# Tru64 UNIX Event Management

## DECUS Symposium 2005

Reinhard Stadler
Technical Consultant

## Agenda

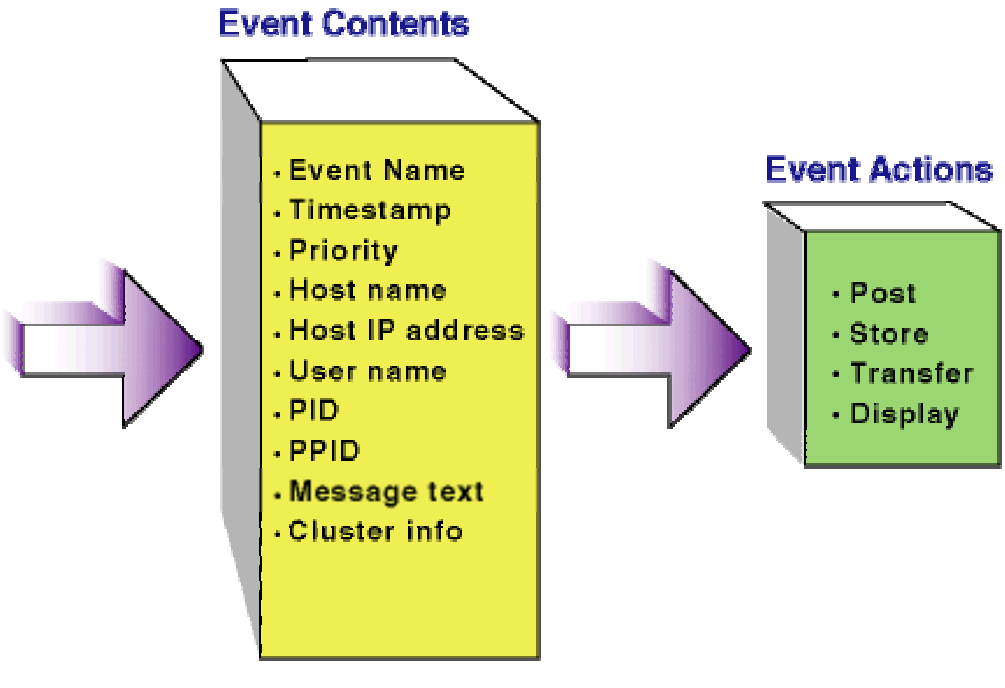
- EVM Overview
- Retrieving and Viewing Events
- Configuring EVM
- Filtering and Forwarding

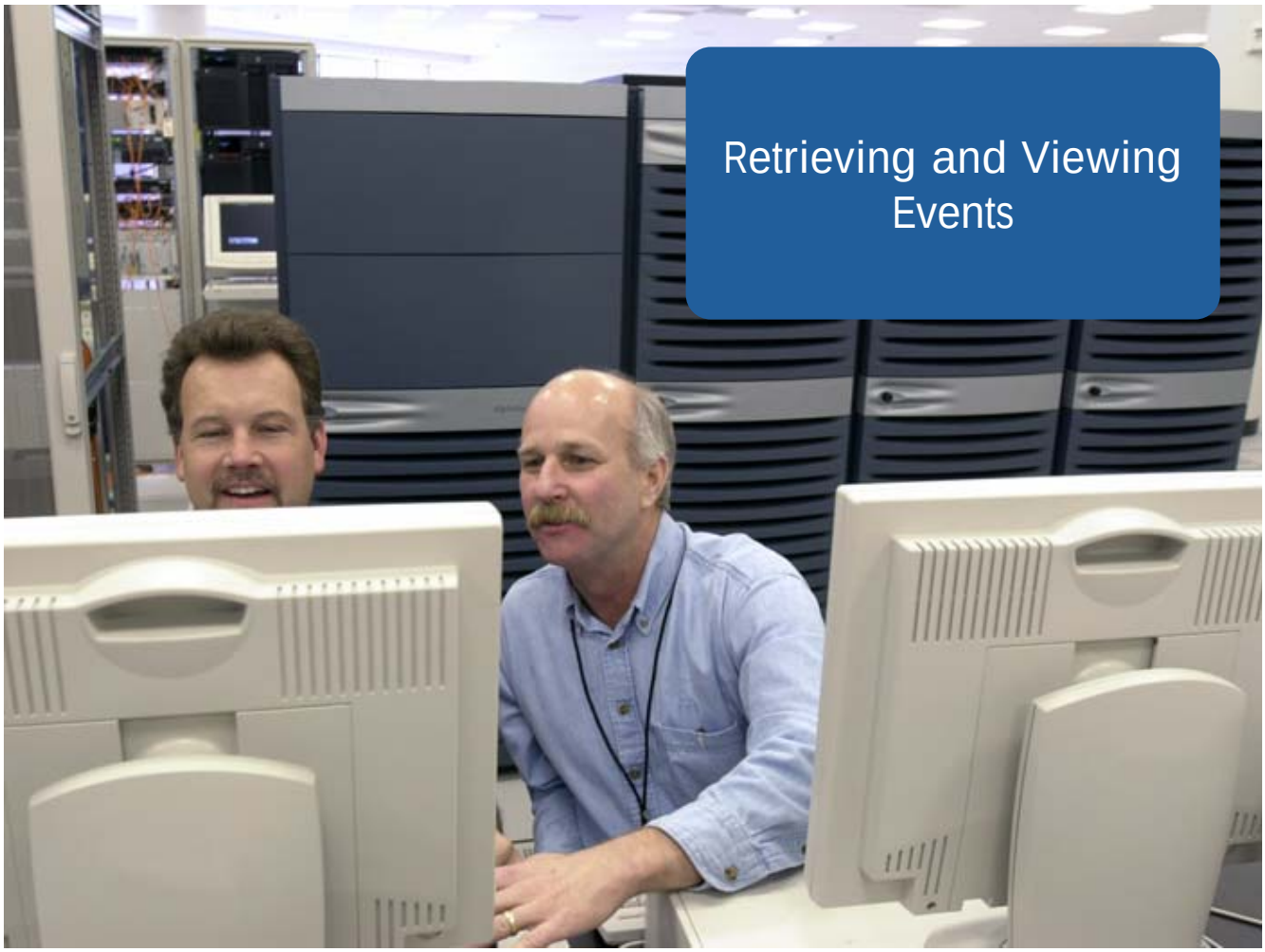
# EVM Features

- EVM provides a single point of event information supplementing rather than replacing the traditional UNIX log files
  - Full set of command-line utilities that you can use to post and handle events
  - Integration of a graphical event viewer with the SysMan application suite
  - Events can be selected or filtered
  - Support for all event channels, including syslog and binlog
  - Configurable notification on occurrence of specific events



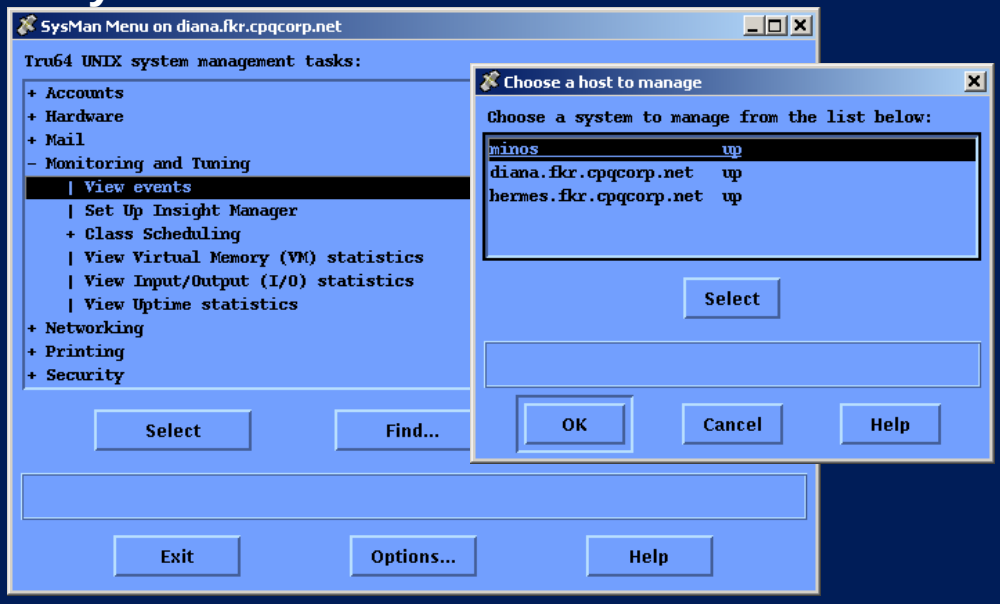
# Event Model



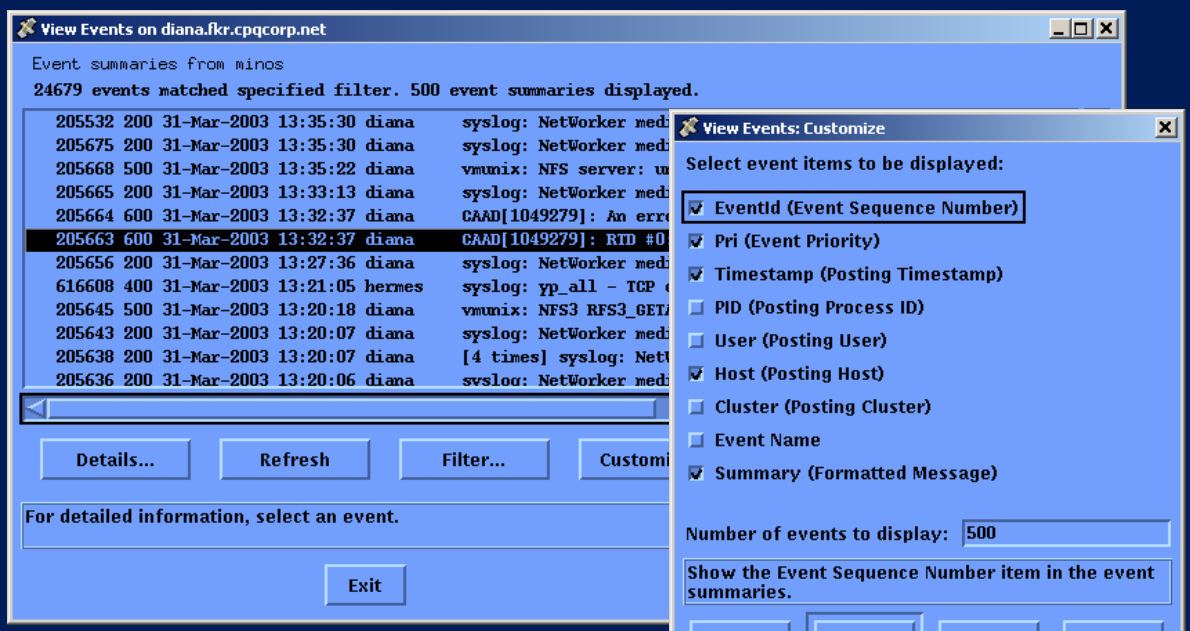



# Using the SysMan Event Viewer

```
# sysman &
```

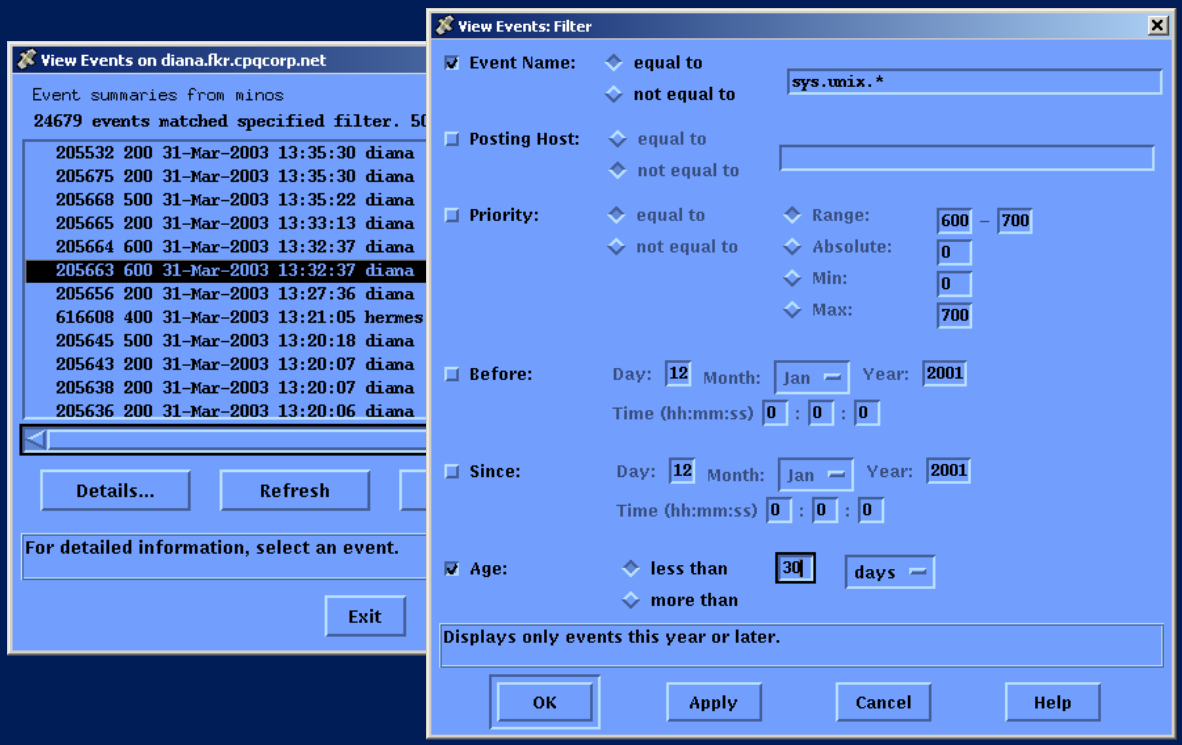


# Using the SysMan Event Viewer

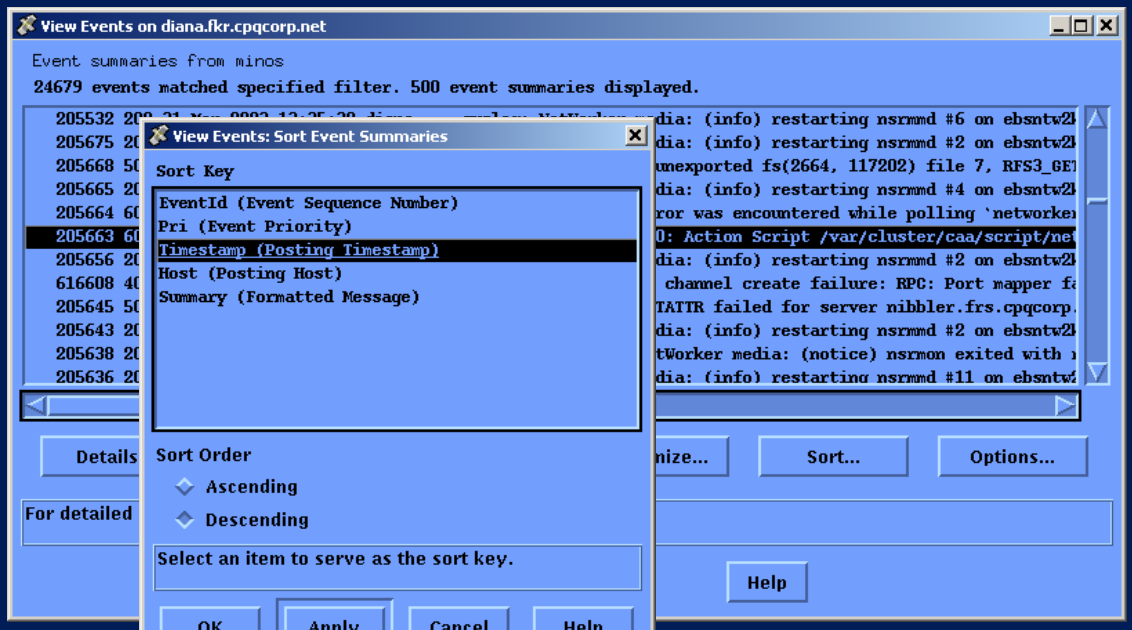





# Using the SysMan Event Viewer

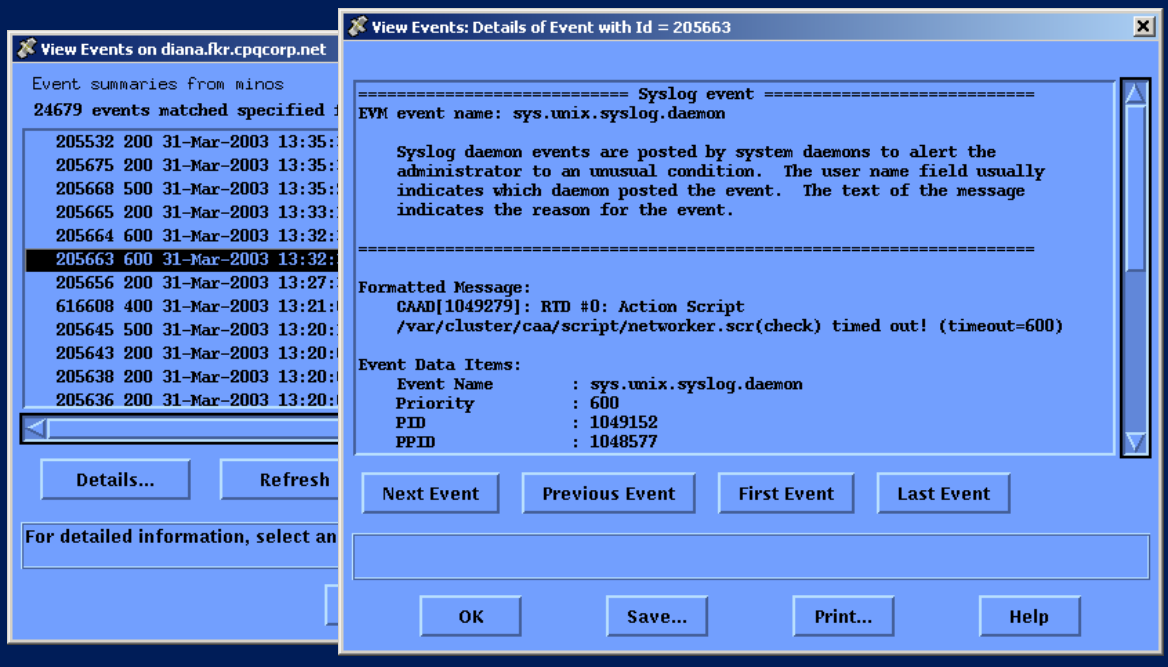


# Using the SysMan Event Viewer






# Using the SysMan Event Viewer



# EVM Command-Line Utilities

- evmget — Retrieves stored events
- evmshow — Accepts one or more EVM events and outputs them in the requested format
- evmsort — Reads a stream of events and sorts them according to supplied criteria
- evmwatch — Outputs specified events as they arrive
- evmpost — Accepts a file or stream of text event sources and posts them to the EVM daemon for distribution

- most common command sequence for event retrieval

```
evmget | evmsort | evmshow
```



# Retrieving Stored Events: evmget

```
evmget [ -A [-t show-template]
  [-s sort-spec] ]
  [-f filter-string]
  [-C channel-list]
```

```
-f filter_expr          Outputs only events
                        that match the
                        filter_expr
```

```
# evmget –f "[ priority >= 600 ]"
```

# EVM event filters

- A specification of a set of interesting events
- Format: `" [keyword expr]"`

- Possible values for keyword are:

```
Name
Priority (from 0 to 700)
Timestamp
Age
BEFore
SINce
Event_ID
...
```



# Complex EVM event filters

- A complex filter is composed of two or more simple filters
- Event filters can be direct or indirect

```
@filename:filtername
```

- Examples

```
"[name sys.unix.binlog] & [pri >= 500]"
"[name sys.unix.syslog] and
 [time 2005:4:5:*:*:*:*]"
"@sys:advfs"
```

the filter named advfs contained in a filter file named sys or sys.evf in `/usr/share/evm/filters`

# Displaying Events Using evmshow

- evmshow accepts one or more EVM events and outputs them in the requested format

```
# evmshow [[-d | -D | -x ]
    [-t show-template]
    [-T timespec] | -r]
    [-c config-file]
    [-f filter-expr] [-F]
    [-k skip-count] [-n show-count]
    [filename | -]
```



# EVM show template

- a string that may contain event data item specifiers of the form `@item_name[%width]`
  - `@timestamp`
  - `@priority`
  - `@name`
  - `@@` (the event's formatted text)
  - E.g. `"@timestamp [@priority] @name @@"`
- EVM_SHOW_TEMPLATE environment variable

## Sorting Events Using evmsort

- Reads a stream of events and sorts them according to supplied criteria

```
evmsort [-s sort-spec]
[-A [-t show-template]]
[filename | -]
```

- The sort order can be specified by supplying a

```
sort_spec
@key_item[+|-] [ :@key_item[+|-] ]
```

`key_item` is the name of any EVM standard data item



## Using the -A Option to Simplify the Command String

- most common command sequence for event retrieval
  **`evmget | evmsort | evmshow`**

**`# evmget -A`**

automatically pipes the output to other EVM commands

```
# evmget -f '[pri >= 600]' | evmsort –s   \
  "priority-:timestamp+" | evmshow | more

# evmget -A -f '[pri >= 600]' -s      \
  "priority-:timestamp+" | more
```

# Monitoring Events Using evmwatch

**# evmwatch**

Outputs specified events as they arrive

```
evmwatch [-A] [-f filter_expr]
[-t show_template]
```

```
# evmwatch –i |                          \
  evmshow –t "@name @priority @@"
```



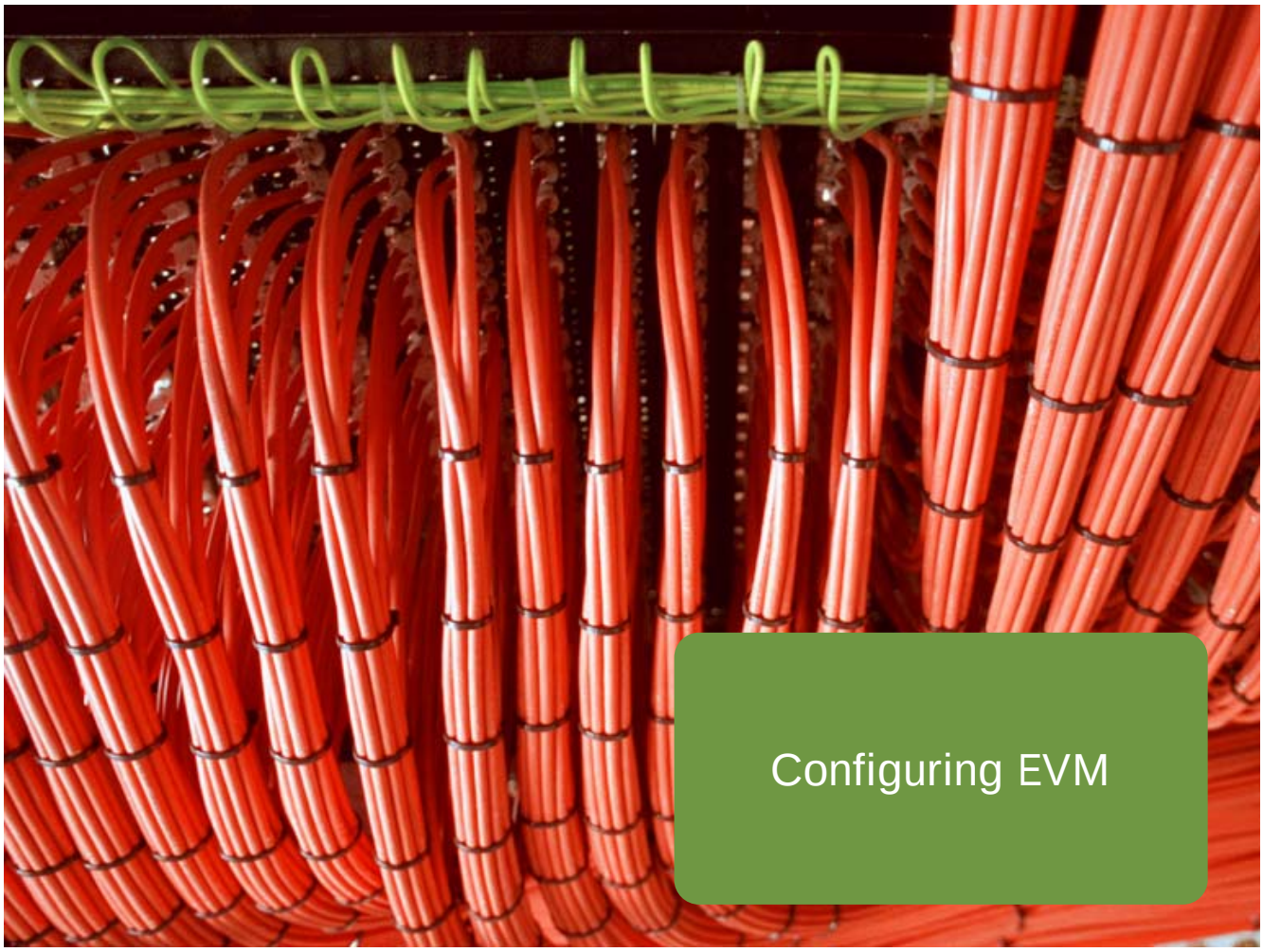
# Posting Message Events Using evmpost

**# evmpost**

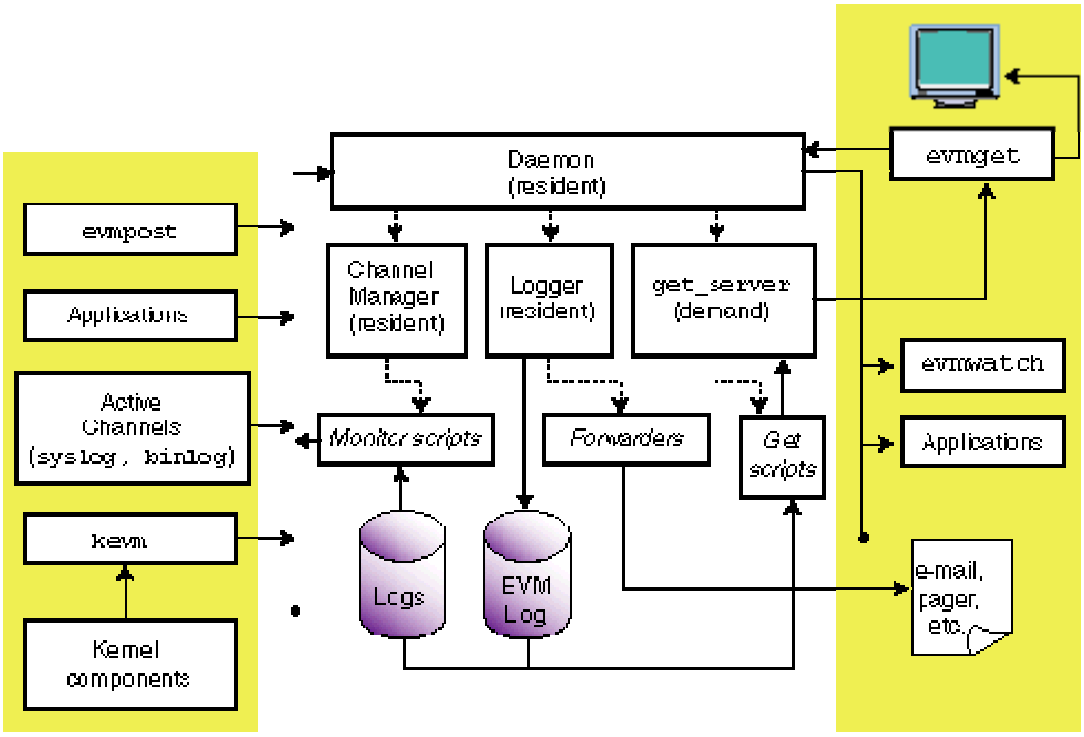
Accepts a file or stream of text event sources and posts them to the EVM daemon for distribution

```
# evmpost [ –f <file> | - ]
```

```
# echo 'event {name ... }' | evmpost
```




# EVM Component Model



# The EVM daemon

- The primary component of EVM
  - Starts the logger
  - Starts the channel manager
  - Receives events from posters and distributes them to subscribers
- Default configuration file /etc/evmdaemon.conf



# The Channel Manager

- Runs channel monitor scripts and cleanup scripts
- For each channel a selection of functions may be defined
  - fn_get — invoked by `evmget_srv`
  - fn_details — used by evmshow -d
  - fn_explain — used by evmshow -x
  - fn_monitor — monitors the status of a channel
  - fn_cleanup — archives or purges logs
  - mon_period — monitor period
- Configuration file /etc/evmchannel.conf

# The Logger

- Runs as a resident process
- Subscribes to a selected set of events
  - Stores them in managed log files for later retrieval
  - Writes high-priority events to the system console
  - Forwards selected events
    (e.g. sends mail to the system administrator when high-priority events occur)
- configuration file /etc/evmlogger.conf



# Forwarding Events

- Specific events can trigger any user difined action
  (e.g. alarms, scripts, …)
- „forward“ statement in the evm logger configuration
- Should be placed in /var/evm/adm/config/logger
- Example:
  trigger a script if NIFF detects a NW interface failure
- netrain.conf

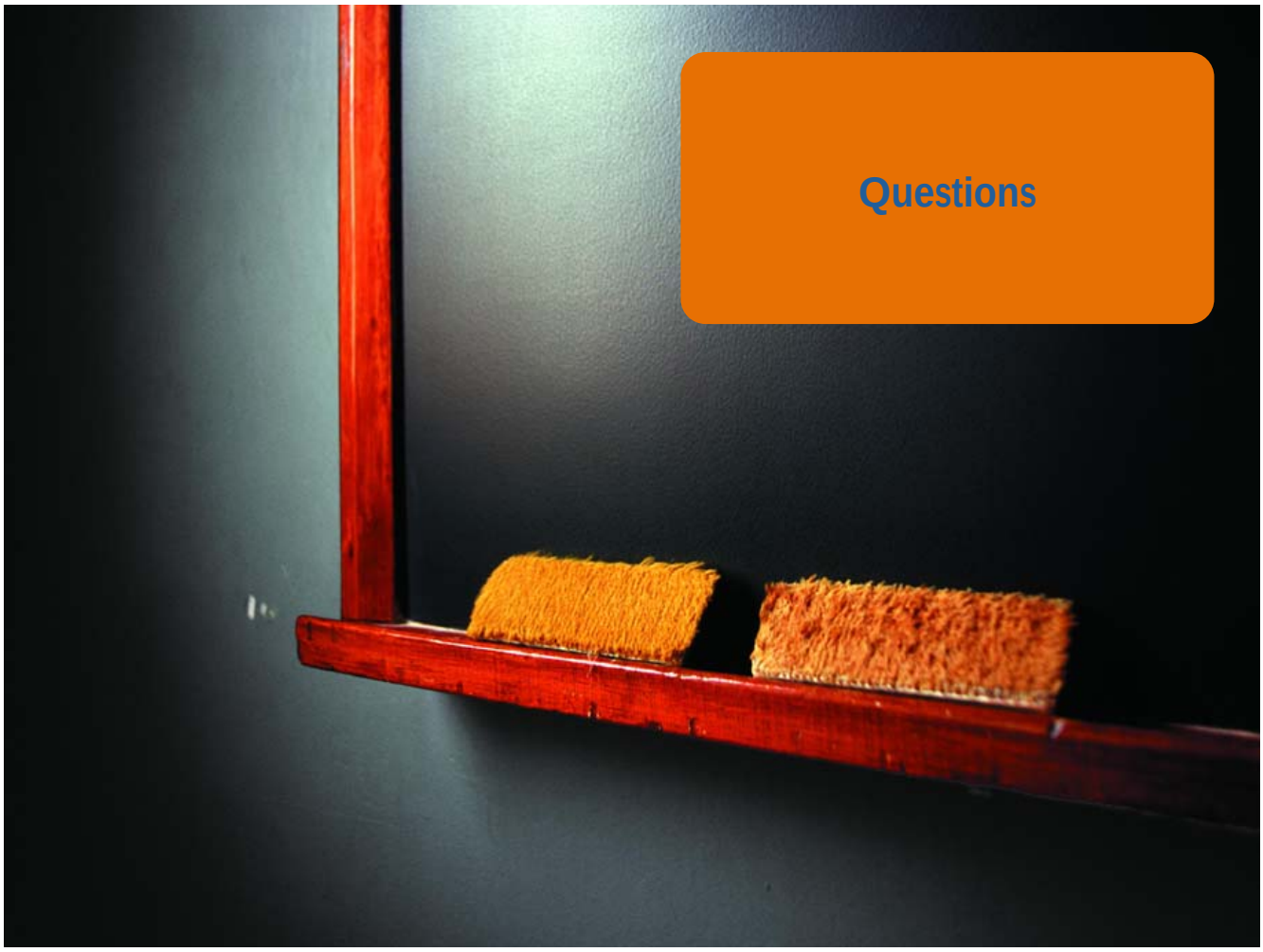
# EVM System Files

- /var/evm/evmlog event logs
  evmlog.yyyymmdd[_nn]
- System-supplied definition files:
  - /usr/share/evm/channels
  - /usr/share/evm/filters
  - /usr/share/evm/templates
- Installation of new definition files:
  - /var/evm/adm/templates
  - /var/evm/adm/channels
  - /var/evm/adm/filters
- Secondary logger configuration files
  - /var/evm/adm/config/logger



# Define a new Event

- Create a template file to define the new event(s)
  - The syntax of a template file is identical to the syntax used to post an event
- Save the template file in /var/evm/adm/templates (create subdirectories if neccesary)
- Instruct the EVM daemon to reload its configuration
- Verify template registration by using the **`evmwatch -i`**
- You can now post your new event(s) and retrieve it from storage


Questions


invent