

README – E-Prime 3.0 LIBLSL Package File

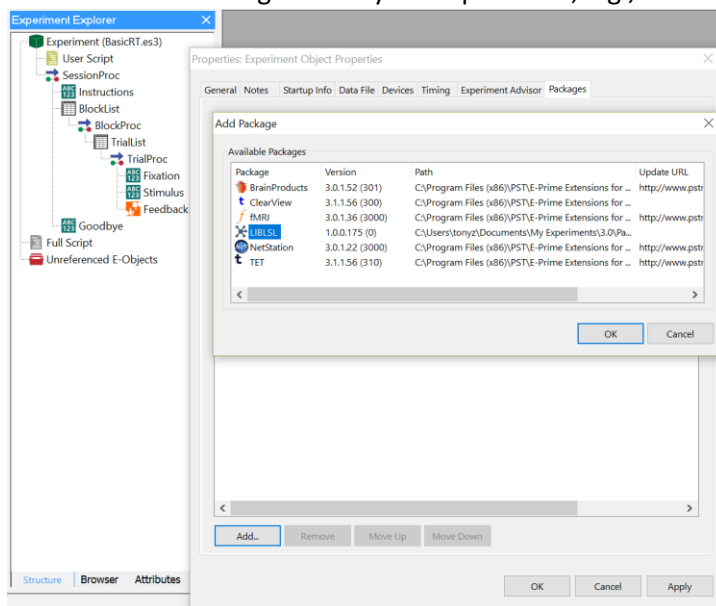
2018-10-07

INSTALLATION

1. Uncompress the distribution file.
2. Move the *LIBLSL* folder to the folder ...*\Documents\My Experiments\3.0\Packages*
 - a. The *LIBLSL* folder contains the *LIBLSL* Package File and related resources.
 - b. By default, it must exist in the E-Prime 3.0 user Packages folder in order to be located by E-Studio 3.0.
 - c. If you want to keep the *LIBLSL* folder in an alternate location, you can select Tools | Options | Packages from the E-Studio menu to add a specific folder to the list of PackageFile Search Folders that E-Studio uses to locate for Package Files.
3. Move the *LIBLSL Samples* folder to the folder ...*\Documents\My Experiments\3.0*
 - a. The *LIBLSL Samples* folder contains any sample experiments related to the *LIBLSL* Package File.
 - b. Each sample experiment subfolder contains the *libls132.dll* file. This dynamic link library contains all the library routines that can be accessed from E-Basic script.
 - c. The *libls132.dll* file MUST exist in your experiment folder (e.g., where your *.es3 and *.ebs3 experiment files exist), or must exist in a folder that is listed in the system PATH environment variable.
 - d. If the *LIBLSL* Package File cannot load the *libls132.dll* successfully when the experiment is launched, a notification dialog will be shown.
 - e. The most recently updated *libls132.dll* and current documentation regarding the routines contained in the library can be obtained from <https://github.com/scen/labstreaminglayer>

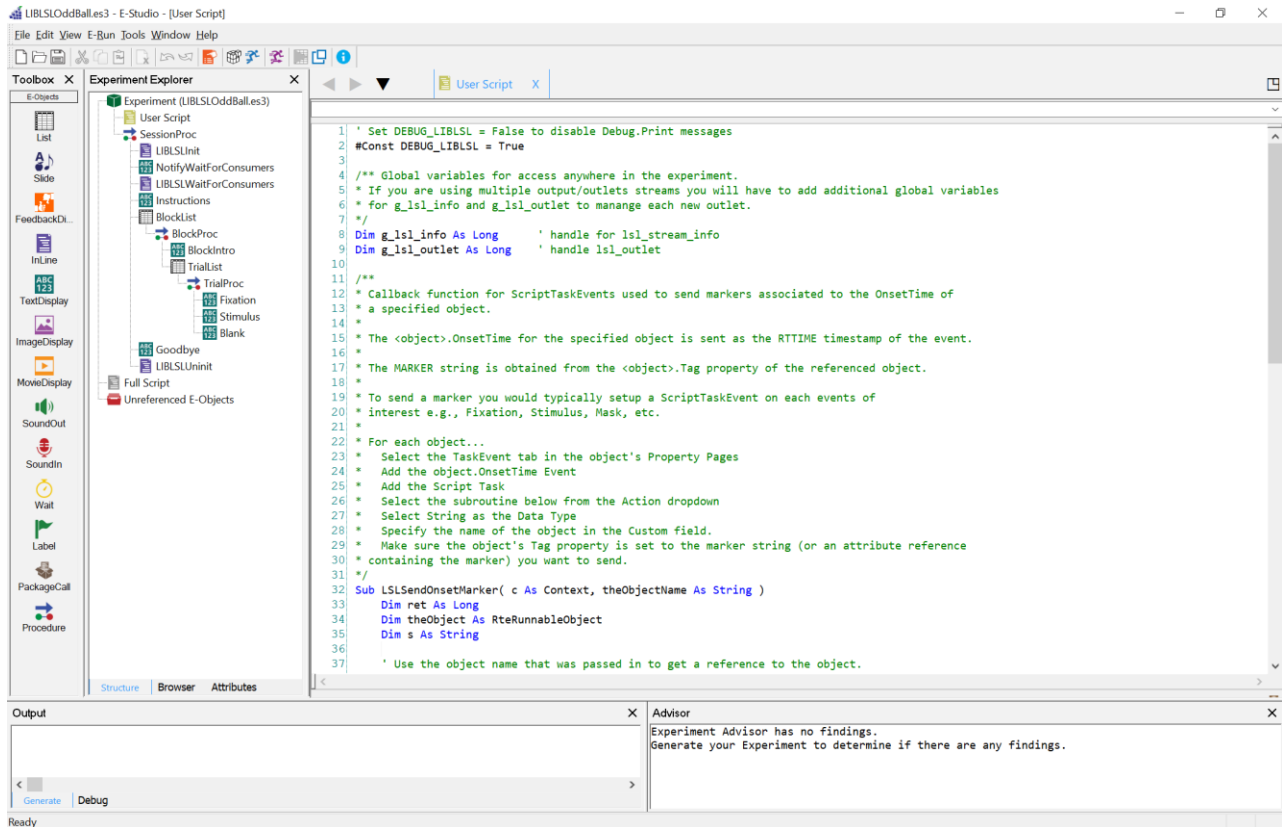
ADDING LSL SUPPORT TO AN E-PRIME EXPERIMENT

1. Follow the instructions above to setup the appropriate folder structure.
2. Copy the *libls132.dll* file from one of the *LIBLSL Samples* folder tree to the folder containing the E-Prime 3.0 experiment you want to use.
3. Add the *LIBLSL* Package File to your experiment, e.g.,

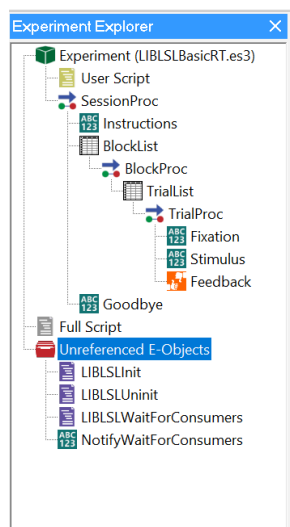


4. Launch a second instance of E-Studio and open the *LIBLSLOddBall.es3* from the ...*\Documents\My Experiments\3.0\LIBLSL Sample* experiments folder.

5. Copy the contents of the User Script from the LIBLSLOddBall experiment into the User Script section of your experiment.

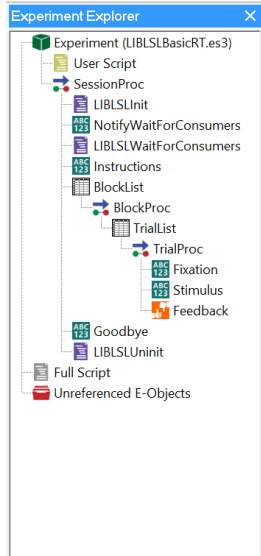


- a. This User Script defines two example Subroutines that can be called from a Script Task event (or optionally from InLine script.)
 - i. The *LSLSendOnsetMarker* subroutine can be called (typically from a Script Task Event) to send a marker via LSL at an object's OnsetTime. The string name of the object is passed as a parameter. The object's .OnsetTime is sent as the timestamp for the event, and the contents of object's .Tag property is sent as the marker.
 - ii. The *LSLSendMarker* subroutine can be called (typically from InLine Script) to send a marker via LSL at the current time. The marker is passed as a parameter and the current time (read from the E-Prime clock) is sent as the timestamp.
6. Copy the following objects from the LIBLSLOddBall experiment into the Unreferenced E-Objects of your experiment.



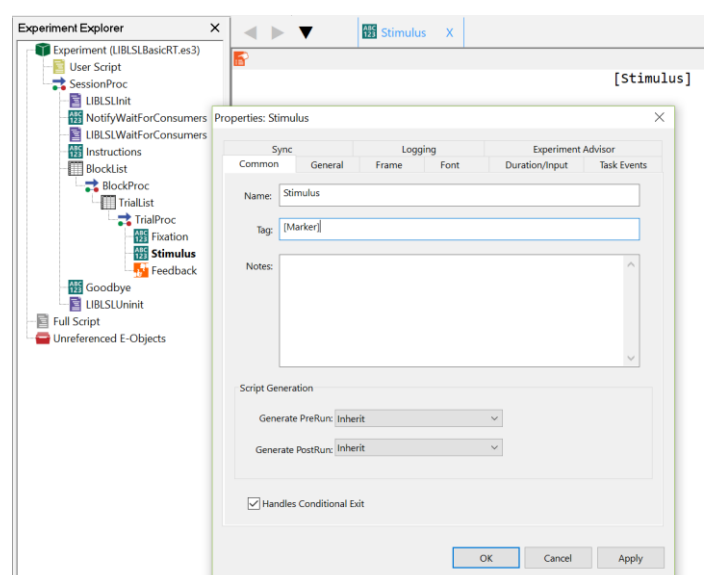
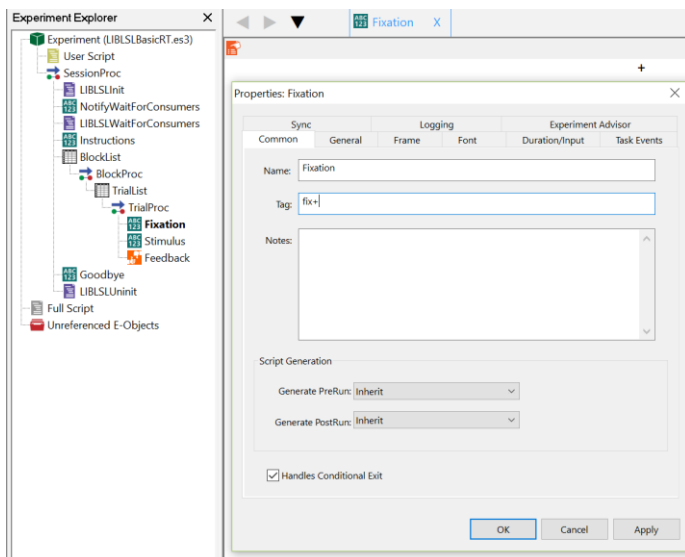
- a. *LIBLSLInit* – An InLine object with script that initializes LSL
- b. *LIBLSLUninit* – An InLine object with script that uninitializes LSL
- c. *LIBLSLWaitForConsumers* – An InLine object with script that waits (30 seconds) for any LSL Consumers connect to the experiment.
- d. *NotifyWaitForConsumers* - (optional) TextDisplay to display a message while waiting for LSL Consumers to connect to the experiment.

7. Move the above objects into the following equivalent locations within your experiment structure.

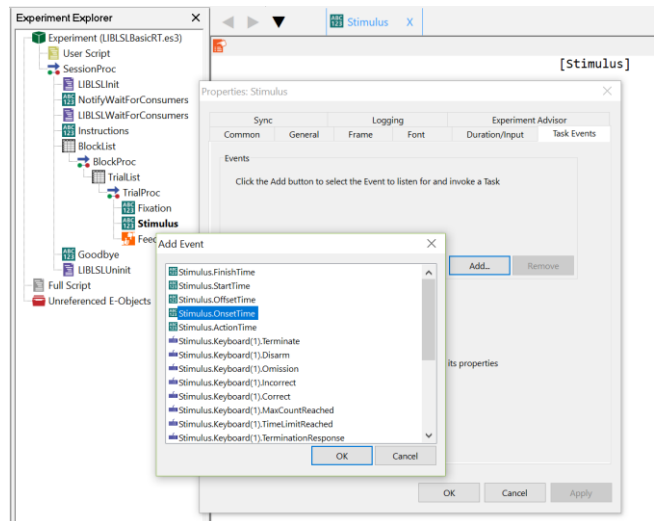
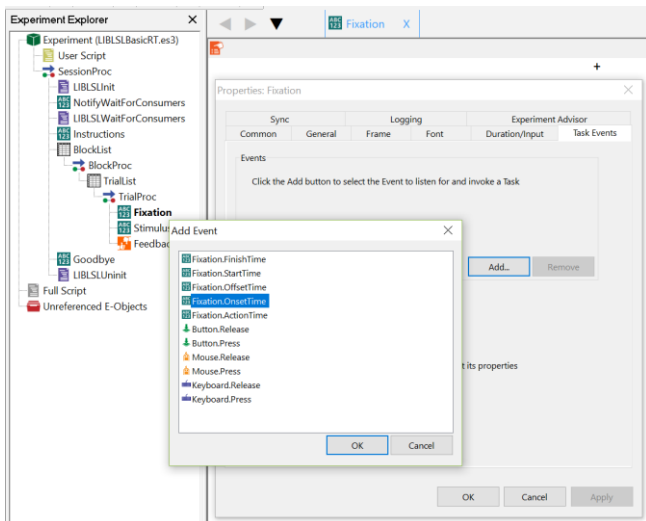


8. For each object in your experiment that you want to have send a marker...

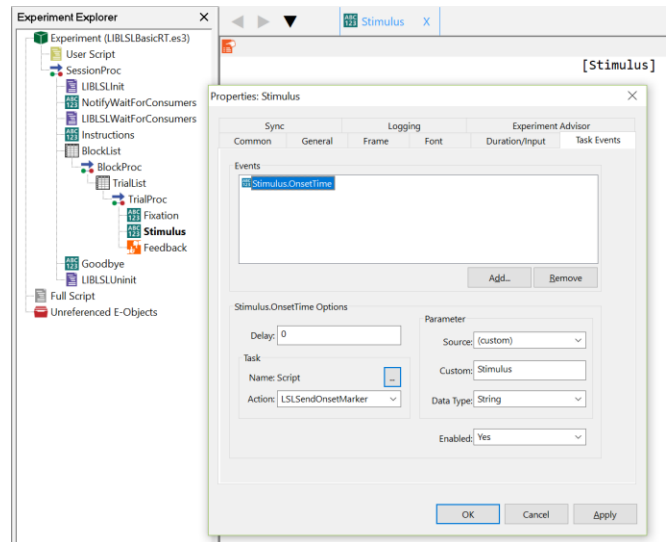
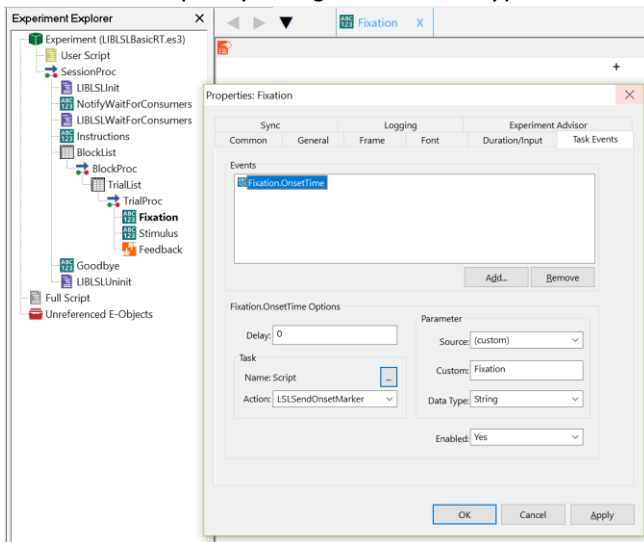
- a. Set the <object>.Tag property to the marker string that you want to send e.g., “fix+” to mark the onset of the Fixation object (left image). You can set the <object>.Tag property to an Attribute reference if the marker value varies trial by trial e.g., “[Marker]” to send a marker unique to each type of stimulus presented by the Stimulus object on each trial (right image).



b. Create a Script Task Event on the <object>.OnsetTime Event

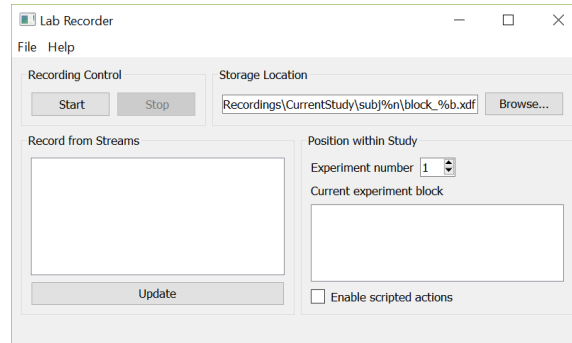


- Select the *LSLSendOnsetMarker* subroutine from the *Action* dropdown
- Specify the name of the object in the *Custom* field
- Specify *String* as the *Data Type*

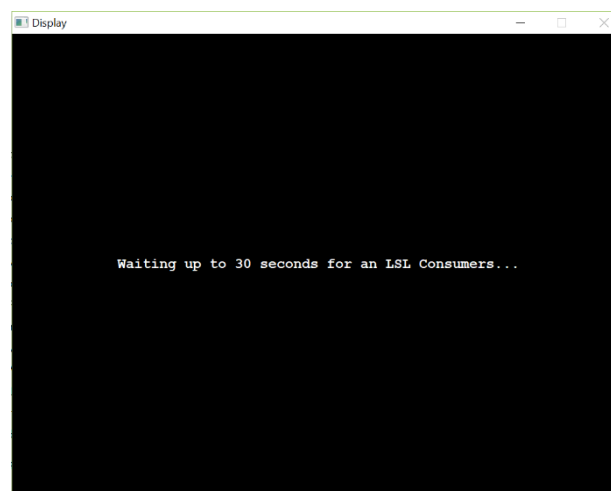


RUNNING AN EPRIME EXPERIMENT WITH LSL SUPPORT

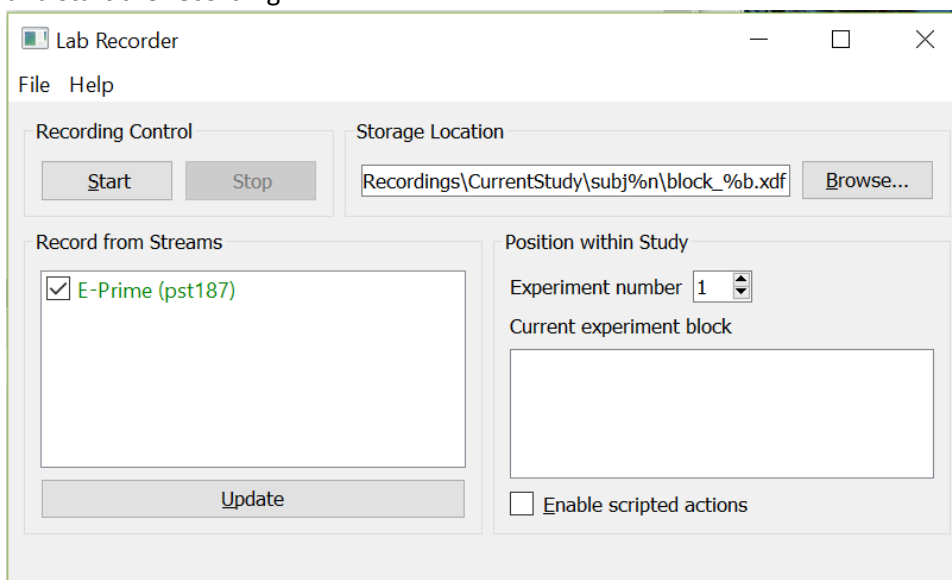
1. Launch the LSL Consumer application(s). The LSL Lab Recorder is used in this example as a consumer.



2. Launch your E-Prime experiment. Note - if testing the experiment while Lab Recorder is running on the same machine it is recommended that you launch/test the experiment using Windowed mode so that you can switch back and forth between applications.



3. After the experiment launches configure the LSL Consumer application to record from the “E-Prime” stream and start the recording.



4. After the experiment completes you can stop the recording. When using Lab Recorder you can review the contents of the *.xdf file to confirm the markers.