Event Task Configuration User Guide

Version 1.1

2008/01/28

www.vivotek.com

Event Task Configuration User Guide

VIVOTEK may make changes to specifications and product descriptions at any time, without notice.

The following is trademarks of VIVOTEK INC., and may be used to identify VIVOTEK products only: VIVOTEK. Other product and company names contained herein may be trademarks of their respective owners.

Revision History

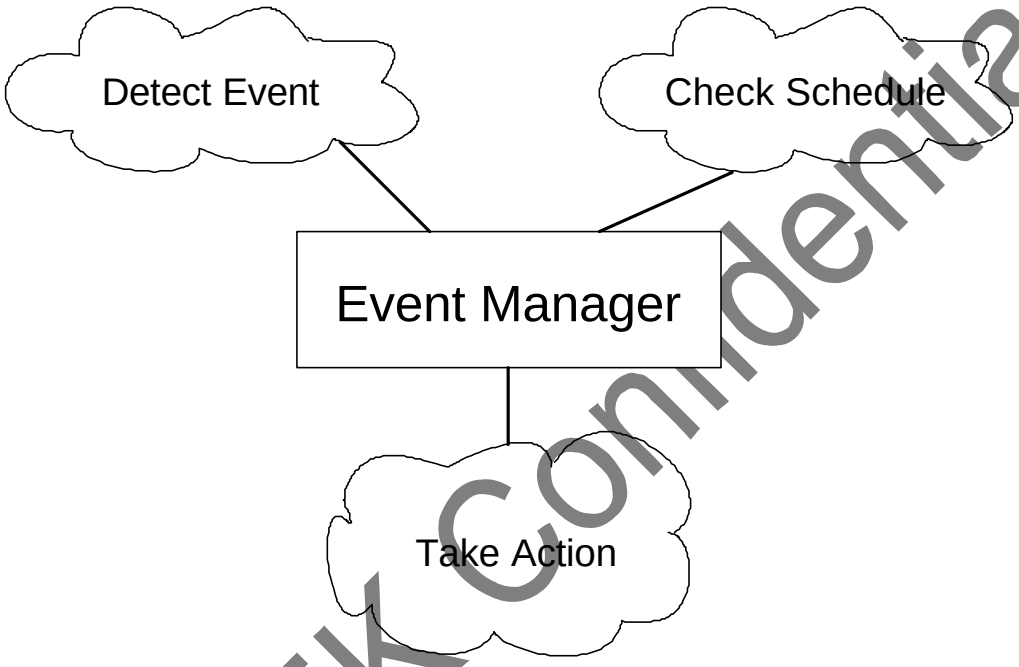
Version	Issue date	Editor	Comment
1.0	2007/12/31	Mei-Yun Hsu	First draft
1.1	2008/01/28	Mei-Yun Hsu	Modify conf file parameter

Table of Contents

Introduction

The event manager is a daemon to schedule tasks based on either time of day or events trigger, e.g. when the motion is detected, or a combination of these. Users can set the tasks to do when the specific conditions he or she required happen. The predefined conditions and scheduled tasks can be recorded in multiples files. So the user can create his event task configuration files for scheduled tasks except the one generated by the system program.

The task configuration file is written in XML format.



1. Activate and Deactivate

- To activate

Please upload the user-defined task configuration files to /mnt/flash/etc/conf.d/event/task by ftp. The user-defined task configuration files would take effect after reboot.

Users can use CGI command, /cgi-bin/admin/eventtask.cgi?method=reload, to test it immediately. When using CGI command, users can check the log to make sure it works. The content of log is "[EVENT MGR]: reload config file".

- To deactivate

Please remove the user-defined task configuration files in /mnt/flash/etc/conf.d/event/task by ftp. Then reboot the server to take effect.

Users can also use CGI command, /cgi-bin/admin/eventtask.cgi?method=reload, to test it immediately. Please check the log to make sure it works. The content of log is "[EVENT MGR]: reload config file".

NOTICE: Please DO NOT remove other task configuration files used by system. Or the application will malfunction!

2. Version Control

There is a "version" attribute in the root element, "eventmgr", in task configuration file. Its value is the version of this task configuration file.

Format:

```
<eventmgr version="xxyy">
  …
</eventmgr>
```

Version format:

Format	Description	Example
xxyy	xx: Major number yy: Minor number	0100

The event manager only supports the task configuration file with the same major number. The event manager supports the task configuration file with version which is equal to or less than the one reported by event manager.

To know the version of task configuration files that event manger supports, please use

```
http://<server-name>/cgi-bin/admin/eventtask.cgi?method=getversion
```

Example of output:

event manager version: 0.1.0.2 supported task version: 0.1.0.1

3. CGI Command

Users can use this CGI command to get version of event manager, reload event manager and execute file.

http://<server-name>/cgi-bin/admin/eventtask.cgi?method=<value>[&file=<value>]

Parameter	Value	Description
method	getversion	Get the version of event manager
	reload	Inform the event manager to reload the task files immediately
	exec	Execute the file indicated by *file* parameter
file	full path of file	Indicate the path of file to execute.

4. Event Type

There are total five event types supported on our server.
Please check the product specification to get the number of DI/DO.

Name	Trigger type*	Number	Status	Description
boot	pulse	1	trigger	The event when the system boot-up. It only occurred once.
motion	level/edge	3	normal, trigger	Motion detection
di	level/edge	Product dependent	normal, trigger	Digital input
do	level/edge	Product dependent	normal, trigger	Digital output
visignal	pulse	Product dependent	loss	Video input signal loss Only for video server.

* Pulse trigger is the trigger without state. Level/edge trigger is the trigger with state.

5. Action Type

When the condition is triggered, users can take the following actions. For example, add message to log or send message to the specified http server.

5.1 FTP

Users can upload files to some specified ftp server.

```
/usr/bin/ftpmput [options] server_address upload_folder source_folder
```

Option	Description
-t SECONDS	Seconds to timeout response wait
-a IPADDR	Use passive mode. IPADDR is the IP address of this server.
-u USERNAME	username for authentication
-p PASSWORD	password for authentication
-P PORT	Assign the port of ftp server if it is not default value, 21.

- Example

Upload snapshot files under /mnt/ramdisk/snapshot_output to folder "Share" on ftp server "172.16.7.22".

```
/usr/bin/ftpmput –t 10 –u root –p 123456 172.16.7.22 Share /mnt/ramdisk/snapshot_output
```

5.2 SMTP (Email)

Users can send email to recipients and attach some files in the mail.

```
/usr/bin/smtpclient [options] recipients
```

Option	Description
-s SUBJECT	Subject line of message
-u FILE	Read subject line from FILE
-f ADDR	Address of the sender
-a FILE	Binary MIME attachment file
-d DIRECTORY	Binary MIME directory attachment
-b FILE	Read mail body from FILE or it reads
-T SECONDS	Seconds to timeout response wait
-S HOST	Host where MTA can be contacted via SMTP
-B HOST	Backup host to use if primary host is unavailable
-P PORT	Port where MTA can be contacted via SMTP
-M NUM	Use MIME-style translation to quoted-printable or base64 1 = application/octet 2 = image/jpeg 3 = plain/text
-L	Log errors to syslog facility instead of stderr
-D	Delete body/subject/attachment files after they're processed

-A NUM	maximum number of attachments per mail split up in several mails if there are more
-U USERNAME	username for authentication
-p PASSWORD	password for authentication

- Example

Send email with title “Captured Media” and attached image files from /mnt/ramdisk/snapshot_output. This mail has no body.

```
/usr/bin/smtpclient -s "Captured Media" -f wang@test.com -S mailserver.test.com -M 2 -T 10 -d
/mnt/ramdisk/snapshot_output wang@test.com < /dev/null
```

Send email with title “Boot up Log” to recipient wang@test.com. The body of mail is the log messages.

```
/usr/bin/smtpclient -s "Boot up Log" -f wang@test.com -b /var/log/messages -S mailserver.test.com
-M 3 wang@test.com
```

5.3 Curl (product dependent)

Users can post data or upload files to some specific http server.

```
/usr/bin/curl [options...] url
```

Please see http://curl.haxx.se/docs/manpage.html for detail.
It only supports BASIC authentication mode on our server.

- Example

Post "Digital Input Changed" data to the http server at 172.16.2.100.

```
/usr/bin/curl -v -u root:pm -d "Digital Input Changed" http://172.16.2.100/cgi-bin/reportevent.cgi
```

Upload log file to the http server at 172.16.2.100.

```
/usr/bin/curl -v -u root:pm -F file=@"/var/log/messages" http://172.16.2.100/cgi-bin/uploadfile.cgi
```

5.4 Http Client (product dependent)

Users can post data or upload files to some specific http server.

```
/usr/bin/httpclient -M <method> [options] -S server_address server_port user_name password
[proxy setting] -U <URI> –r /mnt/flash/httplog
```

Option	Description
-M <method>	Valid values are POST and UPLOAD.

	POST: to post data to the server. UPLOAD: to upload files to the server.
-m MESSAGE	The message to post. It is used in POST method.
-t <type>	Valid values are Jpeg, VideoClip and Text. It is used in UPLOAD method.
-file FILE	The file to upload. It is used in UPLOAD method. This option can not co-exist with *folder* option.
-folder FOLDER	The folder to upload. It is used in UPLOAD method. This option can not co-exist with *file* option.

Proxy setting (optional):

```
-P proxy_address proxy_port user_name password
```

- Example

Post "Digital Input Changed" data to the http server at 172.16.2.100.

```
/usr/bin/httpclient -M POST -m "Digital Input Changed" -S 172.16.2. 100 80 root pm -U
/cgi-bin/reportevent.cgi -r /mnt/flash/httplog
```

Upload log file to the http server at 172.16.2.100.

```
/usr/bin/httpclient -M UPLOAD -t Text -file /var/log/messages -S 172.16.2.100 80 root pm -U
/cgi-bin/uploadfile.cgi -r /mnt/flash/TextLog
```

5.5 Network Storage

To use network storage, please add setting for network storage from server part of application webpage /setup/application.html. In order to limit resource usage, only one network storage is allowed on our server.

To access the network storage on our server, please access */mnt/auto/i[n]*. n is the index of server. Users can know the index of server by using CGI command /cgi-bin/admin/getparam.cgi?server.

For example, the output of /cgi-bin/admin/getparam.cgi?server is as following. Its access point is /mnt/auto/i3.

```
server_i0_name='AGD'
server_i0_type='email'
.....
server_i1_name='F1'
server_i1_type='ftp'
```

```
.....
server_i2_name='D3'
server_i2_type='email'
.....
server_i3_name='Samba'
server_i3_type='ns'
.....
server_i4_name=''
server_i4_type='email'
.....
```

- Example

Assume the index of server of network storage is 3.

Transfer snapshot files under /mnt/ramdisk/snapshot_output to network storage.

```
cp /mnt/ramdisk/snapshot_output/*.jpg /mnt/auto/i3
```

5.5 System Log

Users can send messages to the system logger with the specified priority and TAG.

```
/usr/bin/logger [option] message
```

Option	Description
-p PRI	Enter the message with the specified priority. Valid level names are: alert, crit, debug, emerg, error, info, notice and warning The default is notice.
-t TAG	Mark every line in the log with the specified TAG. The default is “root”.

- Example

Send message “Start to Test” with priority notice.

```
/usr/bin/logger –p info “Start to Test”
```

Here is the example of output:

```
Dec 31 16:57:34    root: Start to Test
```

5.6 Digital Output

Trigger digital output in specific duration.

```
/usr/bin/event_do doindex duration
```

- Example

Trigger digital output 0 for 10 seconds.

```
/usr/bin/event_do 0 10
```

6. Media Type

There are three types of media supported, snapshot, clip and log.

To generate the media contents correctly, users have to setup the sources of media by setup_audio and setup_video scripts in the very beginning.

```
/usr/bin/setup_audio action stream_no apply_to_clip
```

Parameter	Description
action	Its value can be “start” or “stop”.
stream_no	The index of stream source 1: the first stream
apply_to_clip	The action will be apply to clip if it value is 1.

```
/usr/bin/setup_video action stream_no apply_to_clip apply_to_snapshot
```

Parameter	Description
action	Its value can be “start” or “stop”.
stream_no	The index of stream source 1: the first stream 2: the second stream
apply_to_clip	The action will be applied to clip if its value is 1.
apply_to_snapshot	The action will be applied to snapshot if its value is 1.

- Example

To start snapshot capture and stop clip recording of the first stream, please use the following command.

```
/usr/bin/setup_video start 1 0 1
/usr/bin/setup_video stop 1 1 0
```

6.1 Snapshot

Users can capture snapshots when the event is triggered. The number of snapshots before and after trigger can be set by configuration file.

```
/usr/bin/event_media_snapshot action conffile
```

Parameter:

Name	Description
action	Valid values are “start”, “stop”, “trigger” and “wait”. “start” is to start snapshot capture for this instance. “stop” is to stop snapshot capture for this instance. “trigger” is to switch state from pre-trigger to post-trigger. “wait” is to wait the snapshot files ready.
conffile	The configuration file of snapshot. Please use absolute path.

Return value:

On success zero is returned. On timeout, 1 is returned. On incorrect parameters, 2 is returned.

Example of snapshot configuration file:

```
# Snapshot Conf
INSTANCE_NAME="VV_SS0"
STREAM_SOURCE="2"
FILE_NAME="VIVOTEK_%h%m%s"
PRE_TRIG_NUM="1"
POS_TRIG_NUM="2"
FOLDER="/mnt/ramdisk/snapshot_output"
TIME_OUT="4"
```

Parameter	Description
INSTANCE_NAME	The identification for this snapshot. It has to be unique.
STREAM_SOURCE	The index of stream source “1” indicates the first stream “2” indicates the second stream
FILE_NAME	The file name of captured images. The file extension is always “.jpg”. It supports the string composed of alphanumeric characters and the following specific format: “%Y” indicates year in 4 digits “%M” indicates month in 2 digits “%D” indicates day in 2 digits “%h” indicates hour in 2 digits

	"%m" indicates minute in 2 digits "%s" indicates second in 2 digits "%[*width*]S" indicates the sequence number starting from zero. The *width* argument is a positive decimal integer controlling the number of characters of sequence number. The maximum value of *width* is 8. zeros are added until the width is reached. The sequence number starting from 0 from the first pre-trigger images in every trigger.
PRE_TRIG_NUM	Number of images in the pre-trigger buffer.
POS_TRIG_NUM	Number of images in the post-trigger buffer.
LOCATION	The output folder where to put captured images. Please REMOVE this folder after the images are processed.
TIME_OUT	The timeout to wait all images are captured in seconds. Suggest to set timeout as the value of POS_TRIG_NUM*2

Once these images are sent out, please remove the output folder. Then the image buffering will go back to pre-trigger state.

6.2 Media Clip

Users can record clips. Two modes of recording are supported. One is trigger mode and the other is sequential mode. In trigger mode, four actions are supported. They are "start", "stop", "write" and "wait". Once it starts, it will keep recording in pre-event status. When the event is triggered, users use "write" action to inform the recorder to generate clip file. Then users use "wait" action to wait the clip file ready. In sequential mode, only "start" and "stop" actions are supported. Once it starts, it keeps output clip files.

Users can use this command to setup the clip.

```
/usr/bin/event_media_clip action [mode] conffile
```

Parameter:

Name	Description
action	In trigger mode, valid values are "start", "stop", "write" and "wait". In sequential mode, valid values are "start", "stop". "start" is to start clip recording for this instance. "stop" is to stop clip recording for this instance. "write" is to write out clip file in trigger mode. "wait" is to wait the clip file ready in trigger mode.
mode	When the action is "start" or "stop", its value can be "event" or "seq".

	“event” indicates event recording. “seq” indicates sequential recording.
conffile	The setting of clip. Please use absolute path.

Return value:

On success zero is returned. On timeout, 1 is returned. On incorrect parameters, 2 is returned.

Example of clip configuration file:

```
# Media Clip Conf
INSTANCE_NAME="VVTK_VC0"
STREAM_SOURCE="1"
VIDEO_ENABLE="1"
AUDIO_ENABLE="1"
CLIP_LENGTH="5"
MAX_FILE_SIZE="1024"
FILE_NAME="VIVOTEK_%h%m%s"
PRE_EVENT_TIME="2"
LOCATION="/mnt/ramdisk/videoclip_output"
APP_NAME="TEST"
TIME_OUT="10"
```

Parameter	Description
INSTANCE_NAME	The identification for this clip. It has to be unique.
STREAM_SOURCE	The index of stream source “1” indicates the first stream. “2” indicates the second stream.
VIDEO_ENABLE	To record the video if its value is “1”.
AUDIO_ENABLE	To record the audio if its value is “1”.
CLIP_LENGTH	The length of video clip in seconds The clip will be generated when one of limitations, CLIP_LENTH and MAX_FILE_SIZE is reached.
MAX_FILE_SIZE	The maximum size of clip file in Kbytes The clip will be generated when one of limitations, CLIP_LENTH and MAX_FILE_SIZE is reached.
FILE_NAME	The file name of captured clips. The file extension is always “.mp4”. It supports the string composed of alphanumeric characters and

	the following specific format: “%Y” indicates year in 4 digits “%M” indicates month in 2 digits “%D” indicates day in 2 digits “%h” indicates hour in 2 digits “%m” indicates minute in 2 digits “%s” indicates second in 2 digits “%[*width*]S” indicates the sequence number starting from zero. The *width* argument is a positive decimal integer controlling the number of characters of sequence number. The maximum value of *width* is 8. zeros are added until the width is reached. “%*maxvalue*X” indicates the maximum number of sequence number in sequential mode. If the option of sequence number is not enabled, this option is ignored. The value of maximum number should not be over the total digits controlled by *width* argument of sequence number. The *maxvalue* argument is a positive decimal integer. If it is not specified, the default value is 99999.
PRE_EVENT_TIME	Pre-event time in seconds It is used in event mode only.
LOCATION	The location of clip to output
APP_NAME	Assign one for each event It is used in event mode only.
TIME_OUT	The timeout to wait clip ready in seconds. It is used in event mode only. Suggest to set timeout as the value of CLIP_LENGTH+5

The stream source and media content can not be altered after the recording started. To change the setting of recording, please stop it first and start it by new configuration.

6.3 System Log File

The location of system log file is in /var/log/messages.

7. Task Configuration File

Users can describe the trigger conditions what you want and the actions to take when some specific conditions happened in the task configuration file. Users can also limit the total number of process to

run when the events are triggered. Every task configuration file is independent.

There are four parts in a task configuration file. It is written in XML format.

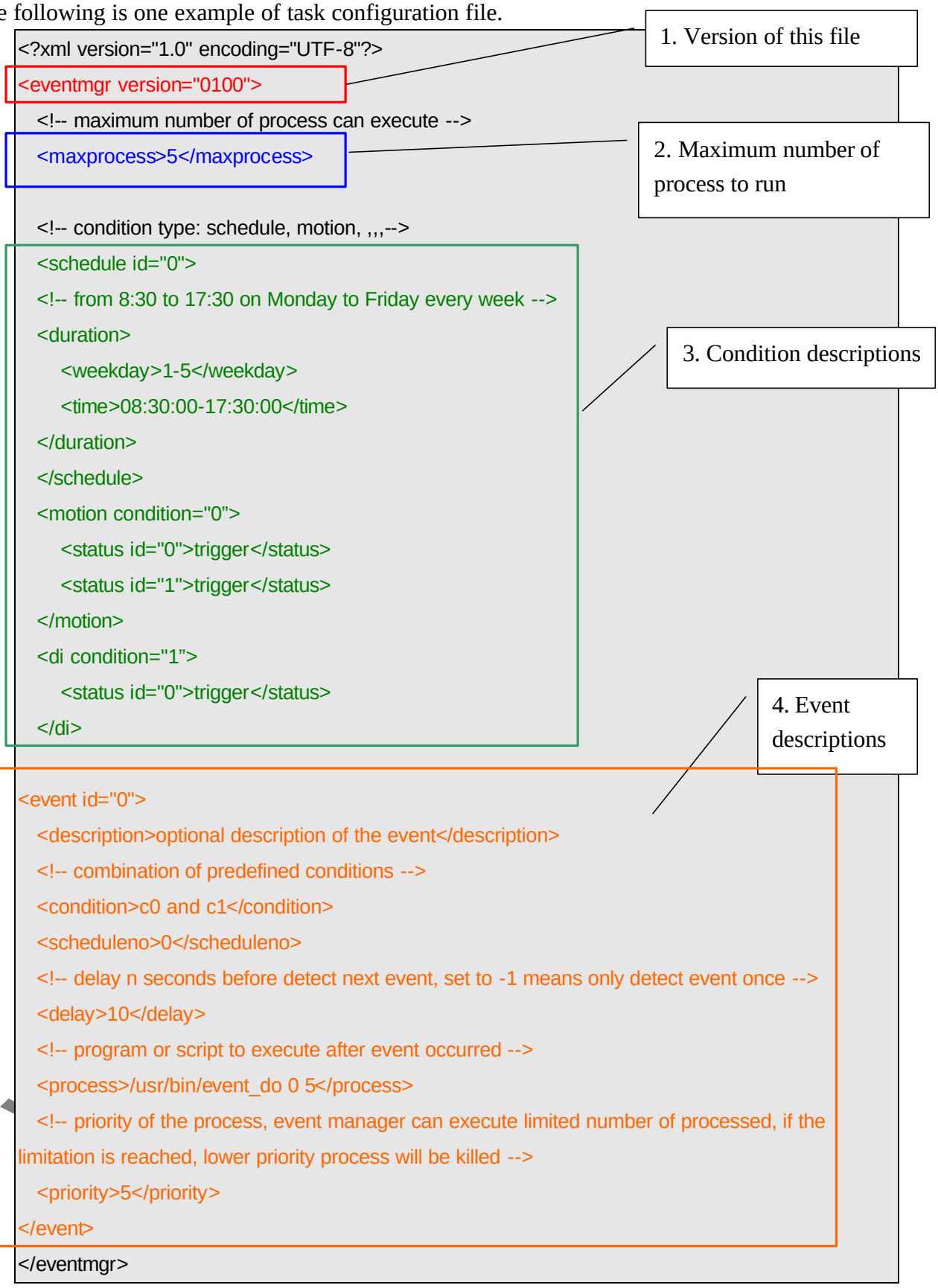
1. The version of the task configuration file
2. The maximum number of process to run

   The event manager will control the number of running processes not to exceed the maximum value. The processes with high priorities will be kept.
3. Condition descriptions

   It includes event condition and/or schedule condition.
4. Event descriptions

   It defines under what conditions the event will be triggered and the tasks to do when the event is triggered. Users can define the event priority and the delay time to detect next event, too.

The following is one example of task configuration file.




```
<?xml version="1.0" encoding="UTF-8"?>
<eventmgr version="0100">
  <!-- maximum number of process can execute -->
  <maxprocess>5</maxprocess>

  <!-- condition type: schedule, motion, ,,,-->
  <schedule id="0">
  <!-- from 8:30 to 17:30 on Monday to Friday every week -->
  <duration>
     <weekday>1-5</weekday>
     <time>08:30:00-17:30:00</time>
  </duration>
  </schedule>
  <motion condition="0”>
     <status id="0">trigger</status>
     <status id="1">trigger</status>
  </motion>
  <di condition="1”>
     <status id="0">trigger</status>
  </di>

<event id="0">
  <description>optional description of the event</description>
  <!-- combination of predefined conditions -->
  <condition>c0 and c1</condition>
  <scheduleno>0</scheduleno>
  <!-- delay n seconds before detect next event, set to -1 means only detect event once -->
  <delay>10</delay>
  <!-- program or script to execute after event occurred -->
  <process>/usr/bin/event_do 0 5</process>
  <!-- priority of the process, event manager can execute limited number of processed, if the
limitation is reached, lower priority process will be killed -->
  <priority>5</priority>
</event>
</eventmgr>
```

7.1 The Version of the Task Configuration File

Eventmgr is the root element and it has a version attribute. The whole task configuration file will be skipped if this root element is missing, the version attribute is missing or the event manager does not support this version. This skip action will be logged.

7.2 The Maximum Number of Process to Run

The total number of running process executed from one task configuration file cannot exceed this maximum number. In the task description part, every task has one priority. All processes listed in one task have the same priority, i.e. the priority of the task. If the running process has reached this limit and there is a new process to execute, the event manager will first check all priorities of running processes and the new process. If the new process had higher priority than one running process, the running process with the lowest priority is terminated and the new process is executed. If the new process had not, the new process will be skipped if its trylater is not set or its value is 0. If the trylater is 1 for this event, it will keep trying until it succeeds to run. The terminated or skipped process is logged.

For tasks with the same priority, the entries listed in front will be run first and terminated last, i.e. the earlier entries precede later entries.

Element name	Type	Description
maxprocess	Integer	Required. The maximum number of running process executed from one task configuration file. Its minimum value is 1. Its maximum value is 12.

For embedded system with limited resource, the recommend value of maxprocess is 4. (motion trigger, sequential, digital input and recording)

NOTICE: The maximum number of process created by event manager for all task configuration files is 12. If the summation of maxprocess of all task configuration files is larger than this value, no process will be run. And this situation will be logged.

7.3 Condition Description

Users can define the condition according to the requirement, e.g. the date or the digital input is rising. The schedule and event condition are numbered independently in zero or positive integer.

7.3.1 Schedule condition

The task manager provides date and time setting for schedule. Users can use it to put the task into

schedule. For example, users can turn on the alarm at PM5: 30 on Monday to Friday in the office and turn it off at AM8: 30 on Sunday to Friday.

In a schedule condition, there is a "*duration*" element containing the detail representation of specific schedule. It is the parent element of date, month, monthday, weekday, time, hour, minute and second. Multiple durations in one schedule condition are supported.

Every schedule condition is numbered in the task configuration file. Every number should be unique. It should be start from zero and increase one for each schedule condition. This number is the value of *id* attribute in every schedule start tag. If there are multiple elements with the same number, the first schedule takes effect and other elements are ignored. The ignored elements are logged.

Format:

```
<schedule id="n">
   <duration>
     …
   </duration>
   <duration>
     …
   </duration>
<schedule>
```

Users can represent the schedule condition in date, month, day of the month, week, day of the week, hour, minute, second and time. The detail format is shown in the table below.

Element name	Type	Period of schedule	Description
date	MM/DD	year	MM indicates the month; DD indicates the day of the month. They are separated by slash.
month	Integer	year	The number of months since January, in the range 1 to 12.
monthday	Integer	month	The day of the month, in the range 1 to 31. (The maximum value depends on the month)
weekday	Integer	week	The number of days since Sunday, in the range 0 to 6.
time	hh:mm:ss	day	hour, minute and second separated by colon. Every value has to be in two digits, for example 08:32:00.
hour	Integer	day	The number of hours past midnight, in the range 0 to 23.
minute	Integer	hour	The number of minutes after the hour, in the range 0 to 59.
second	Integer	minute	The number of seconds after the minute, normally in the range 0 to 59

NOTE: time cannot coexist with (hour, minute, second)

NOTE: date cannot coexist with (month, monthday, weekday)

NOTE: monthday cannot coexist with weekday

Users can use '-'to represent from-to and ',' to separate multiple values. Each element can appear only once in a duration. If there is any invalid value or symbol in the duration, the schedule condition containing this duration is invalid and ignored. The incorrect duration and schedule condition will be logged.
The maximum length in the element is 63 bytes. NO SPACE is allowed.

The schedule condition cannot use for logical combination for event condition. Users can only select one schedule and fill its id value in the content of element named "scheduleno" in event description section.

7.3.2 Example of Schedule Condition

```
<schedule id="0">
<!-- from 8:30 to 17:30 on Monday to Friday every week -->
<duration>
        <weekday>1-5</weekday>
        <time>08:30:00-17:30:00</time>
</duration>
<!-- 4 to 12 o'clock on March 6 and April 3 every year -->
<duration>
        <date>3/6, 4/3</date>
        <hour>4-12</hour>
</duration>
</schedule>

<schedule id="1">
<!-- every 0 and 30 second -->
<duration>
        <minute>0-59</minute>
        <second>0-30</second>
</duration>
</schedule>

<schedule id ="2">
<!-- January to June -->
<duration>
        <month>1-6</month>
</duration>
</schedule>
```

7.3.3 Event Condition

Every event condition is numbered in the task configuration file. The number should be unique. It should begin from zero and increase one for each event condition. This number is the value of *condition* attribute in every event start tag. If there are multiple events with the same number, the first event takes effect and other events are ignored. The ignored events are logged.

Format:

```
<name_of_event_type condition=”n”>
   …
</name_of_event_type>
```

Please check section 3 “Event Type” for the type of event.
n is the integer start from zero, unique for different event condition.

- Pulse or Level Trigger

If the event is pulse-triggered or level-triggered, the content in the task configuration file is as follows,

```
<name_of_event_type condition=”n”>
   <status id=”k”> one_possible_status </status>
</name_of_event_type>
```

The attribute of *id* in the status element is necessary. And it indicates the index of event source and it starts from zero.

For example, this condition became triggered when 1st or 2nd motion detection is triggered.

```
<motion condition=”2”>
<status id=”0”>trigger</status>
<status id=”1”> trigger </status>
</motion>
```

- Edge Trigger

If the event is edge-triggered, the content in the task configuration file is as follows,

```
<name_of_event_type condition=”n”>
   <status id=”k”> one_possible_status~another_possible_status </status>
</name_of_event_type>
```

The attribute of *id* in the status element is necessary. And it indicates the index of event source and it starts from zero.

For example, this is the condition of rising digital input 0.

```
<di condition=”1”>
<status id=”0”> normal~trigger</status>
</di>
```

7.4 Event Description

This section describes that the event will be triggered under what kinds of conditions and the tasks to do when the event is triggered. When multiple events are arranged, you can assign these events different priorities to discriminate their significance. The task of the event with the highest priority will be run first. The task can be assigned is an executable file in the local file system, e.g. a binary or script. You can assign more than one task to run in one event and they are executed in sequence. In order to control the interval of the event trigged, the 'delay' value of event is required.

Every event is numbered and every number is unique. It starts from zero and increase one for each event. This number is the value of *id* attribute in *event* start tag. Only the first event takes effect if there are many events with the same id. Other events are ignored and logged.

The elements in the event are described in the table.

Element name	Type	Description
condition	String	Optional. Logical combination of event conditions. Support three kinds of logical operators: and, or, not. The condition is identified with its number.
scheduleno	Positive integer	Optional. If specified, it should be one of the number id of schedule listed in the file. If not specified, it means always. Adding '!' before the number indicates the inverse scheduling, i.e. the time except the schedule.
delay	0, -1, -2 and positive integer	Required. Delay n seconds before detect next event. Setting as -1 means only detect event once. Setting as -2 means run once when entering the valid interval every time.
process	String	Required. The program or script to execute after event occurred. The maximum length is 255 bytes. The maximum number of argument is 12 including the program filename.
description	String	Optional. The description of the event.
priority	Positive integer	Optional. The minimum value is 0, the lowest priority.

		The maximum value is 255, the highest priority. If not specify, it is the lowest priority.
trylater	0, 1	Optional. It indicates the behavior when the event is triggered again but the process run by the previous triggered has not finished yet. If this element didn't exist, the new process is executed. If this element existed and its value is 0, nothing happens and this event will be ignored. If this element existed and its value is 1, this event will be recorded and try later. It will keep trying until succeed.

If there is any invalid value in the elements, the event is ignored and recorded in the log.

Here is an example of event description.

```
<event id="1">
   <description>suspicious movement</description>
   <condition>c1 and c2</condition>
   <delay>10</delay>
   <process>/usr/bin/event_do 0 2</process>
   <priority>10</priority>
</event>

<event id=”2”>
      <condition>(c1 or c2) and c3</condition>
      <scheduleno>2</scheduleno>
      <delay>-1<delay>
      <process>/usr/bin/ftpmput -u root -p 123456 172.16.2.22 Share /var/log/message</process>
      <process>/usr/bin/event_do 0 10</process>
      <trylater>1</trylater>
</event>
```

8. Log

The Linux system logging utility is used. Please check the log file whether there is any error.

LOG_LEVEL	Condition
ALERT	1. The whole task configuration file is skipped. 2. If the summation of maxprocess of all task configuration files is larger than 10.
ERROR	1. Access error of task configuration file 2. Invalid elements or values in task configuration file 3. Duplicate numbers in task configuration file

WARNING	1. When the process to run is skipped. 2. When event manager terminates the running process.

9. Example

To try these examples, please upload task configuration files to folder /mnt/flash/etc/conf.d/event/task and other scripts and configuration file to folder /mnt/flash. Then reboot the server to load the new setting.

9.1 Setup Required Media in the Beginning and Email System Log When the System Boots Up

Suggest to add the section in blue color in one of your task configuration file to setup media.
Task configuration file:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<eventmgr version="0101">
<maxprocess>1</maxprocess>
<boot condition="1"><status id="0">trigger</status></boot>
<event id="0">
   <description>Prepare Media</description>
   <priority>255</priority>
   <delay>-1</delay>
   <process>/mnt/flash/setup_media</process>
</event>
<event id="1">
   <description>Boot Up</description>
   <condition>c1</condition>
   <delay>10</delay>
   <trylater>1</trylater>
   <priority>2</priority>
   <process>/usr/bin/smtpclient -s “BOOT UP NOTIFY” -f wang@test.com -b /var/log/messages -S
   mailserver.test.com -M 3 wang@test.com</process>
</event>
</eventmgr>
```

Example of script to setup media:

```
#!/bin/sh
# setup_media
```

```
# Stop previous setting if necessary
/usr/bin/event_media_clip stop pre /mnt/flash/media_clip.conf
/usr/bin/event_media_snapshot stop /mnt/flash/media_snapshot.conf

# Start new setting
/usr/bin/event_media_clip start pre /mnt/flash/media_clip.conf
/usr/bin/event_media_snapshot start /mnt/flash/media_snapshot.conf

# Enable audio clip
/usr/bin/setup_audio start 1 1

# Enable video stream 1 clip
/usr/bin/setup_video start 1 1 0

# Enable video stream 2 snapshot
/usr/bin/setup_video start 2 0 1
```

Example of clip configuration:

```
# Media Clip Conf
INSTANCE_NAME="VVTK_VC0"
STREAM_SOURCE="1"
VIDEO_ENABLE="1"
AUDIO_ENABLE="1"
CLIP_LENGTH="5"
MAX_FILE_SIZE="1024"
FILE_NAME="VIVOTEK_%h%m%s"
PRE_EVENT_TIME="2"
LOCATION="/mnt/ramdisk/videoclip_output"
APP_NAME="TEST"
TIME_OUT="10"
```

Example of snapshot configuration:

```
# Snapshot Conf
INSTANCE_NAME="VV_SS0"
STREAM_SOURCE="2"
FILE_NAME="VIVOTEK_%h%m%s"
PRE_TRIG_NUM="1"
POS_TRIG_NUM="2"
LOCATION="/mnt/ramdisk/snapshot_output"
```

```
TIME_OUT="5"
```

9.2 Send Media by Ftp When Digital Input and Motion Detection Are Both Triggered

Task configuration file:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<eventmgr version="0101">
<maxprocess>1</maxprocess>
<schedule id="1">
   <duration><date>12/25-12/31</date><hour>12-15</hour></duration>
</schedule>
<di condition="0"><status id="0">normal~trigger</status></di>
<motion condition="1"><status id="0">trigger</status></motion>
<event id="1">
   <description>DI 0 and MD Window 0</description>
   <scheduleno>1</scheduleno>
   <condition>c0 and c1</condition>
   <delay>10</delay>
   <trylater>1</trylater>
   <priority>255</priority>
   <process>/usr/bin/event_do 0 5</process>
   <process>/mnt/flash/send_media_by_ftp</process>
</event>
</eventmgr>
```

Example of sending media by ftp service:

```
#!/bin/sh

FTP_PROG="/usr/bin/ftpmput"
VC_TARGET_FOLDER="/mnt/ramdisk/videoclip_output"
SS_TARGET_FOLDER="/mnt/ramdisk/snapshot_output"

# send snapshot of stream 2 (1 pre-trigger images and 2 pos-trigger images)
# send clip from stream 1 (audio and video)

# snapshot
[ ! -d $SS_TARGET_FOLDER] && mkdir $SS_TARGET_FOLDER
```

```
/usr/bin/event_media_snapshot trigger /mnt/flash/media_snapshot.conf

# clip
[ ! -d $VC_TARGET_FOLDER ] && mkdir $VC_TARGET_FOLDER
/usr/bin/event_media_clip write /mnt/flash/media_clip.conf

# wait snapshot
/usr/bin/event_media_snapshot wait /mnt/flash/media_snapshot.conf

# wait clip
/usr/bin/event_media_clip wait /mnt/flash/media_clip.conf

$FTP_PROG -u root -p 123456 172.16.2.22 Share $SS_TARGET_FOLDER
$FTP_PROG -u root -p 123456 172.16.2.22 Share $VC_TARGET_FOLDER

rm -rf $SS_TARGET_FOLDER
rm -rf $VC_TARGET_FOLDER
```

NOTE: The media required is setup in the previous example of task configuration file (section 9.1).

9.3 Send Message/Log by Http under Two Kinds of Motion Detection Conditions

The first event with id “1” will be triggered when both motion detection windows are triggered. The second event with id “2” will be triggered when one of motion detection windows is triggered.

Task configuration file:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<eventmgr version="0101">
<maxprocess>2</maxprocess>
<schedule id="1">
   <duration><weekday>1,2,3,4,5</weekday><time>22:00:00-23:59:59</time></duration>
   <duration><weekday>1,2,3,4,5</weekday><time>00:00:00-12:00:00</time></duration>
</schedule>
<motion condition="0"><status id="0">trigger</status></motion>
<motion condition="1"><status id="0">trigger</status></motion>
<motion condition="2"><status id="0">trigger</status><status id="1">trigger</status></motion>
<event id="1">
```

```
    <description>MD Window 0 and 1</description>
    <scheduleno>1</scheduleno>
    <condition>c0 and c1</condition>
    <delay>30</delay>
    <trylater>0</trylater>
    <priority>100</priority>
    <process>/usr/bin/curl -v -u root:pm -d "Both Windows Triggered" http://172.16.7.25/cgi-bin/admin/event.cgi
    </process>
    <process>/mnt/flash/send_log_by_http</process>
</event>
<event id="2">
    <description>MD Window 0 or 1</description>
    <scheduleno>1</scheduleno>
    <condition>c2</condition>
    <delay>30</delay>
    <trylater>0</trylater>
    <priority>100</priority>
    <process>/usr/bin/curl -v -u root:pm -d "One Window Triggered" http://172.16.7.25/cgi-bin/admin/event.cgi
    </process>
</event>
</eventmgr>
```

Example of sending log by http service:

```
#!/bin/sh

HTTP_CLIENT_PROG="/usr/bin/curl"

$HTTP_CLIENT_PROG -v -u root:1234 -F file=@"/var/log/messages" http://172.16.3.28/cgi-bin/uploadfile.cgi

#/usr/sbin/GeneralApp -M UPLOAD -t Text -file /var/log/messages -S 172.16.3.28 80 root 1234 -U
/cgi-bin/uploadfile.cgi -r /mnt/flash/TextLog
```


10. Notice

10.1 Space Issue

The more images and clips generated the more space it used. Because the limitation of memory space on embedded system, please pay attention to the total number of media. You can check the upper bound of media free space on the server in media section of webpage /setup/application.html. The

space of snapshot files is (PRE_TRIG_NUM + PO_TRIG_NUM + 1) ×SIZE_OF_ONE_FILE. The space of clip is about MAX_FILE_SIZE in the configuration file.

10.2 Task Files Maintenance

Please keep original task files if you want to use events of application and recording from web UI.

10.3 Media

Please use event_media_clip and event_media_snapshot to enable recording or snapshot at first. Then setup the sources of media by setup_audio and setup_video scripts.
To change setting of recording or snapshot, please stop it first and start it under new setting.
The timeout to wait media file ready must be long enough. Please check the suggestion values in “media type” section.
Remove the media files once they were processed.

<End of document>