. The text field to the right of this item must contain the command to be executed. *Contool* will write the text of the message to the standard input of the command. For example, using "mail -s 'Contool output' user" as the command would mail the message text to the *user*.

The "Flash icon" choice causes the *contool* icon to flash, alternating between the "Check console" and "Flash" icons.

The "Log message" choice causes the message to be written to the message log, if logging is enabled and is performed after filtering.

The "Open window" choice causes *contool* to open from its iconic state, and to move in front of any obscuring windows.

The "Timestamp" choice causes *contool* to write a timestamp to the console before copying the message into the console. The timestamp is written in conjunction with the timestamp resolution specified in the **Tool Properties** dialog.

Apply This button makes the filters contained in the scrolling list the current set of active filters. Until this button is clicked, all changes made to the filters are not used by *contool*. After clicking **Accept**, the changed filters become the current working set.

Note that even after clicking **Accept**, the configuration file is not updated. To make the changes permanent between invocations of *contool*, press the **Apply and Save** button, or use the **Save Configuration** dialog to save the changed filters.

Apply and Save
This button makes the filters in the scrolling list the current set of active filters and writes those filters and the tool properties to the current configuration file. To write the filters to a different file, or to write just the filters without the tool properties, press the **Apply** button and use the **Save Configuration** dialog instead.

Reset This button discards any changes made to the current filter set, restoring the filter list to match the current filter set in use by *contool*.

TOOL PROPERTIES

The **Tool Properties** dialog allows the user to change the default behavior of *contool*. This includes various tool attributes, and the actions taken when a message arrives which does not match any filter. The various properties include:

Default action
This non-exclusive setting determines the actions taken by *contool* when a message arrives which does not match any filter. The various choices in this setting exactly correspond to the **When saved** setting the **Filters** dialog, above.

Log file
This text field contains the path of the file to which messages will be logged. This field must be filled in before logging is enabled.

Log messages
If this exclusive setting is set to “before filtering”, all messages will be logged. If set to “after filtering”, messages that match filters whose “When matched” behavior is set to “Ignore message” will not be logged.

Archive messages
This exclusive setting determines how messages will be moved from the main console display to the message archive. If set to “Manually”, messages will only be archived when the user selects **Archive Messages** from the **View** menu in the main *contool* window. If set to “When closing contool”, messages are copied to the archive whenever the *contool* window is closed. This mode assumes that you typically open *contool*, read all the messages, and close the window. Each time you open the window, you’ll only see messages that have arrived since you last closed *contool*.

Archived messages can be viewed in the message archive, described below.

Print filter
This text field specifies the command to be used to print the console. The default is “lpr”. Local site dependencies may require a different command.

“All is well” icon
This text field contains the path of a file in either icon or XBM format. Icon-format files may be created with *iconedit*(1); XBM-format files are created with a number of tools. If the path is not absolute, the value of the ICON_PATH environment variable will be used to find the file. ICON_PATH should contain a list of directories, separated by colons. The contained image will be used as *contool*’s regular icon image. This image is displayed whenever no flashing is in effect.

“All is well” icon mask
This text field contains the path of a file in either icon or XBM format. The contained image will be used to mask *contool*’s “All is well” icon image when *contool* is run on a color desktop. Masking is not supported on monochrome desktops. The “All is well” icon is not masked if this field is left blank.

“Check console” icon
This image is alternated with the **“Flash” icon** image whenever flashing is required.

“Check console” icon mask

This image will be used to mask *contool*’s “Check console” icon image when *contool* is run on a color desktop. The “Check console” icon is not masked if this field is left blank.

“Flash” icon

This image is alternated with the **“Check console” icon** image whenever flashing is required. Ideally, all three icons should be the same size.

“Flash” icon mask

This image will be used to mask *contool*’s “Flash” icon image when *contool* is run on a color desktop. The “Flash” icon is not masked if this field is left blank.

Timestamp resolution

This numeric field specifies the minimum number of seconds to wait before writing a new timestamp to the console. Messages which require timestamping will only write a timestamp if this number of seconds have transpired since the last timestamp.

Maximum message text

This numeric field determines the maximum size, in bytes, of messages that will be stored in the console. When writing a message to the console would exceed this limit, some number of bytes, as determined by the **Overflow delete amount**, below, will be removed from the front of the console. This feature prevents the console from becoming so large over time that it begins to swamp system resources.

Overflow delete amount

When writing a message to the console would exceed the **Maximum message text**, above, text will be deleted from the beginning of the console to make room. This numeric field specifies how many bytes to remove to make room. *Contool* will attempt to remove whole messages within the constraints of the console size to preserve a readable console.

Apply This button makes the values in the dialog box the current tool properties. Until this button is clicked, all changes made to the properties are not used by *contool*. After clicking **Accept**, the changed values become the current properties.

Note that even after clicking **Accept**, the configuration file is not updated. To make the changes permanent between invocations of *contool*, use the **Save Configuration** dialog to save the changed properties.

Reset This button discards any changes in the dialog box made to the current properties, restoring the properties to match the current properties in use by *contool*.

MESSAGE ARCHIVE

The **Message Archive** dialog allows the user to view archived messages. The dialog presents a scrolling text window and two buttons.

Clear The **Clear** button removes all the messages from the archive. Normally, the archive works like the main *contool* display: it retains a certain amount of text, and deletes the oldest messages as new messages arrive in excess of that amount. The **Clear** button circumvents this feature and explicitly clears the archive display.

The archive window will hold ten times the amount of text specified for the main console display, as determined by the **Maximum message text** value in the **Tool Properties** dialog, described above.

Print The **Print** button prints the contents of the archive, using the “Print Filter” specified in the **Tool Properties** dialog.

CONFIGURATION FILE FORMAT

Previous versions of *contool* relied on the user editing the configuration file by hand. This version manages the file automatically, and it is not intended that the file be edited directly by users. See EDITING FILTERS, above, for information on modifying the behavior of *contool*.

ENVIRONMENT VARIABLES

Contool uses certain environment variables to control its behavior. They are:

CONTOOL_FILTERS
specifies the file from which *contool* reads its filters and properties. If not defined, ˜/.contool is used. If **-c** is specified, it overrides the use of CONTOOL_FILTERS.

CONTOOL_LABEL
specifies the label to be placed in *contool*'s main window. The default label contains the current release number of *contool*.

CONTOOL_LOGNAME
specifies the name to be prefixed to each message written to the console log file. If this variable is not defined, the machine's hostname is used.

ICON_PATH
contains a colon-separated list of directories which *contool* will search to find the icon files specified in various tool properties.

SIGNALS

Contool will respond to certain Unix signals. They are:

SIGHUP
Upon receipt of SIGHUP, *contool* will close and reopen its log file, if logging is enabled. This guarantees that logged messages are flushed to disk.

SIGUSR1
Upon receipt of SIGUSR1, *contool* will stop blinking its icon. This is a handy way to stop blinking without opening *contool*.

FILES

˜/.contool	configuration file

SEE ALSO

cmdtool(1), ed(1), kill(1), mkfifo(2), signal(3)

AUTHOR

Chuck Musciano
Advanced Technology Department
Harris Corporation
PO Box 37, MS 16/1912
Melbourne, FL 32902
(407) 727-6131
E-mail: chuck@trantor.harris-atd.com
Fax: (407) 729-3363

BUGS

Contool is a view-only tool, and there is no way to type commands on the console.

Window system bugs may cause unusual, but harmless, quirks in the behavior of *contool*. In particular, displaying a menu in a dialog box without actually selecting an item from that menu may cause the dialog box to close.