

Tru64 UNIX

Sending Selected Events to a Cellular Phone or Pager

August 2000

This Best Practice describes how to configure Tru64 Event Management (EVM) to send selected events to a cellular phone or pager.

Contents

Sending Selected Events to a Cellular Phone or Pager

Is This Best Practice Right for You?	1
Before You Begin	2
Messaging Capability	2
Electronic Mail Address	2
EVM Filters	2
EVM Logger Configuration File	3
Using Templates with Evmshow	3
Applying the Best Practice	4
Verifying Success	6
Troubleshooting	7
Comments and Questions	7
Legal Notice	8

Sending Selected Events to a Cellular Phone or Pager

EVM provides a means for system components or applications to indicate when something of interest has happened, such as a disk failure or a task completion. These indications are called events. You can configure EVM to monitor events on your system and to notify you as soon as interesting events occur. In this Best Practice, EVM notifies you by forwarding the event information to a cellular phone or pager.

The procedures in this Best Practice apply to the Tru64 UNIX operating system Version 5.0A or higher. You must be root to apply this Best Practice.

See the Tru64 UNIX Best Practices Web page for more information about Best Practices documentation.

Is This Best Practice Right for You?

Not all Best Practices apply to all configurations, so you must be sure that it is appropriate for your system and circumstances. To use this Best Practice, you must meet the requirements described in the following table:

Requirement	Description
Operating System	Tru64 UNIX Version 5.0A or higher.
Assumptions	The instructions assume that you have a basic understanding of EVM and Bourne shell scripting. They assume that your pager or cellular phone is alphanumeric and that your device is able to receive electronic mail messages.
Restrictions	The format and content of the message depends on the size of the cellular phone or pager display and the service constraints.

Before You Begin

Before you apply the Best Practice for sending selected events to a cellular phone or pager, you must understand some background information and perform the following tasks:

1. Determine whether or not your device is capable of receiving alphanumeric messages.
2. Find out its electronic mail address if your device is capable.
3. Add an event capturing filter to the event logger.
4. Add a forwarding command to the event logger, which in this procedure executes a Bourne shell script you will create in the next step.
5. Create the Bourne shell script, which uses a template to format the selected events for presentation on a cellular telephone or pager and sends the information to the device.

Messaging Capability

Some cellular phones and pagers are capable of receiving alphanumeric messages. Since different services have different options, this Best Practice does not describe them. Read the literature that came with your service or contact your service provider for more information.

Electronic Mail Address

Read the literature that came with your service or contact the service provider for the electronic mail address of the device.

For a complete description of EVM, its logger configuration file, and its filtering capabilities, read Chapter 13 of the *System Administration* guide. Also read the `mail(1)` reference page for more information.

EVM Filters

You can tell the EVM daemon precisely which events you are interested in by using an event filter, which is a character string that uses logical language syntax. By using keywords, you can filter by name, priority, host name, and so forth. You must set a filter to be notified of the desired events.

```
"[name sys.unix.evm]"
```

The previous example is a filter that selects events whose names begin with the components `sys.unix.evm`. The left and right square brackets (`[]`) are required. The filter language supports several keywords, logical grouping using parentheses, and a full set of logical operators that you can

specify symbolically (&, |, or !; meaning, *and*, *or*, and *not*, respectively). Keywords can be abbreviated.

See the `EvmFilter(5)` reference page for more information and a description of available keywords.

EVM Logger Configuration File

The logger configuration file `/etc/evmlogger.conf` is a text file that configures the display, forwarding, or storage of events for the EVM logger. All events meeting the specifications of an `eventlog` statement in the configuration file are written to the specified event log or device. This Best Practice describes how to add forward entries to the `/etc/evmlogger.conf` file to forward these events as formatted text.

See the `evmlogger.conf(4)` reference page for more information.

Using Templates with `Evmshow`

You can use templates with `evmshow` to select the information you want to see about each event and to format the display of the information. By using templates with `evmshow`, you can ensure that the event notifications will be within the maximum number of characters you can display on your device.

```
evmshow -t "@timestamp [@priority] @@"
```

In the previous example, `evmshow` replaces `@timestamp` with the time at which the event took place, `@priority` with the priority level of the event, and `@@` with the event's formatted text as specified in the `format_specifier` of the event template file. The example output appears as follows:

```
05-Jul-2000 10:10:44 [200] EVM: Mark event
```

See the `EvmEvent(5)` reference page for the complete list of data item specifiers, the `evmshow(1)` reference page for more information about show templates, and the `evmtemplate(4)` reference page for more information about EVM template files.

Applying the Best Practice

Before you apply this Best Practice, be sure to follow the recommendations in *Before You Begin*.

1. Either log in as root or enter the `su` command to gain superuser privileges.

```
> su
```

Read the `su(1)` reference page for more information.

2. Create a Bourne shell script similar to the following that can receive a single EVM event from its `stdin` stream, format it, and mail it to your device. The following examples use `page_me` as the file name.

The script uses a template with `evmshow` and the `mail` command, which uses the messaging service's electronic mail account address.

```
#!/bin/sh

string=`evmshow -t "EVM alert [@host_name]: @@"`
(echo Subject: $string
 echo $string
 ) | mail 1234@cellphone.com
```

The previous example executes the `evmshow` command, formats the information using a template, and assigns the result to the variable `string`. Note the use of back quotes (```) in the definition.

The `evmshow` command replaces the data item specifier, `@host_name`, with the literal host name of the system on which the event took place. It also replaces `@@` with the event's formatted text.

The Bourne shell script then executes two `echo` commands. The first creates a `Subject` line using the information assigned to the `string` variable for the text of the subject. The second repeats the `string` variable for the message body text. Some paging services only will accept text from the subject of the mail message and ignore the body text; therefore, you must format the text of the notification into a subject line.

The script pipes the resulting message through the `mail` command, which sends it to your messaging service. You must replace the example electronic mail address, `1234@cellphone.com`, with that of the device. You can use your own electronic mail address for testing purposes.

Read the `evmshow(1)`, `mail(1)`, and `echo(1)` reference pages for more information.

3. Change the permission of the Bourne shell script you just created with the `chmod` command so that it is executable.

```
# chmod 744 page_me
```

Read the `chmod(1)` reference page for more information.

4. Check the contents of the `/etc/evmlogger.conf` file to see if there is an existing `forward` entry. If there is, it is recommended that you copy it and edit the copied entry rather than edit the original.

```
forward {
  name page_me [a]

  # Don't forward mail events through mail
  filter "[prio >= 600] & ![name @SYS_VP@.syslog.mail]" [b]

  suppress [c]
  {
    filter "[name *]"
    period 120 # minutes
    threshold 1 # No. of duplicates before suppression
  }
  command "full_directory_path/page_me" [d]
}
```

- a. Any name can be supplied. [Return to example]
 - b. This line posts any event that has a priority equal to or greater than 600 and is not a mail event. An event with a priority of 600–699 is an Alert and an event with a priority of 700 is an Emergency. You must filter out mail events because the forwarding command makes use of the mail system. If the mail system encounters a problem, it might post a high priority event. This can cause an endless event loop if you continue to forward high priority mail events through the mail system. [Return to example]
 - c. You can suppress duplicate events to prevent unnecessary duplicate notifications. This example suppresses any event that has occurred twice within 120 minutes. [Return to example]
 - d. You must provide the full pathname of the Bourne shell script you created in *Step 2*. [Return to example]
5. Instruct the logger to reload the `/etc/evmlogger.conf` file:

```
# evmreload -l
```

Reloading the logger configuration file causes EVM to begin using the new configuration. Note that you must enter `evmreload -l` every time you modify the `/etc/evmlogger.conf` file.

Read the `evmreload(8)` reference page for more information.

Verifying Success

After you apply the Best Practice for sending selected events to a cellular phone or pager, you can verify whether it was successful.

1. In *Step 2 of Applying the Best Practice*, use an electronic mail address with which you can test the notification or use the address of an available cellular phone or pager.
2. Create an event with a priority higher than the minimum priority entered in `/etc/evmlogger.conf`.

```
# evmpost -a "Test Message" -p 700
```

The previous example posts an administrator's quick message and assigns it a priority of 700. Since the example filter used in *Applying the Best Practice* selects events with a priority of 600 or greater, this test event meets the selection criteria and a mail notification is sent.

3. Create an event that matches the minimum priority entered in `/etc/evmlogger.conf`.

```
# evmpost -a "Test Message" -p 600
```

The previous example posts an administrator's quick message and assigns it a priority of 600. Since the example filter used in *Applying the Best Practice* selects events with a priority of 600 or greater, this test event meets the selection criteria and a mail notification is sent.

4. Create an event with a priority that is less than the minimum priority entered in `/etc/evmlogger.conf`. This verifies whether it successfully filters out events that do not match.

```
# evmpost -a "Test Message" -p 599
```

The previous example posts an administrator's quick message and assigns it a priority of 599. Since the example filter used in *Applying the Best Practice* selects events with a priority of 600 or greater, this test event does not meet the selection criteria; therefore, a mail message is not sent.

Read the `evmpost(1)` reference page for more information.

5. Check your mail program or device for notifications of the two events that match the filter criteria.

If the Best Practice was not successful, see *Troubleshooting* for information about identifying and solving problems.

Troubleshooting

If you determine that the Best Practice was not successful, as described in *Verifying Success*, use the following table to identify and solve problems:

Problem	Possible Solutions
The event neither appeared on the system console nor sent mail.	<ul style="list-style-type: none">• Enter the <code>evmreload -l</code> command to reload the logger file.• Check to see if the event matches the filter parameters in <code>/etc/evmlogger.conf</code>.• Check the values entered in the <code>suppress</code> entry in the <code>/etc/evmlogger.conf</code> file to ensure that the event has not been repeated within the given time period. For testing purposes, temporarily comment out the <code>suppress</code> entry by entering a pound sign (#) at the beginning of each <code>suppress</code> line and then enter <code>evmreload -l</code> to reload the logger file.
The event appeared on the console but no mail was sent.	<ul style="list-style-type: none">• Ensure that the Bourne shell script file is executable.• Select the option to retrieve new messages in your mail viewer.• Ensure that the <code>command</code> entry in <code>/etc/evmlogger.conf</code> calls out the correct name of the shell script.• Ensure that the Bourne shell script includes the correct electronic mail address.• Enter the <code>evmreload -l</code> command to reload the logger file.
The event message was not received on the message device.	<ul style="list-style-type: none">• There may be a delay before the device receives the notification. Wait for a few minutes for the notification to be received.• Enter the <code>evmreload -l</code> command to reload the logger file.

Comments and Questions

We value your comments and questions on the information in this document. Please mail your comments to us at this address:

best_practices@zk3.dec.com

Legal Notice

COMPAQ and the Compaq logo are registered in the U.S. Patent and Trademark Office.

Confidential computer software. Valid license from Compaq required for possession, use, or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendors standard commercial license.

Compaq shall not be liable for technical or editorial errors or omissions contained herein. The information in this publication is subject to change without notice and is provided "as is" without warranty of any kind. The entire risk arising out of the use of this information remains with recipient. In no event shall Compaq be liable for any direct, consequential, incidental, special, punitive, or other damages whatsoever (including without limitation, damages for loss of business profits, business interruption or loss of business information), even if Compaq has been advised of the possibility of such damages. The foregoing shall apply regardless of the negligence or other fault of either party and regardless of whether such liability sounds in contract, negligence, tort, or any other theory of legal liability, and notwithstanding any failure of essential purpose of any limited remedy.

The limited warranties for Compaq products are exclusively set forth in the documentation accompanying such products. Nothing herein should be construed as constituting a further or additional warranty.