



CyFlex® Events

Version 6

January 30, 2024

Developed by Transportation Laboratories

Version History

Version	Date	Revision Description
1	1/25/2016	Initial publication
2	8/23/2018	Formatting changes
3	3/25/2020	Retrofit to new template
4	8/4/2021	Added hypertext linked cross-references to existing references for CyFlex documents and usage help in Sections 1 and 2
5	5/16/2022	Updated all hypertext linked cross-references to cyflex.com usage help descriptions
6	1/30/2024	Rebrand to TRP Laboratories

Document Conventions

This document uses the following typographic and syntax conventions.

- Commands, command options, file names or any user-entered input appear in Courier type. Variables appear in Courier italic type.
Example: Select the `cmdapp-relVersion-buildVersion.zip` file....
- User interface elements, such as field names, button names, menus, menu commands, and items in clickable dropdown lists, appear in Arial bold type.
Example: **Type**: Click **Select Type** to display drop-down menu options.
- Cross-references are designated in Arial italics.
Example: Refer to *Figure 1*...
- Click intra-document cross-references and page references to display the stated destination.
Example: Refer to *Section 1 Overview on page 1*.
The clickable cross-references in the preceding example are *1*, *Overview*, and *on page 1*.

Related Documents

[Fast Data Logger Application User Guide](#)

[Creating User Computations and User Variables](#)

[Compressed History Data](#)

CyFlex Documentation

CyFlex manuals are available at <https://cyflex.com/>. View **Help & Docs** topics or use the **Search** facility to find topics of interest.

Table of Contents

1	OVERVIEW	1
2	USER-CREATED EVENTS	2
2.1	SIGNAL EVENTS	2
2.2	TRANSITION EVENTS	2
3	COMMON EVENT TERMINOLOGY	3
3.1	MESSAGE EVENTS	3
3.2	TIMER EVENTS.....	4
4	SPECIAL SYSTEM EVENTS	5
4.1	EMERGENCY EVENT	5
4.2	ABORT_LIMIT EVENT.....	5

1 Overview

An “event” in CyFlex refers to a method of communicating and signaling between processes. An event might contain data or might be used as a signal without any information within. An event of zero length contains no data and is used simply as a signal.

Most events in the system are created by applications for specific purposes. Two particular types of events can be created by users, refer to *Section 2 User-Created Events* on page 2. More complicated events containing data are used to communicate between applications and are created and used by the applications.

Events have **names** which are alphanumeric strings. The naming convention is similar to variable names except that event names may begin with numerals and are limited to 32 characters. The names must be unique in a system. An attempt to create an event could result in an error being logged indicating that the name is already in use.

When an application sends an event, it could be received by one application, by multiple applications, or possibly by none. If there is no application to receive an event, the sending application will log an error.

A system task, called the Event Administrator acts somewhat like an email server in that it receives all events, stores them in a queuing system, and delivers them to their final destination when the receiving task is ready to process them.

Each application can have a “queue” of events, so it is possible for a series of events to be received by the Event Administrator in rapid succession, faster than the receiving task can process them.

- If the number of events received in a such a burst such is smaller than the queue, the application will receive them in the sequence which they were sent.
- If an event is received by the Event Administrator after the queue of events is filled, the sending application may be informed that it will have to retry the send. This is the case if a characteristic for an event is specified to be “data-critical”.

@Note:

The data-critical characteristic cannot be specified in user-created events.

There are two fundamental types of events: messages and timers:

1. Messages are always sent by a CyFlex application.
2. Timers can be created and received by CyFlex applications, but the timers are actually set by the Event Administrator.

2 User-Created Events

Both of the following types of user-created events are message types.

2.1 Signal Events

Signal events are messages of zero length and are used to signal an application to perform a specific function.

Examples:

- Start or stop a process such as data logging; refer to the [Fast Data Logger Application User Guide](#)
- Perform a particular calculation; refer to [Creating User Computations and User Variables](#)

Signal events can be created in the files that specify user computations (`gen_labels` files). Refer to usage help for [gen_labels](#) and [Creating User Computations and User Variables](#).

2.2 Transition Events

A transition event signals the occurrence of the change of value of a logical, integer, string, or real variable. For example, logical variables associated with limits, digital inputs and outputs, as well as user-created logical variables all allow transition events to be associated with them.

For a CyFlex logical variable, a *true-transition* event is the event which occurs when the value is changed from FALSE to TRUE. A *false-transition* event is the event which occurs when a logical variable's value is changed from TRUE to FALSE. An event does not occur if an application sets a variable to a value which is not a change from its current value. For example, if a logical variable's value is TRUE and the value is set to TRUE.

Transition events associated with integer and string variables occur whenever the value of the variable changes.

Transition events associated with real variables occur when the change in value exceeds the 'history tolerance' value specified for that variable. Refer to [Compressed History Data](#) for a discussion of "history tolerance".

3 Common Event Terminology

Events can be of various types and configuration. The following sub-sections are intended to promote understanding of some of the terminology. Some of these terms may show up in system error messages.

3.1 Message Events

Events get created, destroyed, sent (set), queued, and received. The following terms are often used in the documentation to describe these actions.

- **Create:** an application performs the function of creating an event by specifying the type and other information such as message length, temporary/permanent, etc.
- **Attach:** an application attempts to locate a system event by name and informs the system that it wants to receive this event whenever it is sent. If the event does not exist, the application will receive an error message. When applications such as the Data Logger have events specified to start or stop logging, they will attempt to attach to those events. Multiple applications may attach to the same event.
- **Set:** when an application sends an event, it is often referred to as have been “set”. Setting an event includes transferring data to attached applications if the event is of the message type and has a length greater than zero.
- **Destroy:** an event can be removed from the system, but only by the application that created it, and only if it is a “temporary” event. Any application which is attached to an event that is destroyed will receive a special error message.
- **Queue:** all applications which receive events may have a buffer called a “queue” in the Event Administrator which store events if they are received faster than they can be processed

Characteristics of message events:

- Time-critical or Data-critical
 - Time-critical implies that the data contained in the messages is less important than receiving the message as a signal. If the queue fills, the oldest event will be removed to make room for the newer event.
 - Data-critical implies that the information within the event should not be lost if the event queue in the Event Administrator is full. Therefore, the sending task will be notified of the queue status and can retry to send the data as the receiving application(s) process the events.
- Temporary or Permanent
 - Permanent events will remain in the system even if the task that creates them runs to completion or dies.
 - Temporary events will be removed from the system when the task that creates them comes to an end.
- Length of message
- Queue size
 - Maximum number of events of this name that will be stored by the Event Administrator in the queuing system.

3.2 Timer Events

Characteristics of timer events:

- Temporary or Permanent
- Periodic or One-shot
 - A periodic event will be sent by the Event Administrator once created for as long as the event exists
 - One-shot timers are set once. An application that “attaches” to a one-shot timer event will receive the event later at a time period specified as the timer interval. These are often used by applications as time-outs should they be waiting for another event.
- Timer interval
- Queue size

Timer events are set by an administrator task. The application that creates a timer event does not have to set the event. Periodic timers are generated repeatedly at the interval specified.

One-shot timers are set once. An application that “attaches” to a one-shot timer event will receive the event later at a time period specified as the timer interval.

4 Special System Events

CyFlex will create special events when the system is launched by a `go`. Two of these are listed below.

4.1 Emergency Event

This event is normally used to initiate a rapid transition of the test cell and/or test fixture to a safe condition. Various applications may receive this event and start a sequence of actions to perform the required actions. These actions are defined by user specifications for applications such as the Test Manager ([gp_test](#)) or the PLC Simulator ([evnt_rsp](#)).

The most common way of generating the “emergency” event is for the “limit” application to sense that a limit value has been exceeded, for example, perhaps a temperature or pressure reading exceeds a limiting value.

4.2 abort_limit Event

This event is similar to the emergency event except that it is generally used for situations that don’t require a rapid safety sequence. Instead, the test system can be taken to a safe condition in a slow and orderly sequence that may take several minutes and reduce the possibility of damage to the test object.