Innovative Teaching Approaches in development of Software Designed Instrumentation and its application in real-time systems

The Advanced Applications of LabVIEW

Lecture 4: Producer/Consumer User Event Design Pattern

Co-funded by the
Erasmus+ Programme
of the European Union

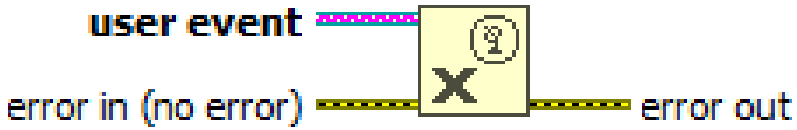
User Event

- User event allows to generate the event from block diagram.
- The user event is commonly used in the advanced design pattern as master-slave or producer-consumer.
- Functions, which are used for user event, are put below:

Create User Event

Returns a reference to a user event. LabVIEW uses the **user event data type** you wire to determine the event name and data type of the event. Wire the **user event out** output to a Register For Events function to register for the event. Wire the **user event out** output to a Generate User Event function to send the event and associated data to all Event structures registered for the event.

Destroy User Event



Releases a user event reference by destroying its associated user event refnum. Any Event structures registered for this user event no longer receive the event.

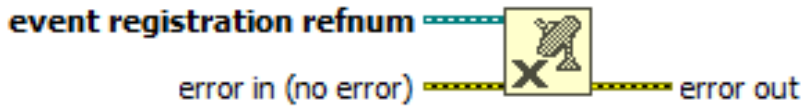
User Event

Register For Events

Dynamically registers events. The events for which you can register depend on the type of the reference you wire to each **event source** input. Wire the **event reg refnum out** output to an Event structure or to another Register For Events function.

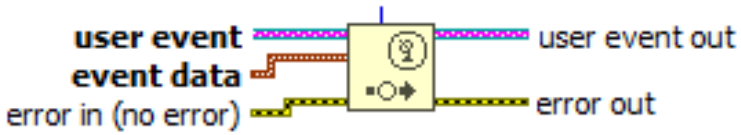
Detailed help

Unregister For Events



Unregisters all events associated with an event registration refnum.

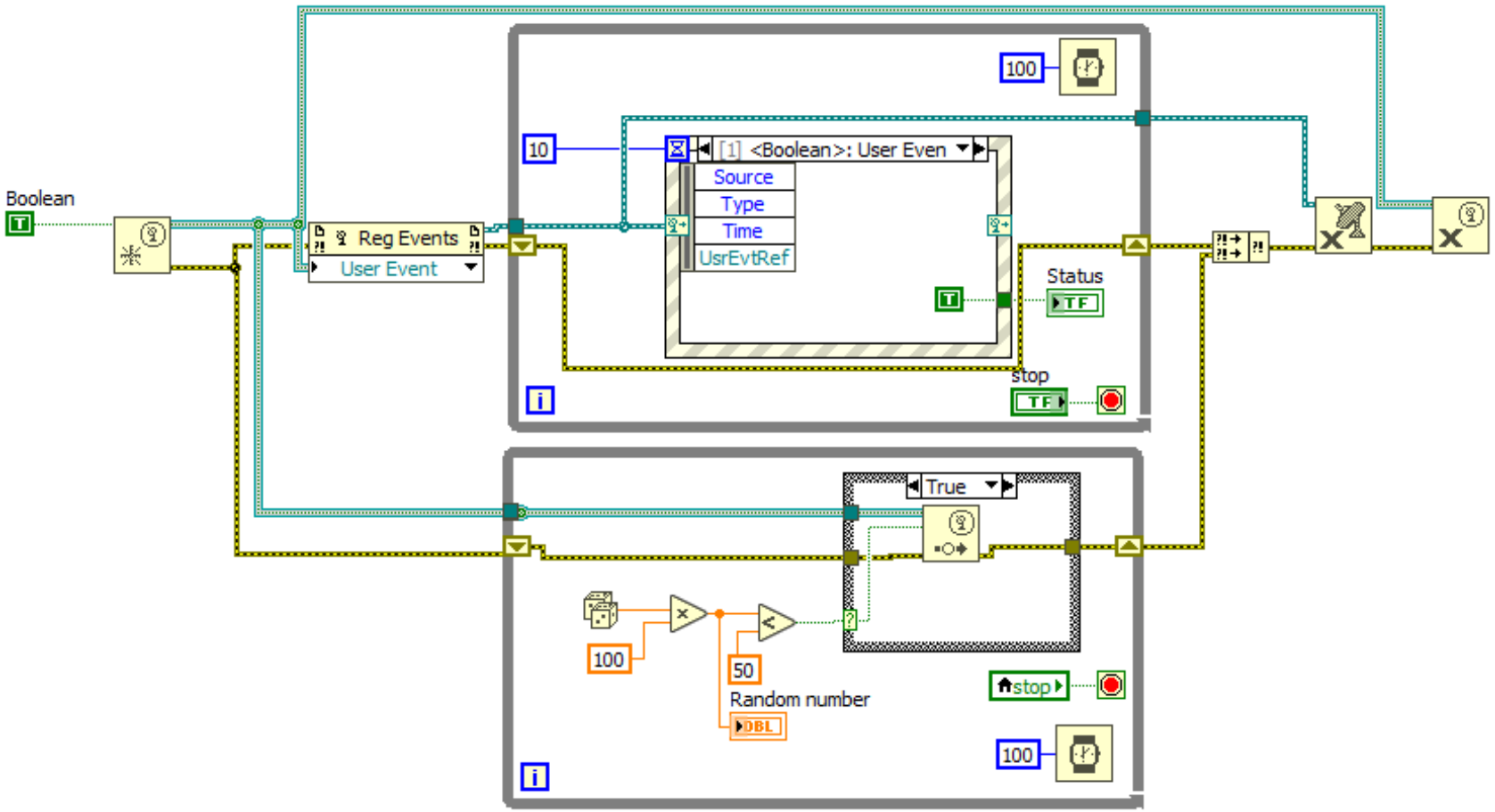
Generate User Event



Broadcasts the user event you wire to the **user event** input and sends the user event and associated event data to each Event structure registered to handle the event.

User Event example

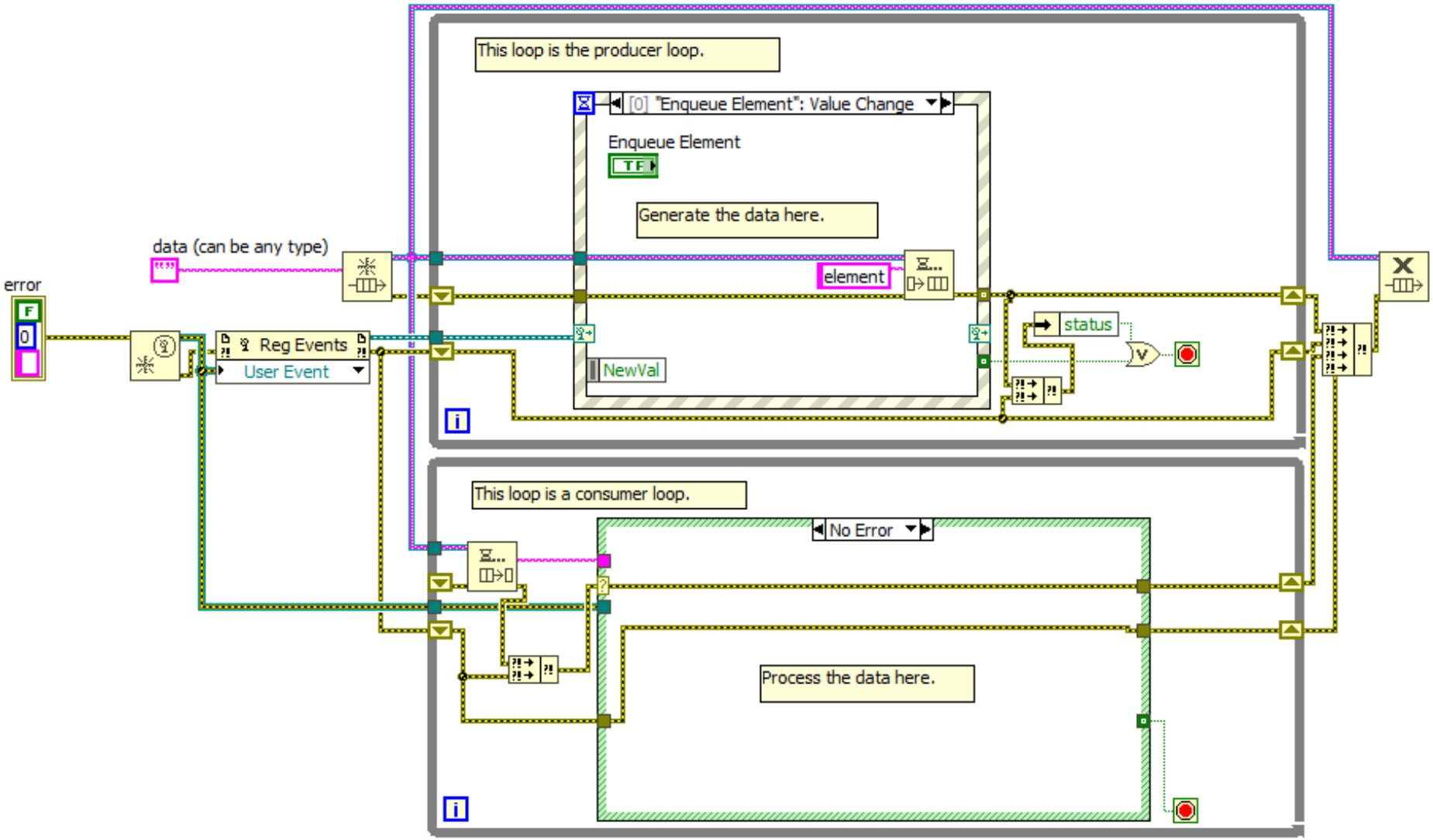
- Example: a random number is generated and when the number is smaller than 50, the user event is generated. Than a frame: "<Boolean>: User Event" will run and the "Status" led will be turned on.



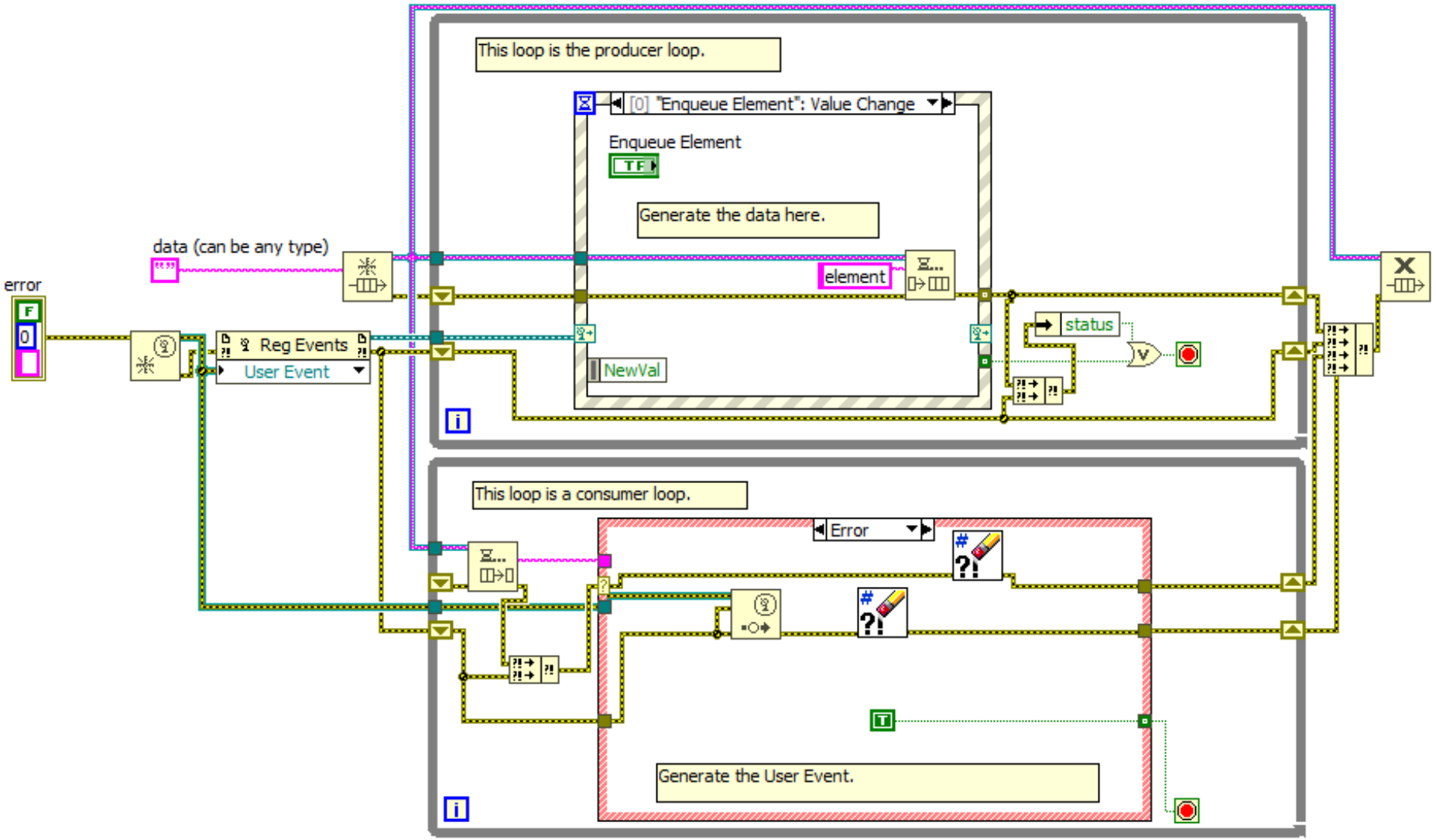
Producer/Consumer User Event Design Pattern

- The Producer/Consumer Design Pattern uses the queue for communication between two loops. The producer loop sends a messages or values to consumer loop.
- A disadvantage of this design pattern is a lack of possibility to send a message or value from consumer loop to producer loop.
- The consumer loop should send the message about e.g. errors to producer loop, where decision of further steps should be taken. Such a possibility is created by a modified version of the pattern called: producer / consumer user event design pattern.

Producer/Consumer User Event Design Pattern



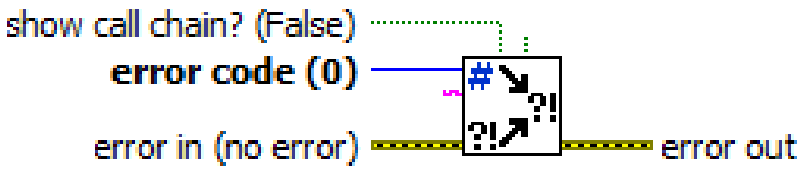
Producer/Consumer User Event Design Pattern



Errors

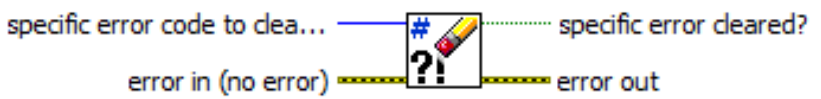
- The errors and warnings are transmitted as the cluster, which consists on: status - boolean value, code - 32-bit signed integer, source - string.
- The error has a “True” status value, the warning has a “False” status value.
- Functions below are commonly used with dealing with errors:

Error Cluster From Error Code.vi



Converts an error or warning code to an error cluster. This VI is useful when you receive a return value from a shared library call or when you return user-defined error codes.

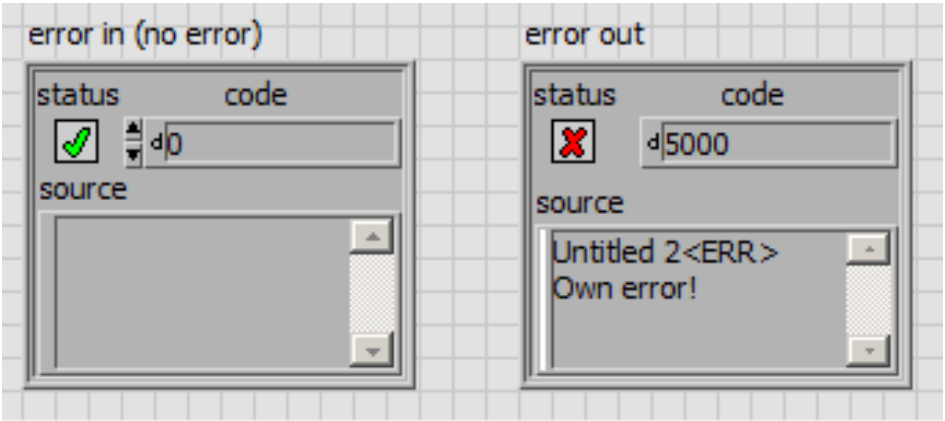
Clear Errors.vi

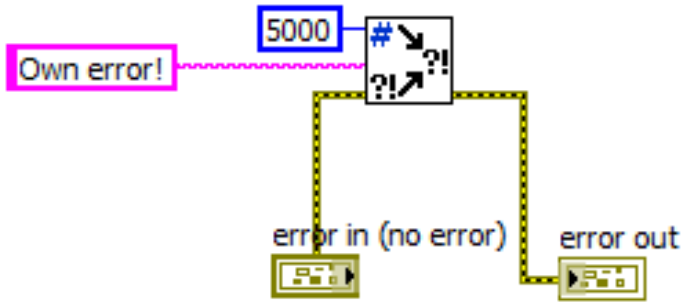


Resets the error **status** to no error, **code** to 0, and **source** to an empty string. Use this VI when you want to ignore an error. By default, this VI ignores all errors. Wire an error code value to **specific error code to clear** if you only want to ignore a specific error.

Errors

- Sometimes, there is a need to generate own errors.
- Own errors can have the codes: (-8999;-8000), (5000, 9999) and (500 000, 599 999).





Thank you for attention!

Lecture was prepared based on materials from: "LabVIEW Core 3 Course Manual".

This project has been funded with support from the European Commission. This communication reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.